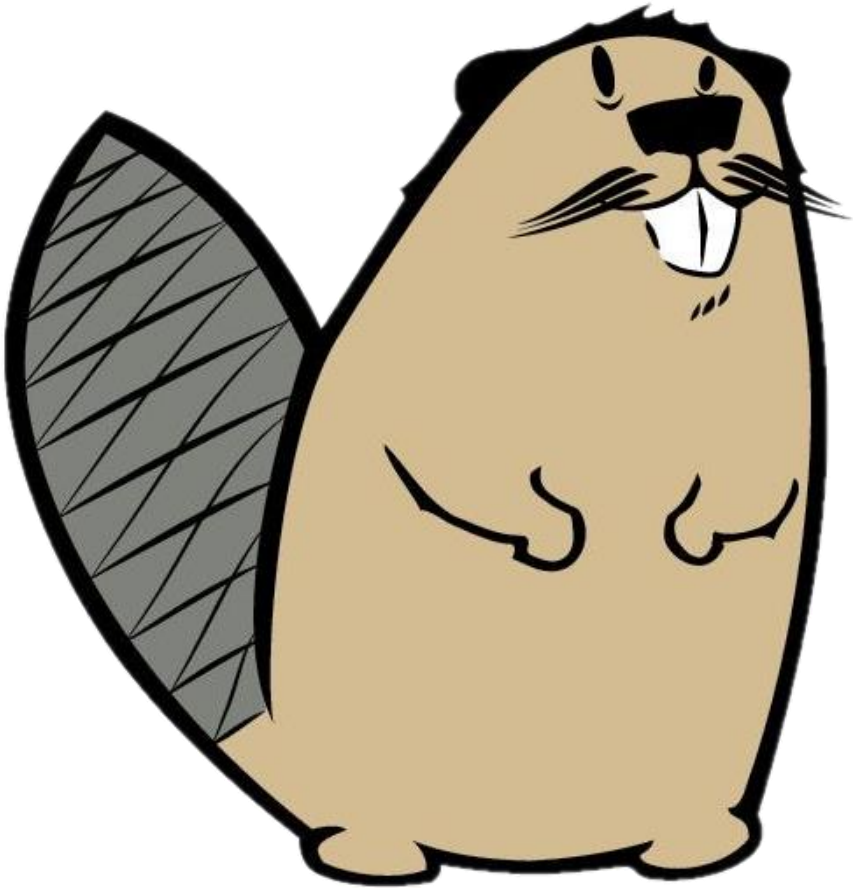
HÓDÍTSD MEG A BITEKET!

INFORMATIKAI GONDOLKODÁST TÁMOGATÓ, NEMZETKÖZI BEBRAS KEZDEMÉNYEZÉS MAGYAR MEGVALÓSULÁSA



2023

MI IS AZ E-HÓD

Az e-HÓD/HÓDítsd meg a biteket a nemzetközi BEBRAS-kezdeményezés magyar partnere.

A nemzetközi Bebras, melyhez 2023-ban már több mint 70 ország kapcsolódott, 2015-ben elnyerte az Informatics Europe „Best Practices in Education" díját.

A kezdeményezés alapja Dr. Valentina Dagiene litván professzor által életre keltett verseny, melynek célja, hogy rövid, gyorsan (kb. 3 perc alatt) megérthető és megoldható feladatokkal megvalósítsa az alábbiakat:

- felkeltse az érdeklődést az informatika iránt;
- feloldja az informatikával kapcsolatos félelmeket, negatív érzéseket;
- megmutassa az informatika sokszínűségét, felhasználási lehetőségeit és területeit.

A kérdések három nehézségi szinten csak strukturált és logikus gondolkodást igényelnek, semmilyen különleges informatikai tudás nem szükséges a megválaszolásukhoz. A feladatok érdekes problémákat mutatnak be. Nem tesztek, inkább szórakoztató gondolkodtató feladványok.

Magyarországon 2023-ban tizenharmadik alkalommal, öt korcsoportban vehettek részt a diákok 4-től 12. osztályig.

A versenyt az ELTE IK és az NJSZT Közoktatási Szakosztálya szervezi.

Az alábbi dokumentumban a 2023-as magyar verseny feladatai és megoldásai találhatóak.

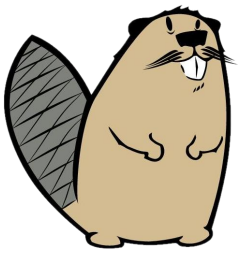
További információkért látogasson el a HTTP://E-HOD.ELTE.HU/ weboldalra, vagy írjon email-t az info@e-hod.elte.hu címre.

RÉSZVÉTEL

A részvétel mindenki számára ingyenes.

A verseny november második és harmadik hetében kerül lebonyolításra, osztályonként kiválasztható, hogy az adott héten melyik napon mikor oldják meg a feladatokat (8:00 és 16:00 között). Ezzel biztosítható, hogy akár egy tanóra keretein belül tudjanak részt venni egész osztályok.

A résztvevő diákoknak egy-egy internet kapcsolattal rendelkező számítógépre van szükségük. A feladatok megjelenítése és elküldése minden böngészőn működik. A verseny befejezése után, a hód hetet követően kerülnek nyilvánosságra a megoldások, melyek lehetőség szerint átbeszélhetők ugyancsak akár egy tanóra keretein belül.



SZABÁLYOK

- A verseny lebonyolítása iskolai helyszíneken történik.
- A résztvevők online kapják meg és válaszoljak meg a kérdéseket;
- A versenyre fordítandó idő 45 perc, 18 feladat három nehézségi szinten: könnyű, közepes és nehéz (legkisebb korosztályban 12 feladat);
- A verseny alatt semmilyen más számítógépes program, alkalmazás nem használható;
- A verseny során nyugalmas környezetet kell biztosítani;
- A terem a verseny során nem hagyható el;
- Az esetleges számítógéppel, internettel kapcsolatos észrevételeket a kontakt személynek kell összegyűjtenie és továbbítania a szervezők felé;
- A verseny célja: minél több pont összegyűjtése helyes válaszok megjelölésével, helytelen válaszok eseten pontlevonás történik;
- A kérdések tetszőleges sorrendben megválaszolhatók;
- A kérdések, problémák megértése a feladat részét képezi. Ezért a feladatok megbeszélése és értelmezéssel kapcsolatos kérdések nem megengedettek;
- A megoldások a verseny befejezése után, a hód hetet követően kerülnek nyilvánosságra.

ÉRTÉKELÉS, PONTOZÁS

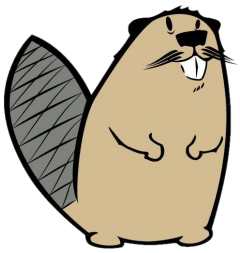
A Kishód korcsoportban 12, minden más korcsoportban 18 feladatot kell megoldani három nehézségi szinten. Minden helyes válasz pontot ér, minden helytelen válaszért pontlevonás jár.

Nem megválaszolt kérdés esetében az összpontszám változatlan marad.

Az alábbi táblázat mutatja, hogy a feladatok nehézségétől függően hány pont kerül jóváírásra, illetve levonásra:

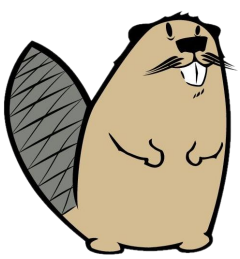
	Könnyű	Közepes	Nehéz
Helyes válasz	6 pont	9 pont	12 pont
Helytelen válasz	-2 pont	-3 pont	-4 pont

Összesen (18 feladat esetében) maximum 216 pont érhető el, mivel mindenki 54pontról indul.

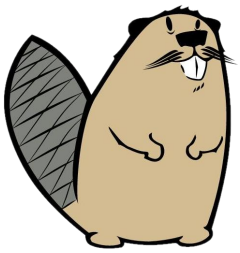


TARTALOMJEGYZÉK

Mi is az E-HÓD 2
Részvétel 2
Szabályok 3
Értékelés, Pontozás 3
Tartalomjegyzék 4
Feladatok 6
Az állatkertben (2023-AT-01) 7
Növénytársulások (2023-AU-01b) 10
Hódkockák (2023-AU-05b) 13
Emma tennivalói (2023-BE-01) 16
Kutak (2023-BR-05) 19
Ricca (2023-CA-02) 22
Esernyő (2023-CH-01) 24
Tárolás (2023-CH-05) 27
Ütköző robot (2023-CZ-01) 30
Modellvasút (2023-CZ-02) 33
Almaszeletek (2023-CZ-03) 37
Konfliktusdetektor (2023-DE-01) 40
Virágbolt (2023-DE-02) 43
Emese álomháza (2023-DE-04) 46
Kártyatérkép (2023-DE-06) 48
Zerobot dilemma (2023-DE-08) 51
Dominó (2023-DE-09) 53
Hódvár (2023-HU-37) 57
Ogham kód (2023-IE-02b) 60
Vonatrakodás (2023-IN-03b) 62
Nyelviskola (2023-IR-02) 64
Hódok és vidrák (2023-KR-01) 67



Emelem kalapom (2023-LT-01) 69
Fotó (2023-LT-02) 72
Forgó csövek (2023-ME-03b) 74
Hídépítés (2023-NZ-01) 77
Tamás és barátai (2023-PE-02) 80
Postfix jelölés (2023-RO-02) 82
Labdarendezés (2023-SA-01) 85
Répaültetés (2023-SK-02) 87
Közel vagy távol (2023-SK-07) 91
Hegymászás (2023-TW-04) 94
Számzár (2023-UA-01) 97
Önvezető autó (2023-US-03) 100
Kincsesládák (2023-UY-01) 103
Véletlen képek (2013-DE-02) 106
Rekurzív festés (2016-AT-06) 108
Bontsd részekre(2017-RU-05) 110
Cipővásárlás (2019-KR-04) 112
Támogatóink, köszönetnyílvánítás 116



FELADATOK

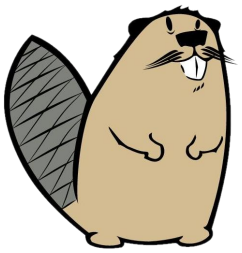
Kishód: 2023-CH-01, 2023-CZ-03, 2023-DE-06, 2023-LT-01
2023-CA-02, 2023-DE-04, 2023-SK-02, 2023-US-03
2023-AT-01, 2023-DE-02, 2023-KR-01, 2023-SA-01

Benjamin: 2023-DE-06, 2023-AT-01, 2023-DE-02, 2023-KR-01, 2023-SA-01, 2023-LT-02
2023-CA-02, 2023-DE-04, 2023-SK-02, 2023-US-03, 2023-AU-01b, 2023-UY-01
2023-AU-05a, 2023-CH-05, 2023-CZ-02, 2023-IN-03b, 2023-PE-02, 2023-SK-07

Kadét: 2023-CA-02, 2023-DE-04, 2023-SK-02, 2023-US-03, 2023-AU-01b, 2023-UY-01
2023-AU-05a, 2023-CH-05, 2023-CZ-02, 2023-IN-03b, 2023-PE-02, 2023-SK-07
2023-BR-05, 2023-CZ-01, 2023-HU-37, 2023-IE-02b, 2023-IR-02, 2023-ME-03b

Junior: 2023-AU-05a, 2023-CH-05, 2023-CZ-02, 2023-IN-03b, 2023-PE-02, 2023-SK-07
2023-BR-05, 2023-CZ-01, 2023-HU-37, 2023-IE-02b, 2023-IR-02, 2023-ME-03b
2023-BE-01, 2023-DE-08, 2023-NZ-01, 2023-RO-02, 2023-TW-04, 2023-UA-01

Senior: 2023-BR-05, 2023-CZ-01, 2023-HU-37, 2023-IE-02b, 2023-IR-02, 2023-ME-03b
2023-BE-01, 2023-DE-08, 2023-NZ-01, 2023-RO-02, 2023-TW-04, 2023-UA-01
2013-DE-02, 2016-AT-06, 2017-RU-05, 2019-KR-04, 2023-DE-01, 2023-DE-09

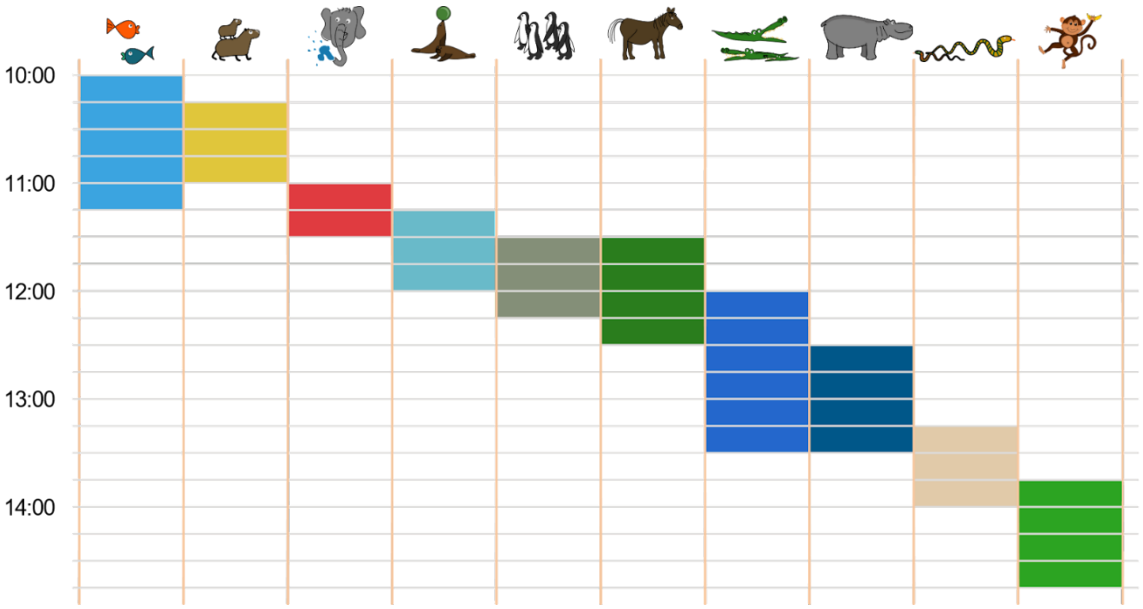


AZ ÁLLATKERTBEN (2023-AT-01)

KISHÓD – KÖZEPES
BENJAMIN – KÖNNYŰ

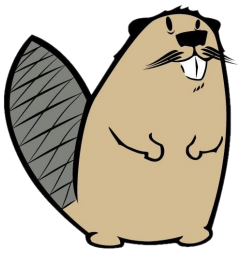
Anna ma az állatkertbe kirándul és szeretne minél több állatbemutatót megnézni.

Az alábbi plakáton látszik az összes bemutató időpontja. Például, hogy a majmok bemutatója 13:45-kor kezdődik és 14:45-kor ér véget.



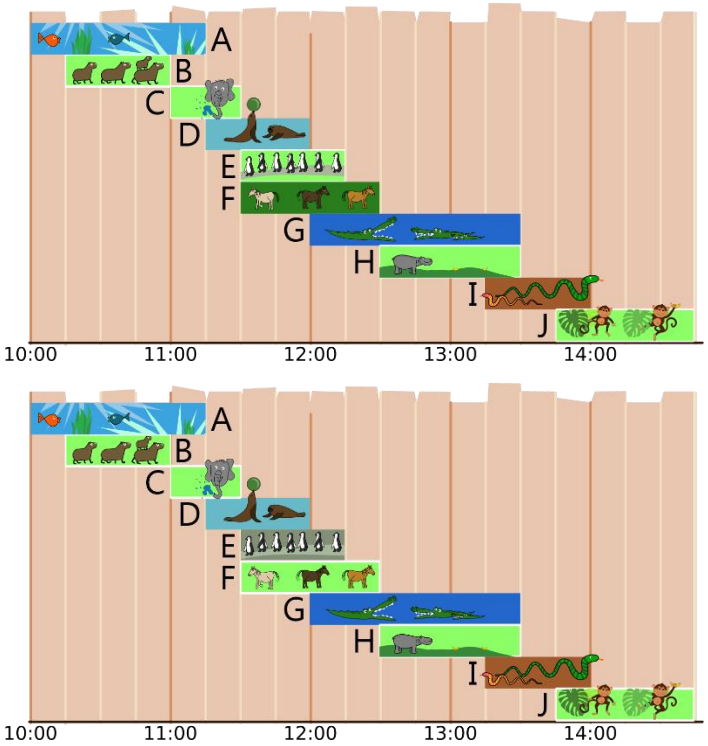
Ha Anna részt vesz egy bemutatón, akkor azon az elejétől a végéig ott kell lennie, és egy időben csak egyetlen bemutatón tud részt venni.

Segíts Annának kiválasztani úgy a bemutatókat, hogy a lehető legtöbb bemutatón tudjon részt venni.



Megoldás

Anna legfeljebb 5 bemutatón tud részt venni egymás után. Az alábbi a két lehetséges, helyes megoldás (zölddel jelölve).

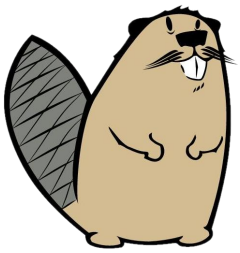


Az egyik módszer az, amikor felsoroljuk az összes lehetséges összeállítást, ahogy a bemutatón részt tudunk venni. Az összes lehetséges összeállítás közül a legtöbb bemutatót tartalmazó a legjobb megoldás. Az összes lehetséges összeállítás megtalálása viszont nagyon időigényes.

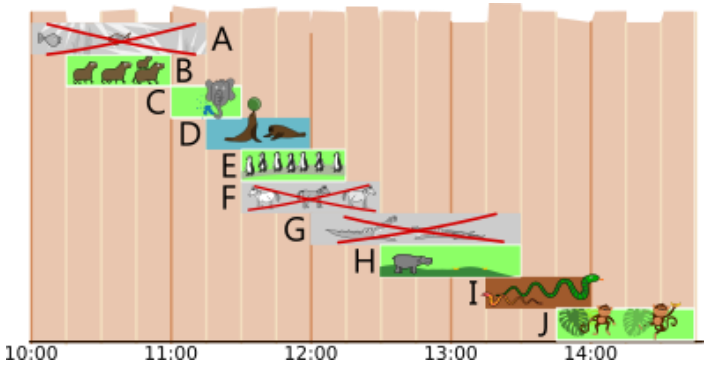
Hogyan lehet bebizonyítani, hogy nincs olyan összeállítás, ahol egy nap alatt ötnél több bemutatón lehet részt venni? Tegyük fel, hogy lehetséges lenne 6 bemutatón részt venni, majd mutassuk meg, hogy ez a feltételezés ellentmondáshoz vezet.

Ha alaposan megnézzük a plakátot, láthatjuk, hogy az egész napot 19 időegységre osztották. Az egyes bemutatók 2, 3, 4, 5 vagy 6 időegység hosszúságúak (lásd az alábbi táblázatot). Annak érdekében, hogy minél több bemutatót láthasson, Anna a lehető legtöbb rövid bemutatót választja.

Időegység	Bemutató
2	C
3	B, D, E, I
4	F, H, J
5	A
6	G



De a 6 legrövidebb bemutató együttesen legalább 18 időegység hosszú (2 + 3 + 3 + 3 + 3 + 3 + 4). Ezek között vannak többek között a C, D és E bemutatók. Mivel a C és E bemutatók pontosan egymás után következnek, a kettő között a D bemutatón nem lehet részt venni (lásd az alábbi képet). Tehát a D bemutatót ebben az esetben ki kellene cserélnünk egy másikra, amely legalább 4 időegységig tart (mivel már 3 hosszúságú nincs).



A D bemutató nélkül összesen legalább 19 időegységre van szükségünk (2 + 3 + 3 + 3 + 4 + 4). Összesen 3 olyan bemutató van, amely 4 időegység hosszú. Ezek a bemutatók azonban az előzőekkel egy időben zajlanak. Így a 2 + 3 + 3 + 3 + 3 + 4 + 4 + 4 kombináció sem lehetséges, és legalább 20 időegységre van szükségünk 6 bemutatóhoz. Ez azonban hosszabb, mint az állatkert nyitvatartási ideje, ami nem lehetséges. Ez tehát egy ellentmondás! Megállapítjuk, hogy nincs olyan érvényes terv, amely 5 bemutatónál többet tartalmaz.

MIÉRT INFORMATIKA?

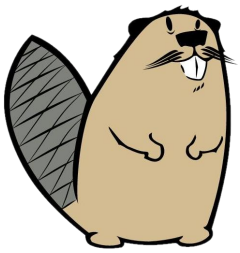
A feladatban szereplő, bemutatókat tartalmazó plakát egy időrend. Az időrendhez tartozó ütemtervek készítése nem könnyű, az informatikában ütemezési problémának nevezik. Az állatkert természetesen azt szeretné, hogy a látogatók minél több bemutatót láthassanak, de bizonyos feltételeket be kell tartani. Például csak akkor lehet bemutatót tartani, ha az állatgondozóknak van idejük, a rendelkezésre álló ketrecek szabadok, és a bemutatók összeegyeztethetők az állatok életritmusával.

Az életben sok hasonló probléma van, amelyekre hasonló megfontolások alkalmazhatók. Ilyen például az iskolai órarend összeállítása vagy a filmek elosztása a mozikban. Az órarendek készítése legtöbbször olyan időigényes, hogy még kis példák esetén is gyakran lehetetlen őket papíron kidolgozni. A számítógép processzorainak_is sok feladatot kell végrehajtaniuk és egymás után feldolgozniuk. Azt, hogy melyik processzor mikor mit csinál, az operációs rendszer villámgyorsan és észrevétlenül dolgozza ki. Az ütemezés a számítástechnika egyik nagy problémája, amellyel ma is sok kutatás foglalkozik.

WEBOLDAL

https://people.inf.elte.hu/csgqaai/doksi.html

https://hu.wikipedia.org/wiki/Matematikai_optimaliz%C3%A1l%C3%A1s



NÖVÉNYTÁRSULÁSOK (2023-AU-01b)

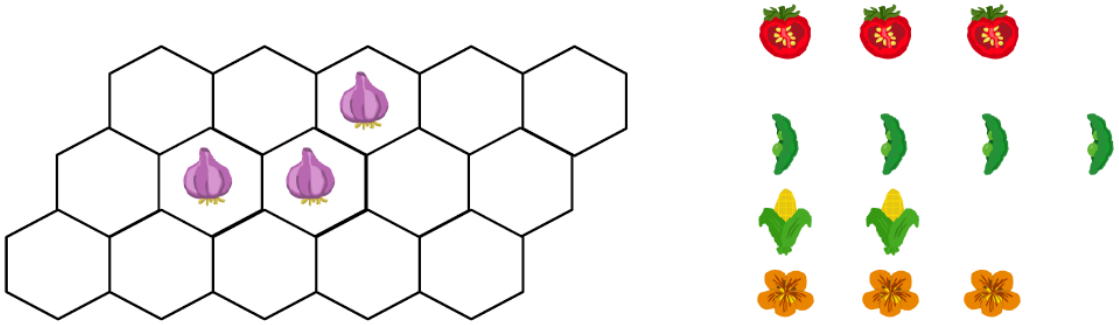
BENJAMIN – KÖZEPES
KADÉT – KÖNNYŰ

Liza zöldségeket szeretne ültetni, mégpedig öt különböző zöldségfélét. Egyes zöldségek ültethetők egymás mellé, míg mások nem.

A képen látható, hogy mely zöldségek ültethetők egymás mellé (♥), és melyek nem (✖).

Liza hatszögletű parcellákra osztotta a teljes ágyást. Minden parcellába pontosan egy fajta növényt akar ültetni, mégpedig úgy, hogy az egymással közvetlenül határos parcellákba nem kerülhetnek olyan fajták, amelyek nem ültethetők egymás mellé.

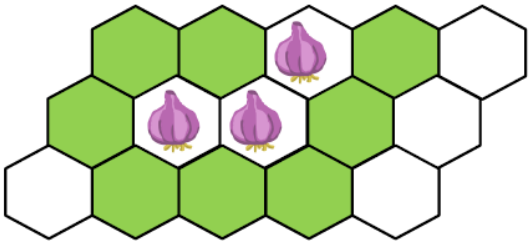
Liza már három parcellába ültetett fokhagymát, az ábrán látható módon.

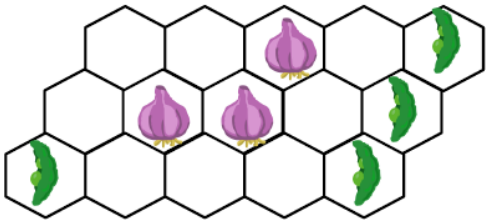


Ültesd be az összes megmaradt szabad parcellát a mellékelt növényekkel, betartva az ültetési szabályokat. Minden parcellába kerüljön egy zöldség és minden zöldséget el kell ültetni.

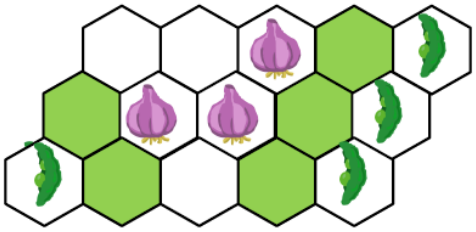
Megoldás

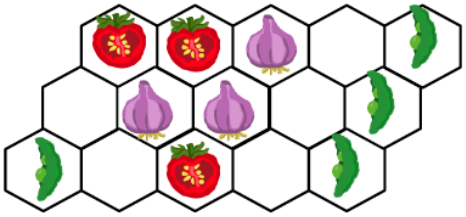
Mivel a bab () nem ültethető fokhagyma () mellé, Liza nem ültethet babot az első ábra színezett parcelláiba. Így csak a szabad parcellák maradnak a bab számára, amit teljesen ki is töltenek.



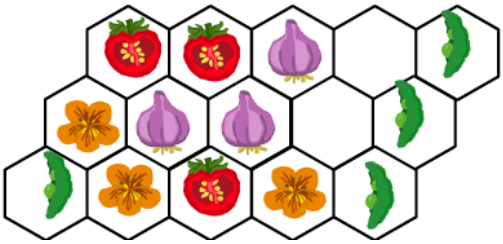


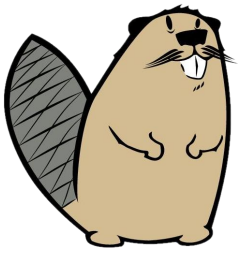
Mivel a paradicsom () nem lehet a babok () mellett, Liza nem ültethet paradicsomot a következő képen berajzolt színes parcellákba, de a három fennmaradó parcellába igen.





Mivel a paradicsom () nem lehet a kukorica () mellett, Liza csak hódsalátát () tehet a paradicsomok mellé. A fennmaradó két parcellában pont elfér a két kukorica, amik így nem kerülnek rossz szomszéd mellé.





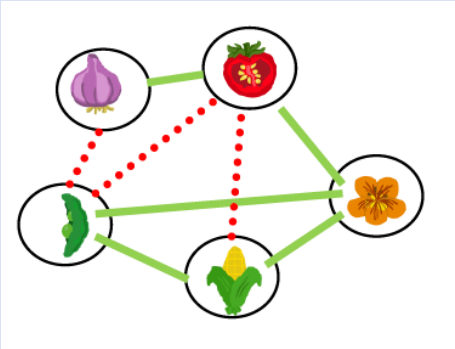
MIÉRT INFORMATIKA?

Ezt a feladatot nagyon sokféleképpen lehet megoldani. Túlságosan időigényes lenne az összes lehetséges megoldást felírni, majd abból választani.

Ebben a feladatban a termények kiválasztásának sorrendje segíthet. Az első lépésben a babok elhelyezése fontos kiindulópont a feladat megoldásához. Ez az első kiválasztás lényegesen lecsökkentette a következő lépések lehetőségeit.

A növények közötti kapcsolatokat az alábbi ábrán látható módon lehet ábrázolni. Két növény

zöld vonallal van összekötve, ha ültetetők egymás mellé, és szaggatott piros vonallal, ha nem.



Az olyan ábrákat, amely csomópontokból (pontokból) és élekből (összekötő vonalakból) áll, gráfoknak nevezzük. A gráfok sokféle probléma szemléltetésére használhatók. Sok esetben egy feladat adatainak és kapcsolatainak gráf formában való megjelenítése nemcsak a probléma szemléltetésére szolgál, hanem fontos lépés a megoldáshoz vezető úton.

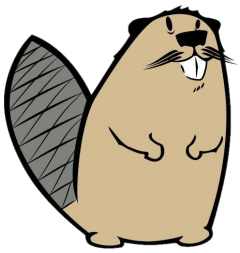
WEBOLDALAK

https://hu.wikipedia.org/wiki/Divide_et_impera_(informatika)

https://hu.wikipedia.org/wiki/Felt%C3%A9teles_utas%C3%ADt%C3%A1s

https://lexiq.hu/dinamikus-programozas

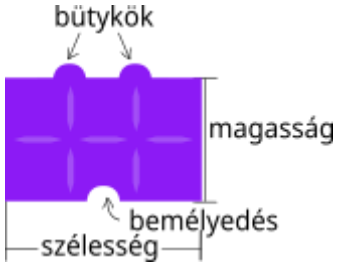
https://www.inf.szte.hu/~rfarkas/Alga17/alga-gyak-04.pdf



HÓDKOCKÁK (2023-AU-05b)

BENJAMIN – NEHÉZ
KADÉT – KÖZEPES
JUNIOR – KÖNNYŰ

A hód építőelemek négy tulajdonságban különböznek egymástól:

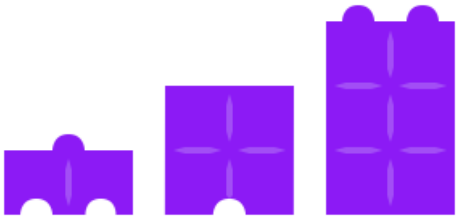


- szélesség: keskeny, közepes, széles
- magasság: kicsi, közepes, nagy
- a tetején található bütykök száma: nulla, egy, kettő.
- az alján található bemélyedések száma: nulla, egy, kettő

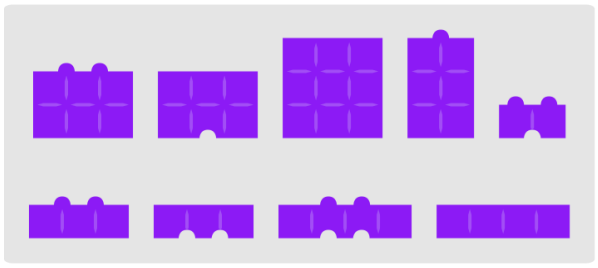
Az építőelemeket hármasával csoportokba szeretnénk osztani úgy, hogy egy csoporton belül minden tulajdonságukban vagy mind azonosak, vagy teljesen különbözőek legyenek.

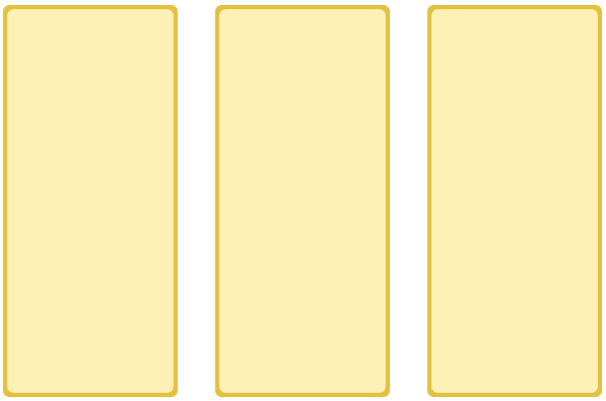
Például ez egy megfelelő csoport, mivel a három építőelem mindegyike

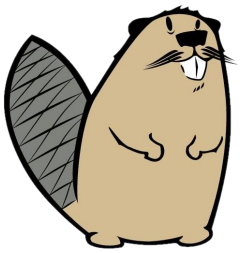
- azonos szélességű,
- különböző magasságú,
- különböző számú bütyök és
- különböző számú bemélyedés található rajta.



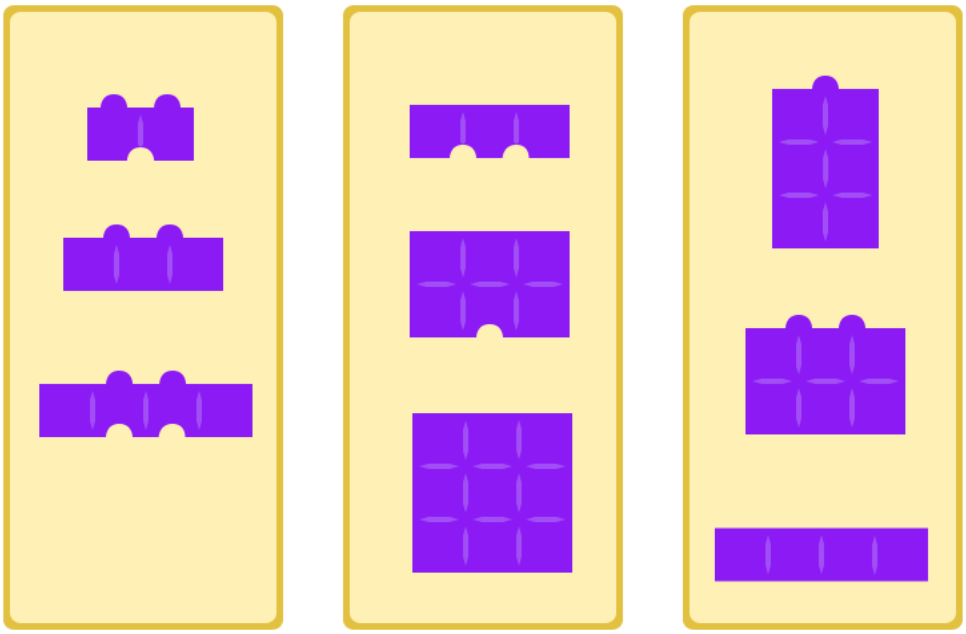
Oszd a 9 építőelemet 3 csoportba úgy, hogy mindegyik csoportba három építőelem kerüljön és megfeleljenek a szabálynak.







Megoldás



A táblázat a három csoport építőelemein mutatja, hogy mely tulajdonságok esetében különböznek, illetve azonosak:

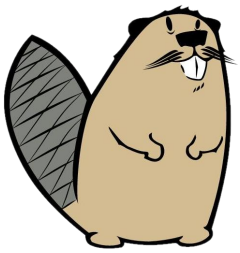
Tulajdonság	1. csoport	2. csoport	3. csoport
szélesség	különböző	azonos	különböző
magasság	azonos	különböző	különböző
bütykök száma	egyenlő	egyenlő	különböző
bemélyedések száma	különböző	különböző	azonos

De vajon ez az egyetlen megoldás?

Megfontolandó: Ha egy tulajdonság esetében az értékek minden csoportban különbözőek, a különböző értékeknek pontosan annyiszor kell előfordulniuk az összes építőelemben, ahány csoport van.

Ha ez nem így van, akkor legalább egy olyan csoportnak kell lennie, amelyben az adott tulajdonság értékei mind megegyeznek.

Az összes építőelemet közelebbről megvizsgálva látható, hogy a szélesség esetében a keskeny és a széles értékek csak kétszer fordulnak elő. Tehát kell lennie egy olyan csoportnak, amelyben minden építőelem „széles" szélességű. Az öt széles építőelem közül egy sincs, amelyiknek csak egy bütyke van, ezért nem lehet különböző számú bütyköket tartalmazó csoportot alkotni. Van azonban három olyan építőelem, amelynek nincs bütyke- és mindegyik különböző magasságú és



különböző számú bemélyedéssel rendelkezik. Ezáltal az egyik csoportot egyértelműen létrehoztuk és csak így hozható létre.

A másik két csoportban a szélességeknek mind különbözőnek kell lenniük. A fennmaradó hat építőelemnél a nagy és közepes értékek csak egyszer fordulnak elő a magasságban. Tehát kell lennie olyan csoportnak, amelyben minden építőelem magassága kicsi. Az 1. csoport az egyetlen lehetséges három építőelemből álló csoport, amelynek magassága kicsi, és amely megfelel az elképzeléseinknek. Így marad a 3. csoportba tartozó három építőelem. Ezek is egy hármas csoportot alkotnak.

MIÉRT INFORMATIKA?

Ebben a Hód feladatban a Hód építőelemeket négy tulajdonság segítségével írjuk le.

Ahhoz, hogy az építőelemeket hármas csoportokra lehessen osztani, ismerni kell az egyes elemek (objektumok) tulajdonságainak értékeit.

Egy számítógépes program, amelynek feladata a hármas csoportokat összerakni, általában nem látja ezeket a tulajdonságokat, és szüksége van egy adatszerkezetben lévő leírásra. Például leírhatjuk az elemeket egy adatbázisban egy táblázat soraival. A táblázat oszlopai megfelelnek a tulajdonságoknak, és a sor tartalmazza egy elem (objektum) értékeit a megfelelő oszlopokban:

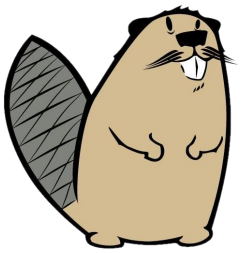
Az adatbázis-táblák tervezése az informatikusok gyakori tevékenysége.

Alaposnak kell lenniük, és figyelembe kell venniük, hogy az objektumok mely tulajdonságai fontosak a számítógépes program feldolgozási szempontjából. A tervezés során azt is figyelembe szokták venni, melyik tulajdonságot melyikkel kell legtöbbször együtt lekérdezni – és a lekérdezés bonyolultságához, idejéhez is optimalizálják a tárolást

WEBOLDALAK

https://www.setgame.com/sites/default/files/instructions/SET%20INSTRUCTIONS%20-%20HUNGARIAN_0.pdf

https://hu.wikipedia.org/wiki/Ramsey-t%C3%A9tel



EMMA TENNIVALÓI (2023-BE-01)

JUNIOR – NEHÉZ
SENIOR – KÖZEPES

Emma otthon van. Három feladatot kell elvégeznie, majd hazatérnie:

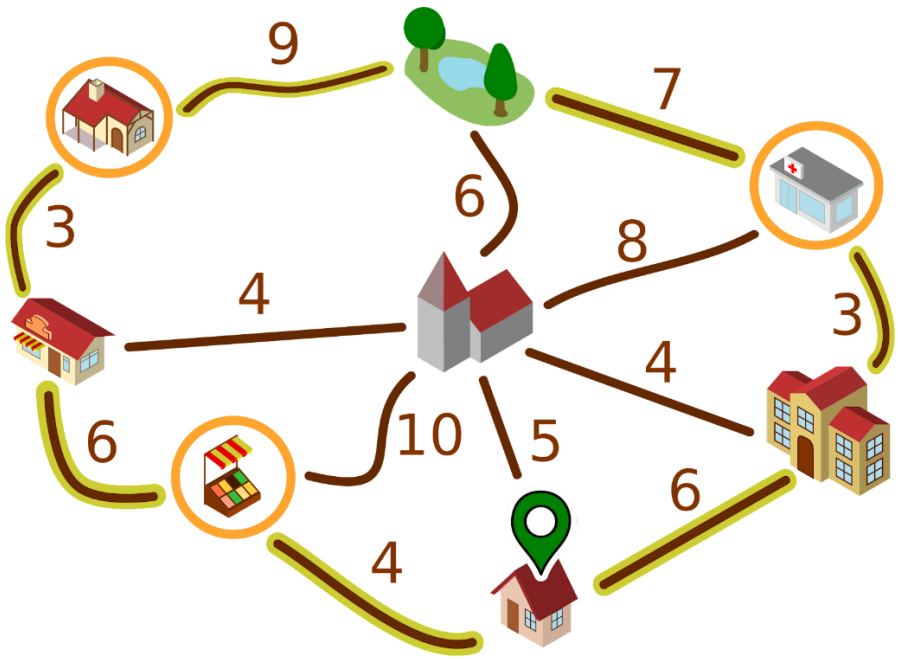
- átvenni egy csomagot a kisboltban ,
- gyümölcsöt venni a piacon , és
- elhozni a gyógyszert a gyógyszertárból .

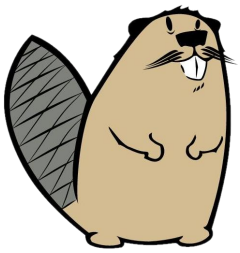
Emma nem tudja, mennyi időt tölt majd az egyes helyeken, de szeretné, ha legalább az útja a lehető legrövidebb ideig tartana.

Fel is írta egy térképre, hogy a hány perc alatt jut el az egyes helyekről a másikra. Bejelölt egy lehetséges utat, ami 4 + 6 + 3 + 9 + 7 + 3 + 6 = 38 percbe telik majd.

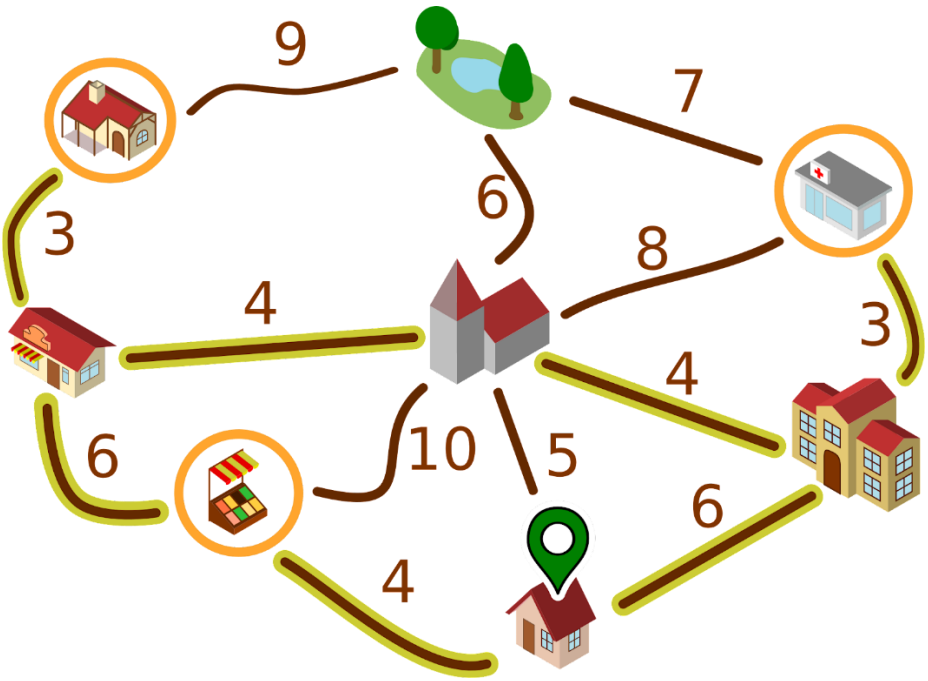
Azonban eszébe jutott, hogy talán gyorsabban is végezhetne, ha az úton néhány útszakaszt kétszer használna (oda-vissza is azon menne).

Módosítsd az első tervét és határozd meg a legrövidebb utat, amit Emma választhat!

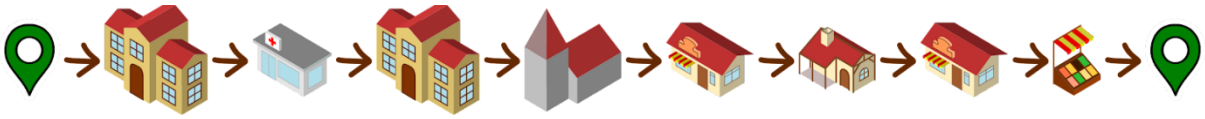




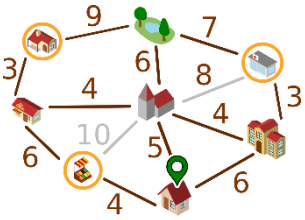
Megoldás



Emma a következő útszakaszokkal teheti meg leggyorsabban az utat (illetve ugyanezt az ellenkező irányban):

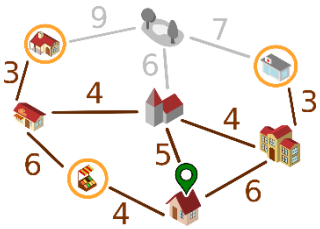


Ehhez összesen 6 + 3 + 3 + 4 + 4 + 3 + 3 + 6 + 4 = 36 percre van szüksége.



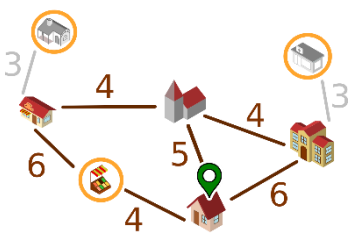
Nézzük meg, miért nem találhat rövidebb utat!

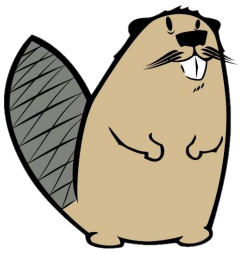
A szürkével átszínezett 8 és 10 percig tartó utakat eltávolíthatjuk, hiszen találunk a piac és a gyógyszertár (valamint Emma otthona) között rövidebb utat.

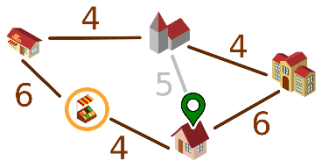


A parkba vezető utakat is elhagyhatjuk, hiszen a parkba nem kell eljutni, és az oda, illetve azon át vezető utaknál van gyorsabb lehetőség az egyes helyek között.

Emmának el kell mennie a patikába  és a kisboltba  , ahova és ahonnan vissza kell térnie a pékséghez  illetve az iskolához  .
Ez mindkét esetben 3+3=6, azaz összesen 12 percébe kerül.





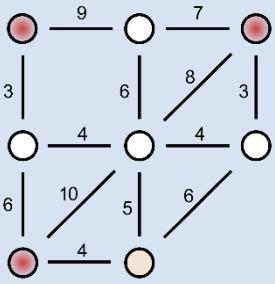


Az útvonal kezdete és vége Emma otthona, ahonnan három helyre (pékség, iskola és piac) kell eljutnia a cél teljesítéséhez. Az ezt teljesítő legrövidebb útvonal a szürke út kivételével minden útszakaszon keresztül vezet, és 4 + 6 + 4 + 6 = 24 percet vesz igénybe. Itt az egyes szakaszok többszöri megtétele sem rövidíti le az időt a körúthoz képest.

A fenti 12 perccel együtt ez 36 percet jelent. Az előző megfontolások azt mutatják, hogy ennél rövidebb útvonal nem létezik.

MIÉRT INFORMATIKA?

A helyes válasz indoklására a terv egyszerűsített ábrázolása szolgált. A tervet sokkal absztraktabb (elvontabb) módon is be lehetett volna mutatni.



Ez a reprezentáció tartalmazza az Emma útja szempontjából fontos információkat, nevezetesen:

- objektumokat: a helyek, az útvonal szempontjából fontos helyek külön megjelölve; és
- az objektumok közötti kapcsolatokat: a helyek közötti távolságokat, amelyek mindegyikéhez meg van adva a hossz.

Az objektumok közötti kapcsolatok modellezésének fontos eszközei a gráfok. A gráfok csomópontokból (az objektumok) és élekből (objektumpárok, a kapcsolatok) állnak. Emma terve súlyozott gráfként modellezhető, ahol az egyes kapcsolatoknak számértékeket (súlyokat) adunk.

Az informatikusokat a gráfokról feltehető kérdések és a kérdések megválaszolására használható algoritmusok érdeklik. A súlyozott gráfok egyik fontos kérdése a következő: Mi a legrövidebb (vagy leggyorsabb) út két csomópont között? A "gráf kérdése" ebben a Hód feladatban hasonló: Mi a legrövidebb körút egy csomópontból, amely meglátogat egy sor más csomópontot? Az informatika számos olyan algoritmust ismer, amelyek hatékonyan képesek meghatározni a legrövidebb utakat gráfokban. Ilyen algoritmusokat használnak például útvonaltervezésre szolgáló alkalmazások, a kiszállító cégek a kiszállítási útvonalat optimalizálva.

WEBOLDALAK

https://www.cs.bme.hu/~kiskat/algel/alg3-handout.pdf

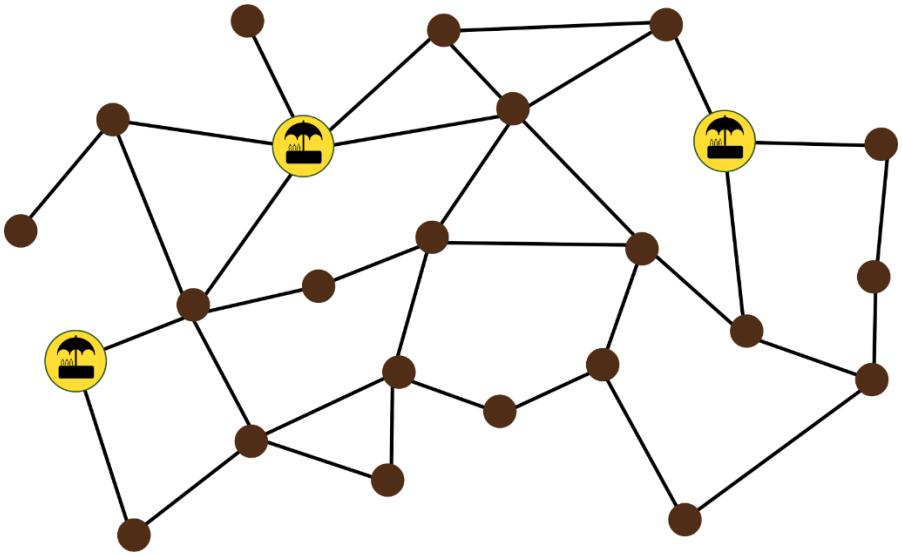
https://hu.wikipedia.org/wiki/Gr%C3%A1felm%C3%A9let

KUTAK (2023-BR-05)

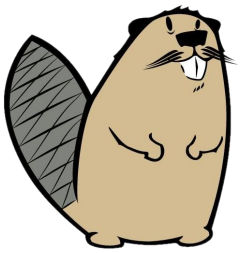
KADÉT – NEHÉZ
JUNIOR – KÖZEPES
SENIOR – KÖNNYŰ

A nyár forró a városban. A polgármester ezért szökőkutakat állíttat fel, mégpedig úgy, hogy minden utcasarokról legfeljebb két utcaszakaszt kelljen gyalogolni ahhoz, hogy valaki elérjen egy szökőkutat.

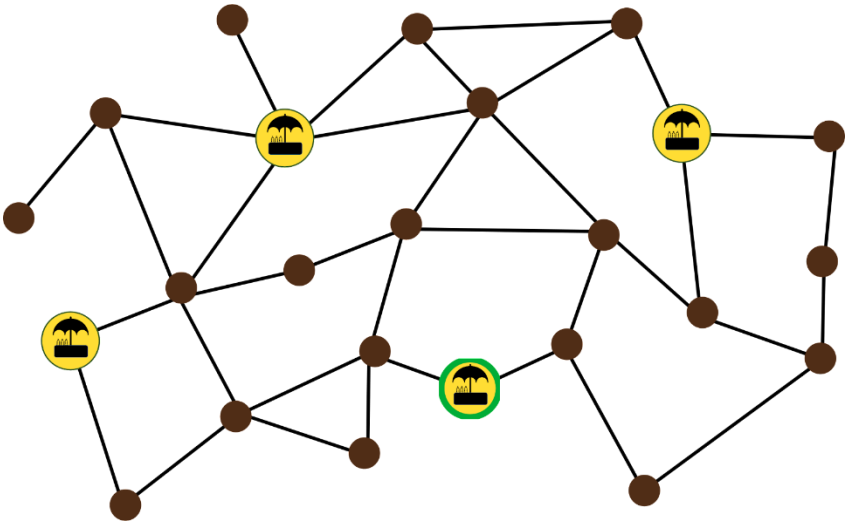
Itt van a város térképe. A vonalak az utcaszakaszok, a pontok pedig az utcasarkok. Három sarkon már van szökőkút .



Állíts fel még egy szökőkutat, hogy a polgármester célja teljesüljön!

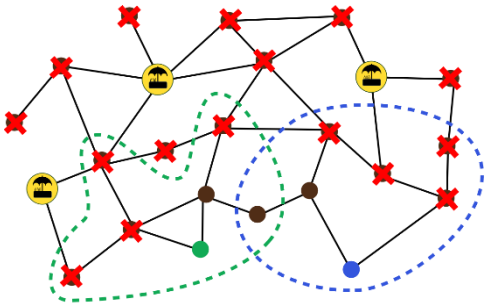


Megoldás

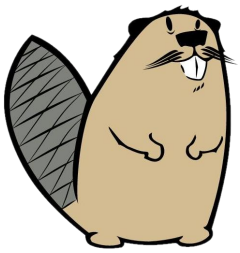


Ezzel az új szökőkúttal minden utcasarokról legfeljebb két utcaszakaszt kell gyalogolni ahhoz, hogy elérjünk egy szökőkutat. Tehát a polgármester célja megvalósult.

Hogyan lehet kitalálni, hogy melyik utcasarkon kell elhelyezni ezt az új szökőkutat? A várostérképen jelöljük be az összes olyan utcasarkot (a képen egy keresztet használunk erre), amely legfeljebb két utcaszakaszra van a már felállított kutak egyikétől. Ezeknél a sarkoknál már teljesül a cél.



A kimaradt öt utcasaroktól is legfeljebb két utcaszakaszt szeretnénk gyalogolni a következő kútig. Ha a két legalsó és egymástól több, mint 2 útszakasz távolságra lévő sarok (zöld vagy kék pont) esetében megkeressük az összes többi olyan sarkot, ahonnan két utcaszakaszon keresztül lehet eljutni (az ábrán szaggatott vonalakkal körbekerítve), akkor a középső alsó utcasarok az egyetlen, amely teljesíti ezt a feltételt.



MIÉRT INFORMATIKA?

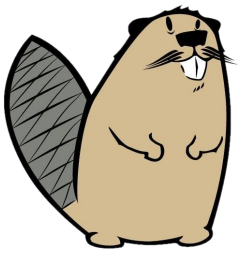
A várostérképet gráfként lehet modellezni. Ez az informatikában gyakran használt adatábrázolási mód az objektumok közötti kapcsolatok modellezésére és az ezekkel a kapcsolatokkal kapcsolatos kérdések megválaszolására. Itt az utcasarkok objektumoknak és így a gráf csomópontjainak tekinthetők. A kapcsolat, amelyet ezután a gráf éleiként modellezünk, a sarkok szomszédsága, amelyeket egy utcaszakasz köt össze.

Ebben a Hód feladatban a csomópontok egy olyan részhalmazát kell megtalálni (a szökőkutak felállításához), hogy az ezen a részhalmazon kívüli minden csomópont egy legfeljebb két él hosszúságú "kútcsomóponthoz" vezető úton keresztül kapcsolódjon. Az informatikai terminológiában ezt "k-dominál halmaz"-nak nevezzük.

Az ilyen "minimális távolságú k-dominál halmazok" az utóbbi időben egyre nagyobb szerepet játszanak. Pl: A közösségi hálózatok adatainak automatikus feldolgozásakor (például az álhírek terjedésének felderítéséhez) a felhasználók közötti rajongói vagy követői kapcsolatokat gráfként modellezik. Ezek a gráfok olyan nagyok lehetnek, hogy a felhasználóknak csak egy reprezentatív (minél kisebb) kiválasztását lehet figyelembe venni – például egy "3-dominál halmazt".

WEBOLDAL

https://hu.wikipedia.org/wiki/Gr%C3%A1fhatv%C3%A1ny



RICCA (2023-CA-02)

KISHÓD – NEHÉZ
BENJAMIN – KÖZEPES
KADÉT – KÖNNYŰ

Evelin most tanulja meg, hogyan néz ki egy ricca. Ehhez öt riccáról készült képe van, és ezek alapján állításokat fogalmaz meg arról, hogyan néznek ki a riccák.

A barátja, Laci most egy hatodik képet mutat neki egy riccáról. Evelin ekkor észreveszi, hogy az új kép alapján az egyik állítása határozottan rossz.

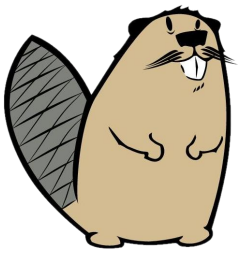
Melyik ez az állítás?

A) Minden riccának van foga.

B) Néhány riccának szárnya van.

C) A riccáknak vagy szarvuk van, vagy három szemük, de soha nem mindkettő.

D) Ha a riccáknak pontosan két karjuk van, akkor pontosan két lábuk is van.



A helyes válasz a D)

Az A) válasz olyan állítást fogalmaz meg, amely minden riccára érvényes. Ha akár csak egy riccának nem lenne foga, akkor ez hamis lenne. Azonban az összes általunk ismert riccának van foga, így Evelin mondata nem lehet egyértelműen hamis.

A B) válasz olyan állítást fogalmaz meg, amely csak néhány riccára igaz. Egy riccának van szárnya. De akkor sem mondhatjuk biztosan, hogy az állítás hamis, ha a nekünk bemutatott riccák közül egyiknek sem volt szárnya. Tehát ez a mondat sem lehet egyértelműen hamis.

A C) válasz két állítást kapcsol össze "vagy"-"vagy" szavakkal, és pontosan akkor lesz igaz, ha a két állítás közül csak az egyik igaz. Ez mind a hat kép esetében így van: négy riccának szarva van, de három szeme nincs, a másik két riccának nincs szarva, de három szeme van. Nincs olyan ricca a hat közül, aminek három szeme és szarva is lenne egyszerre, vagy se szarva, se három szeme nem lenne. Tehát az állítás nem egyértelműen hamis.

Marad a D) válasz. Ez egy "ha"-"akkor" állítás formájában van megfogalmazva. Ha a „ha” feltétel igaz, akkor az „akkor”-kijelentésnek is igaznak kell lennie. A feltétel igaz az összes riccára a képeken. Az első öt képen az „akkor” kijelentés is igaz (minden riccának is két lába van). Tehát Evelin (az akkori tudása szerint) helyes állítást írt. Azonban a Laci által adott képen a riccának kettőnél több lába van, nevezetesen öt. Ezért Evelin állítása a D) válaszban határozottan hamis.

MIÉRT INFORMATIKA?

A szárnyak, karok, lábak és szemek száma, valamint az, hogy a riccáknak vannak-e fogaik vagy szárnyaik a riccák tulajdonságai. Amikor leírjuk a riccákat, akkor állításokat fogalmazunk meg ezekről a tulajdonságokról. Ez egy modellhez vezet arról, hogy mik a riccák.

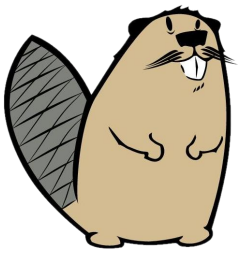
A számítógépek is modellekkel dolgoznak. Néhányat egyértelműen megfogalmaznak a programozók, informatikusok, mint például a név, a születési dátum és a lakcím egy nyilvántartásban.

De van, amikor a számítógép maga kell, hogy felállítsa a megfelelő modellt, „megtanulja” hogyan néz ki valami, milyen tulajdonságokkal rendelkezik. Ilyenkor egy lehetséges tanulási folyamat az, hogy pl. képeket kap, amelyeket össze kell hasonlítania, és létre kell hoznia az állításokat. Egy-egy újabb képnél pedig fel kell ismernie, melyik állítása nem igaz és azt módosítania, kiegészítenie kell, hogy pontosabb legyen a modell, amit létrehozott.

WEBOLDALAK

https://en.wikipedia.org/wiki/Pattern_recognition

https://hu.wikipedia.org/wiki/Logikai_m%C5%B1velet



ESERNYŐ (2023-CH-01)

KISHÓD – KÖNNYŰ

Ez Anna esernyője:

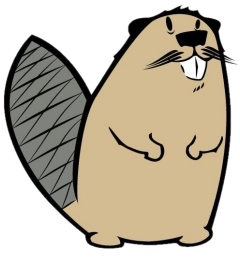
A négy kép közül az egyik Anna esernyőjét mutatja oldalról. Melyik az?

A

B

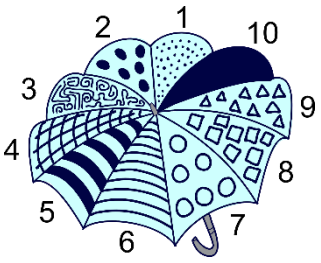
C

D



A helyes válasz a C)

Anna esernyőjén minden minta pontosan egyszer fordul elő.

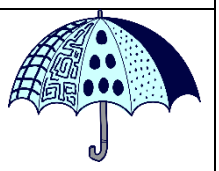
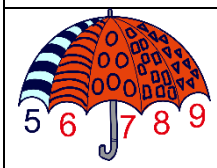
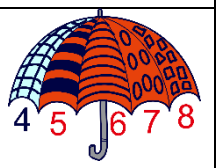


Ahhoz, hogy megtaláljuk a helyes képet, sorban összehasonlítjuk az egyes képeket Anna esernyőjével:

- kiválasztjuk azt a mintát, amelyik balra legtávolabb van, és megkeressük a helyét Anna esernyőjén.
- ellenőrizzük, hogy a szomszédos minták megegyeznek-e az Anna esernyőjén lévővel.

A négy kép mindegyike csak öt minta sorozatát mutatja, nem mind a tízét. Teljesen biztosan tehát nem tudhatjuk, hogy a négy kép bármelyikének mintasorozata teljesen megegyezik-e Anna esernyőjének mind a tíz mintájának teljes sorozatával.

A C kép az egyetlen, amelynek az öt mintából álló mintasorozata teljesen megegyezik Anna esernyőjének mintasorozatával. Emiatt csak a C kép mutathatja Anna esernyőjét. Az összes többi képen a minták sorozata nem, vagy csak részben egyezik meg Anna esernyőjének mintáival. Tehát ezeken a képeken nem lehet Anna esernyője.

A	B	C	D
			
			

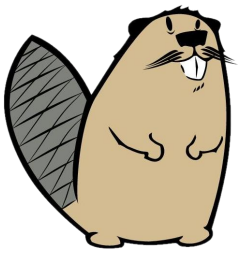
MIÉRT INFORMATIKA?

A válaszlehetőségek mindegyikében a mintasorozatnak csak egy része látható. Bár csak részleges információt tartalmaznak, meg tudjuk határozni, hogy a négy kép közül melyik mutatja Anna esernyőjét: Egy kép csak akkor mutatja Anna esernyőjét, ha a mintasorozat teljes egészében előfordul Anna esernyőjének mintasorrendjében.

Az "esernyőminta-keresés" elvét alkalmazzuk a szöveges dokumentumban való keresésnél is. A számítógép a megadott részinformációval (keresőszó) megegyező karakterláncokat keres a dokumentumban. A karaktersorozat karakterek (pl. betűk, számok, speciális karakterek) sorozata.

A keresésre a következők vonatkoznak:

- Minél hosszabb a keresőszó, annál kevesebb a lehetséges találat és annál nagyobb az esélye, hogy pontosan a keresett szót találjuk meg a dokumentumban.
- Minél rövidebb a keresőszó, annál több lehetséges találat van, és annál kevésbé pontos a keresés.

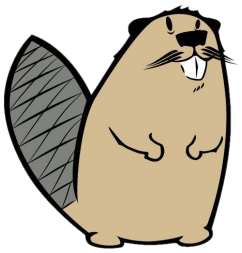


A keresés javítása érdekében különböző keresési eljárásokat dolgoztak ki. Ezek célja, hogy a lehető leggyorsabban pontos keresést végezzenek és megfelelő eredményt adjanak. Ezeket a keresőalgoritmusokat folyamatosan fejlesztik, és ezek ezért nagyon rövid idő alatt hatalmas mennyiségű adatot tudnak átkutatni (pl. az internetes keresőmotorok ilyen keresőalgoritmusokat is használnak).

WEBOLDAL

https://hu.wikipedia.org/wiki/Keres%C5%91algoritmus

https://hu.wikipedia.org/wiki/String

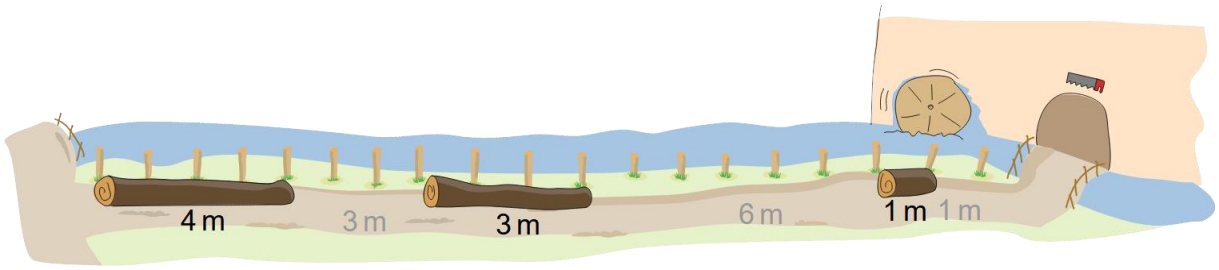


TÁROLÁS (2023-CH-05)

BENJAMIN – NEHÉZ
KADÉT – KÖZEPES
JUNIOR – KÖNNYŰ

Liza, a hód különböző hosszúságú rönköket vág a patakparton. Amikor levág egy rönköt, azonnal lerakja a 18 méter hosszúságú keskeny úton. A rönköket egymás után teszi le. A rönkök lerakásánál a következő szabályt követi: minden rönköt a bal oldalról haladva az első olyan szabad helyre rak le, ahová az adott rönk befér.

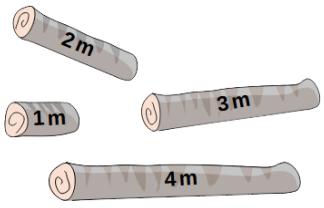
Miután eladott néhány rönköt, és ezeket elvitték az útról, az út így néz ki:



Ezután Liza 1,2,3 és 4 méteres rönköket szeretne levágni.

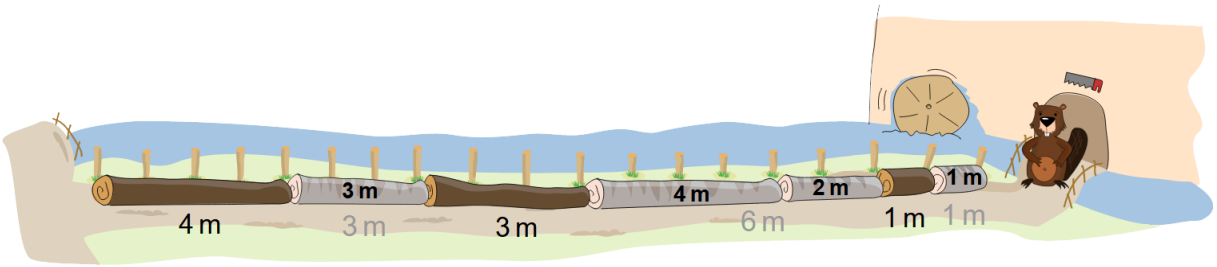
Milyen sorrendben kell levágnia a rönköket, hogy a szabály szerint mindet azonnal az útra tudja tenni?

1. 2. 3. 4.



Megoldás

Ha a rönkök a (3 m, 4 m, 2 m, 1 m) sorrendben vannak, akkor Liza mindet az útra tudja helyezni: A 3 méteres rönk esetében a bal szélső 3 méteres rés az első elérhető rés, amelybe a rönk befér; Liza oda helyezi a rönköt. A 4 méteres rönk ezután a bal oldali 6 méteres résbe kerül. A következő rönk befér a maradék 2 méteres résbe, és Liza az utolsó rönköt az 1 méteres résbe helyezi. Ezután az út így néz ki:



Helyes sorrend még a (3 m, 2 m, 4 m, 1 m) és a (4 m, 3 m, 2 m, 1 m) is.

Minden más sorrend esetében Liza nem tudja a szabálynak megfelelően lerakni az összes rönköt: Az 1 méteres rönknek mindig utolsónak kell lennie a sorban, mert csak ez a rönk töltheti ki az utolsó szabad helyet. A 2 méteres rönk nem kerülhet a 3 méteres rönk elé, mert különben a 3 méteres résbe kerülne, és így egy második 1 méteres rés keletkezne.

Az említett három sorozattól eltekintve nincs olyan sorozat, amely megfelel a feltételeknek.

MIÉRT INFORMATIKA?

Ez a hód feladat a "ládapakolási probléma" optimalizációs feladatkör egyik példája. A ládapakolási problémában különböző méretű tárgyakat kell elhelyezni adott számú tárolóban, amelyek maguk is különböző méretűek lehetnek. A tárgyak itt a fatörzsek, a tárolók pedig az út szabad helyei.

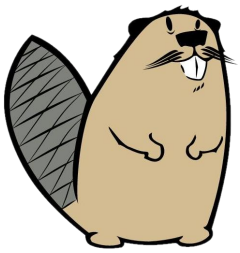
A ládapakolási probléma nagyon különböző helyeken fordulhat elő:

a) Egy bútorraktárban kis és nagyméretű bútorokat kell helytakarékosan tárolni.

b) Egy fuvarozó cég pénzt takaríthat meg, ha okos csomagolással kevesebb teherautóra van szüksége az áruk szállításához.

c) Egy számítógép operációs rendszerének különböző méretű fájlokat kell tárolnia a merevlemezen. Amikor fájlokat törölnek, a merevlemezen újrahasználható részek keletkeznek. Ezeket a réseket be kell tölteni, hogy ne vesszen kárba a tárhely, hasonlóan az ebben a hód feladatban szereplő úton lévő résekhez.

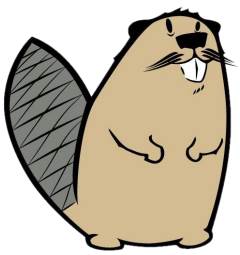
Az informatikában a ládapakolási problémát a legnehezebb problémák között tartják számon; garantáltan optimális megoldásokat csak kis esetekre tudnak számítógépes programokkal megoldani. Vannak azonban különböző eljárások és stratégiák, amelyekkel a ládapakolási probléma megoldásai könnyebben meghatározhatók. Feladatunkban segítséget Liza szabálya adja: minden rönköt mindig a balról első olyan pozícióba helyez, ahová az elfér. Ezt a stratégiát "első illeszkedésnek" nevezzük. A feladatban láthatjuk, hogy ez a megközelítés rossz eredményhez is vezethet. Csak akkor lehet az összes szabad helyet betölteni, ha a rönköket egy bizonyos sorrendben illesztjük be.



WEBOLDALAK

https://vik.wiki/images/e/ed/Opre_16_memory.pdf

https://hu.wikipedia.org/wiki/F%C3%A1jlrendszer-t%C3%B6redezetts%C3%A9g

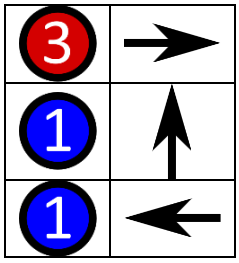


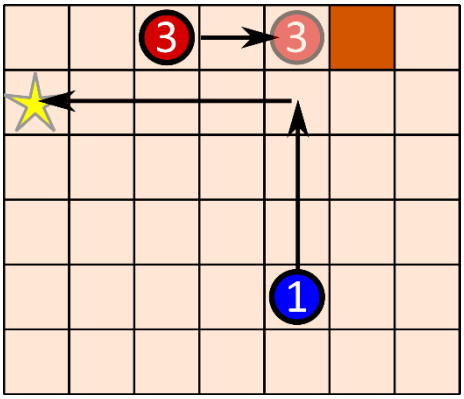
ÜTKÖZŐ ROBOT (2023-CZ-01)

KADÉT – NEHÉZ
JUNIOR – KÖZEPES
SENIOR – KÖNNYŰ

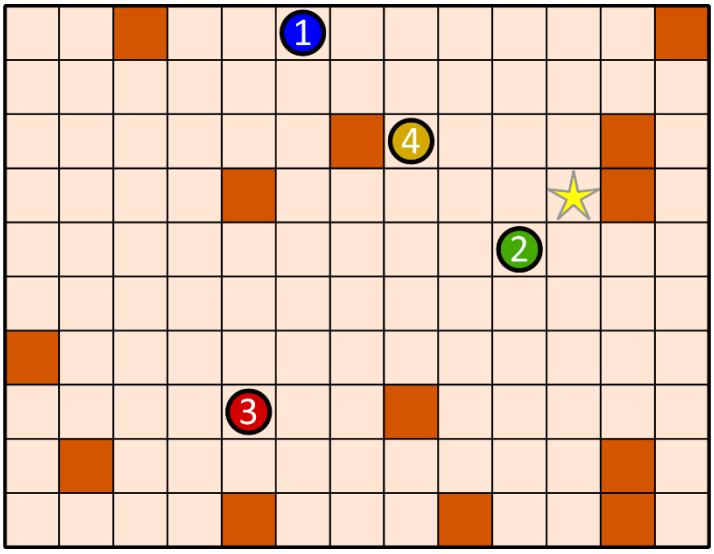
Egy robotot (1, 2, 3, ...)a következő utasításokkal tudunk irányítani: fel, le, balra, jobbra (↑ ↓ ← →). Mikor elindul egy irányba, akkor egyenesen halad előre addig, amíg el nem ér egy akadályt (■) vagy egy másik robotot. Ezután megáll, amíg új utasítást nem kap.

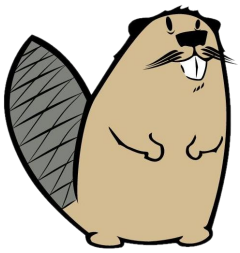
Az utasítások ügyes elrendezésével biztosíthatod, hogy egy robott elérje a csillagot ★. Ebben a példában ezt a következő 3 utasítással teheted meg:



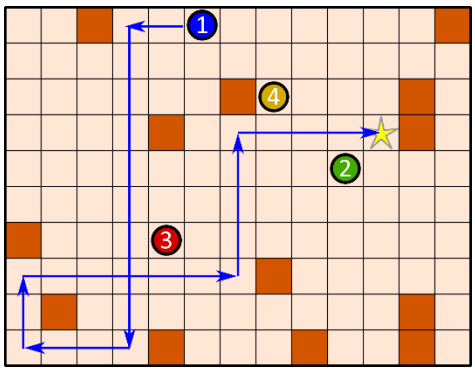


A következő táblán **juttasd el a kék robotot (1) a csillaghoz (★) a lehető legkevesebb lépéssel!**





Megoldás

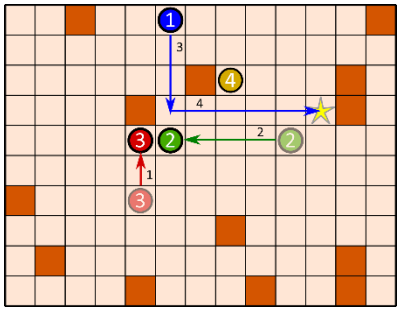


4 lépéssel megoldható a feladat.

A következő képeken minden egyes robotmozgást egy nyíl ábrázol.

Ha a kék robot (1) egyedül akarja elérni a célt, akkor hét parancsra van szüksége:

A mindössze négy lépésből álló megoldásban három robot együttműködik.



Először a 3-as robot (3) megy felfelé az akadályig, majd a 2-es robot (2) megy balra a 3-as robotig. Ezután az 1-es robot (1) megy lefelé, amíg nem találkozik a 2-es robottal. Végül ismét az 1-es robot megy jobbra egészen a csillagig.

Hogyan találjuk meg a megfelelő utasítássorozatot?

Kezdhetjük hátulról, és végig gondolhatjuk, hogy mi lehet kék robot (1) utolsó mozgása a cél felé. Három lehetőség van:

a) balról érkezik, mint a mi megoldásunkban. Ehhez kell egy akadály (a 2-es robot) ahhoz, hogy jó sorban álljon meg. De ennek a robotnak is meg kell állnia, tehát előbb a 3-as robotot fel kell küldeni az akadályig.

Fentről vagy lentről jön. Ebben az esetben a csillag alá vagy fölé kell egy akadályt (robotot) mozgatni.

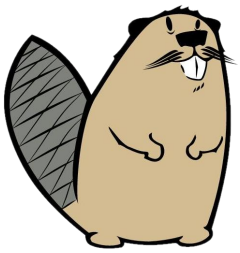
b) a 4-es robot könnyen a csillag fölé mozgatható (1 lépésben), de az 1-es robotnak 5 lépésre van szüksége ahhoz, hogy alulról elérjen a csillaghoz.

c) a csillag alá akadályt (robotot) mozgatni szintén több lépésből oldható meg, ráadásul az 1-es robot beállítása a megfelelő „oszlopba” is több lépést jelent.

MIÉRT INFORMATIKA?

Ebben a hód feladatban több robot dolgozott együtt egy cél elérése érdekében. Különböző feladatokat kaptak. A kék robotnak kellett elérnie a célt, a többiek pedig akadályként szolgáltak az irányváltáshoz.

A feladatmegosztás fontos szempont a robotikában. Például egy automatizált raktárban a különböző robotok együtt dolgoznak az áruk tárolásán, kiemelésén és szállításán. Minden tevékenységet úgy koordinálnak, hogy a lehető legkevesebb haszontalan üresjárat keletkezzen, minden szállítási útvonal a lehető legrövidebb legyen, kevés energia kerüljön felhasználásra és így a raktár összességében a lehető leghatékonyabban működjön. A rajrobotok a robotika egy speciális területét jelentik. Ezek – a mi feladatbeli robotjainkhoz hasonlóan – egyszerű gépek, amelyek egy feladatot nagy csoportban, közösen oldanak meg. A mezőgazdaságban ma már

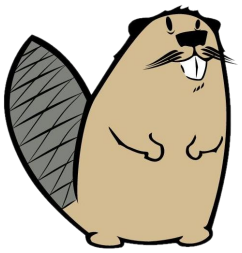


robotrajok képesek gondoskodni a kukorica vetéséről, a növények fejlődésének és a talaj állapotának megfigyeléséről, végül pedig akár a gabona betakarításáról is. Az egyes rajrobotok kicsik és egyszerű kialakításúak, de a raj egésze nagy dolgokra képes. Ez az elv érvényes a multiágens rendszerekre is, amelyek egyszerű szoftveregységek, amelyek együttesen képesek összetett problémák megoldására. Az informatika feladata az optimális koordináció és együttműködés algoritmusainak kifejlesztése.

WEBOLDALAK

https://doosanrobotics.hu/mi-is-pontosan-a-cobot/

https://www.inf.u-szeged.hu/~csendes/Optimalizalas2.pdf



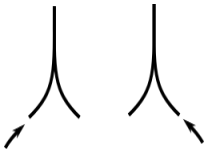
MODELLVASÚT (2023-CZ-02)

BENJAMIN – NEHÉZ
KADÉT – KÖZEPES
JUNIOR- KÖNNYŰ

Gabi épített egy modellvasutat egy főpályaudvarral  és 4 másik állomással.

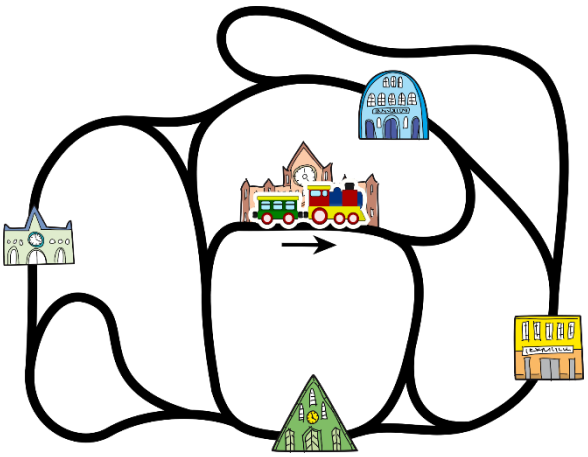
A vonat programozható (képes utasítássorozatot végrehajtani) és az utasításoknak megfelelően átkapcsolja a váltókat. Az összes utasítás végrehajtása után a legközelebbi állomáson pedig megáll.

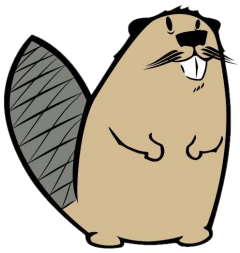
Az utasítások csak **B** és **J** betűk lehetnek. Amikor a vonat elér egy vasúti váltót, az alábbi táblázat alapján választja ki, hogy merre menjen:

Vasúti váltó	Utasítás	Továbbhaladás iránya
	B	
	J	
	A következő utasítás nem itt kerül végrehajtásra	

Gabi úgy helyezi el a vonatot a főpályaudvaron  , hogy a vonat előre nézzen a képen látható irányba.

Ha megadja a **JB** utasítást, akkor a vonat a következő állomáson áll meg: 

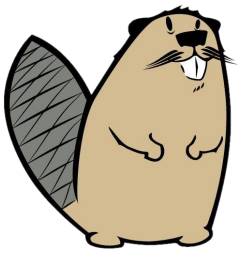




Ha Gabi visszahelyezi a vonatot a főpályaudvarra  a képen látható irányba, és megadja a **BJ** utasításokat, akkor a vonat most is azon az állomáson  áll meg, mint korábban.

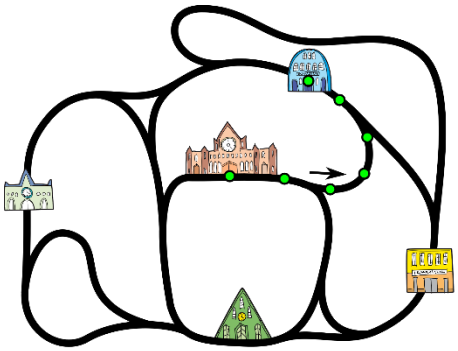
Melyik állomáson áll meg a vonat az BBJJ utasítások végrehajtása után, ha az a főpályaudvarról  indul és a képen látható irányba néz?

A)  B)  C)  D) 

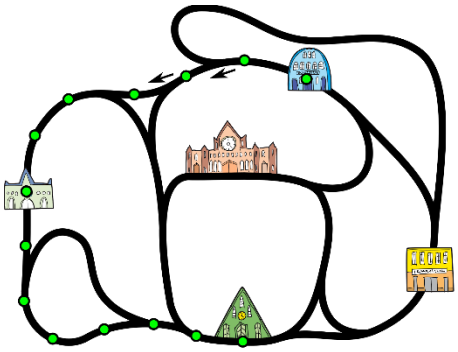


A helyes megoldás a B)  állomás.

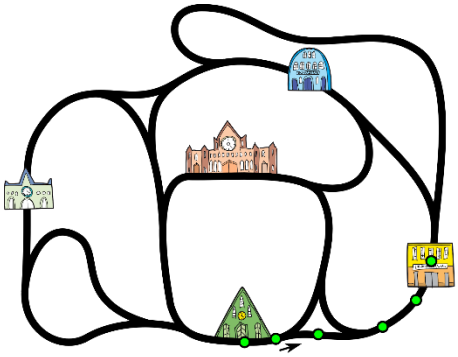
A **B** utasítás végrehajtása után a vonat áthalad az A)  állomáson.

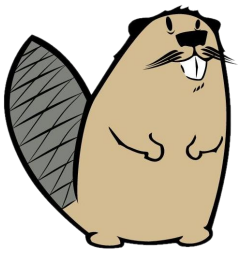


A **BJ** utasítások után a vonat áthalad a D)  állomáson. Ezek után a vonat továbbhalad a C)  állomás irányába, hiszen a fennmaradó utasítássorozat utolsó tagja nem kerül végrehajtásra, mivel csak olyan elágazás van, ahol nem kell a következő utasítást végrehajtani.



A következő **J** utasítás végrehajtása után a B)  állomásra megy. Mivel nincs további utasításunk, a vonat ezen az állomáson megáll.





MIÉRT INFORMATIKA?

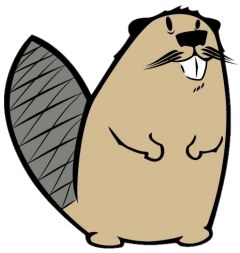
A program, utasítások sorozata. Az utasítások és bizonyos feltételek (a vasúti váltó megközelítési iránya) együttesen határozzák meg a vonat útvonalát. Az ilyen utasításoknak egyértelműnek és érthetőnek kell lenniük a végrehajtó számára a hibák elkerülése érdekében.

Az automatikus vasútirányítás arra épül, hogy a feltételek és az utasítások tárolva vannak, ezért az útvonal később rekonstruálható.

WEBOLDAL

https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%B3g%C3%A9p-programoz%C3%A1s

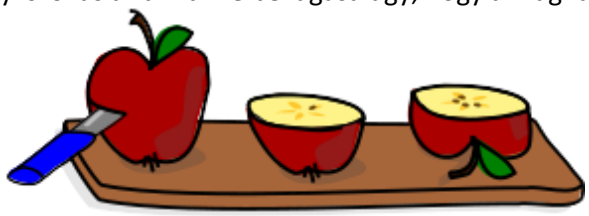
https://hu.wikipedia.org/wiki/Felt%C3%A9teles_utas%C3%ADt%C3%A1s



ALMASZELETEK (2023-CZ-03)

KISHÓD – KÖNNYŰ

Csehországban karácsonyi szokás az almák félbevágása úgy, hogy a magház csillag formájú legyen.

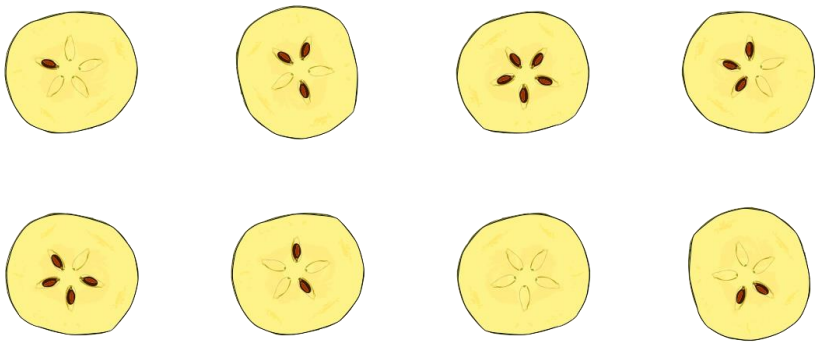


Egy családban négy almát vágnak félbe. Mind a nyolc almafelet összekeverve leteszik az asztalra.

Amikor kettévágták az almákat, néhány mag a magház egyik felében maradt, néhány pedig a másikban.

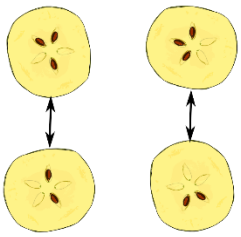
A nyolc almafélből melyek illenek össze? Párosítsd össze az almafeleket!

Megoldás



Két egyforma almafél magházainak száma mindig öt. De nem elég minden esetben megszámolni a magokat.

Két almafélnek egyenként három magja van, és két almafélnek egyenként két magja van. Az azonos számú maggal rendelkező almafeleket a teli és üres magházak mintázata alapján lehet megkülönböztetni. A kép jobb oldalán látható almafelekben az összes teli magház egymás mellett helyezkedik el, közöttük nincsenek üres magházak. A bal oldali felekben egy teli magház külön áll, mindkét oldalán üres magházakkal.



MIÉRT INFORMATIKA?

Ebben a hód feladatban az informatikának számos különböző területét megtaláljuk.

Az informatikában a valóságot sokszor _gráfok_ segítségével szeretjük modellezni. A gráfokat aztán a számítógépek feldolgozhatják, hogy megoldják a modellhez kapcsolódó problémákat.

Jelentsen a felső 4 almafelen egy fekete csomópont egy teli magházat, és egy él kössön össze két, az almában szomszédos teli magházat.

Jelentsen az alsó 4 almafelen egy fekete csomópont egy üres magházat, és egy él kössön össze két, az almában szomszédos üres magházat.

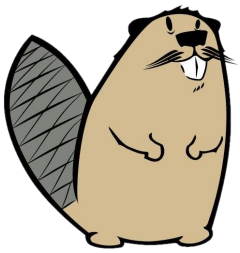
Az egyik felső almafélben csak egy magházban van mag, így azt egy fekete pont jelöli. A hozzá tartozó alsó almafélben csak egy üres magház van, így azt is egy fekete pont jelöli.

Hasonlóan kialakíthatóak a gráfok az egyes felső és alsó almafelekre.

Az egymáshoz tartozó almafelekre ugyanazokat a gráfokat kapjuk, bár lehet, hogy kissé el kell forgatnunk azokat.

Gyakorlati példa erre az arcfelismerés. Erre az egyik megközelítés az, hogy a fülek, az orr, a száj és más jellemzők helyzetét egy gráffal modellezzük. Ez a távolságokat is rögzíti. Amikor összehasonlítjuk Bob referenciaarcát egy magát Bobnak kiadó személy arcával, akkor tulajdonképpen a megfelelő gráfokat hasonlítjuk össze. Minél hasonlóbbak a gráfok, annál nagyobb a valószínűsége annak, hogy Bobot látjuk.

Amikor olyan gráfokat kell összehasonlítanunk, amelyekben a csomópontok pontos helyét nem tudjuk meghatározni, akkor nem azt kérdezzük, hogy a gráfok azonosak-e, hanem azt, hogy izomorfikusak-e. Ennek során azt vizsgáljuk, hogy az egyik gráf csomópontjait (és éleit) el lehet-e



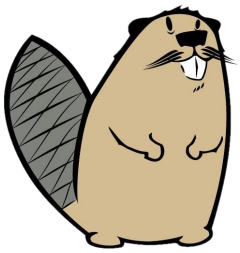
úgy mozgatni, hogy a másik gráfot kapjuk. Egy kis forgatás, mint ebben a feladatban, sajnos ritkán elég. Az a kérdés, hogy egy ilyen átalakítás mindig hatékonyan lehetséges-e, az egyik legnagyobb kihívást jelentő, még nyitott kérdés az informatikában.

WEBOLDAL

https://hu.wikipedia.org/wiki/Gr%C3%A1f

https://hu.wikipedia.org/wiki/Topol%C3%B3gia

https://hu.wikipedia.org/wiki/Arcfelismer%C3%A9s



KONFLIKTUSDETEKTOR (2023-DE-01)

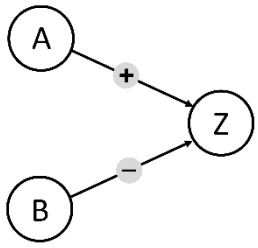
SENIOR – NEHÉZ

Anna és Ben egy "konfliktusdetektort" akarnak építeni, amely megmutatja, ha eltérő véleményük van. Ehhez olyan egységeket használnak, amelyeknek két állapota lehet: `Igen` és `Nem`. Két egység összekapcsolható egy kábellel, amely pozitív (+) vagy negatív (−) jelet küldhet az egyiktől a másikig. Ha egy egység `Igen` állapotban van, akkor jelet küld az összes belőle kimenő kábelen. (Ha `Nem` állapotban van, akkor nem küld jelet.)

Egy egység akkor kerül `Igen` állapotba, ha több pozitív, mint negatív jelet kap. Ellenkező esetben pedig `Nem` állapotba kerül.

Anna az „A" egység állapotát, Ben pedig a „B" egység állapotát állítja be.

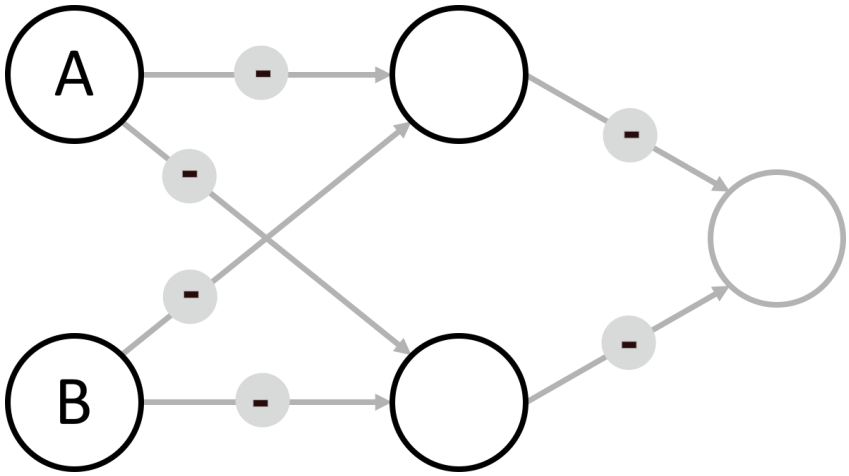
Először Anna és Ben ezt a gépet építette meg:

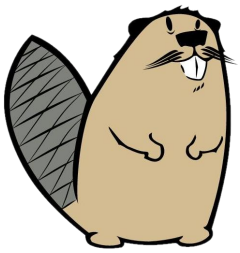


Észrevették, hogy a „Z" egység csak akkor kerül `Igen` állapotba, ha „A" `Igen` és „B" `Nem`. De ők nem ezt akarják.

Ezután megépítik a következő képen látható konfliktusdetektort. „Z" csak akkor lehet `Igen`, ha „A" és „B" különböző állapotban van (`Igen` és `Nem` vagy `Nem` és `Igen`). Ellenkező esetben „Z"-nek a `Nem` állapotban kell lennie. Sajnos a kábelek még nincsenek helyesen beállítva.

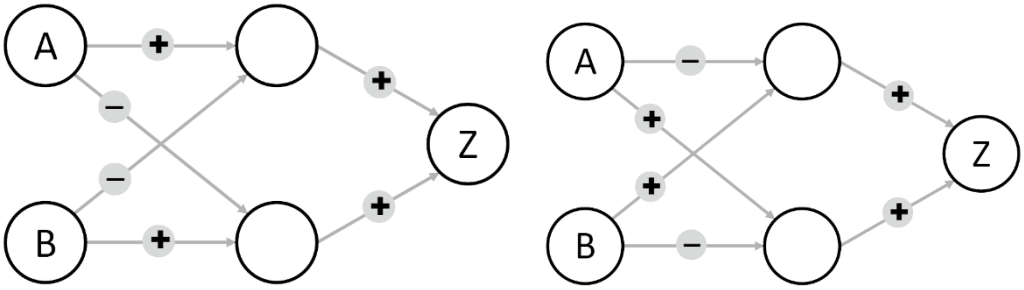
Határozd meg minden egyes kábel esetében, hogy pozitív (+) vagy negatív (−) jelet küldjön-e, hogy a konfliktusdetektor megfelelően működjön.



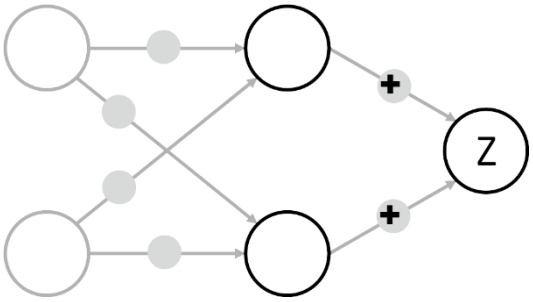


Megoldás

Két lehetséges megoldás létezik

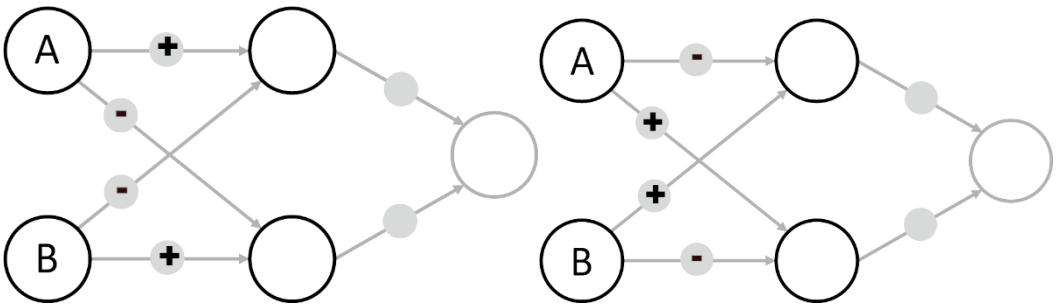


A konfliktusdetektornak pontosan két különböző bemenet („`Igen` és `Nem`”, valamint „`Nem` és `Igen`”) esetén kell `Igen`-t mutatnia a kimeneten. Ugyanakkor az egységek úgy vannak felépítve, hogy csak akkor adnak kimenetet, ha több pozitív, mint negatív bemenet érkezik. Így értelemszerűen a „Z” egységnek mindkét bemenetre pozitív jelet kell kapnia (ha mindkét kábel negatív lenne, akkor soha nem jutna el az `Igen` állapotba, ha pedig a két kábel közül az egyik negatív lenne, akkor csak egy esetet lehetne figyelembe venni az előtte lévő szinten). Tehát a jobb oldali két kábelt „+” kábelnek kell választani.

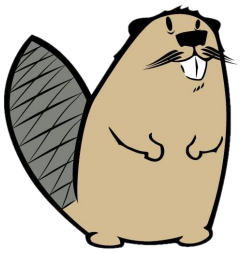


Innentől szimmetrikusan gondolkodhatunk tovább. Tehát a két egység közül az egyiket választjuk az egyik esethez, a másik egységet pedig a másik esethez. Mindkét egységnek pontosan egy `Igen` esetén kell jelet küldenie. Egyébként pedig nem küldhet jelet. A feladat példájában szereplő összeállítással pontosan ez az eset áll fenn (ha mindkettő pozitív lenne, akkor az „igen és igen” vagy a szimmetrikus másik esetben is jelet küldene kimenetként, ha pedig mindkettő negatív lenne, akkor soha nem küldene jelet): az egyik vezetéknek „+” -nak, az egyiknek „-” -nek kell lennie.

Mivel a feladat szimetrikus, kétfélekképpen választhatjuk az egyik kimenő kábelt „+”-nak, a másikat pedig „-”-nak:



Így elő is állt a két lehetséges megoldásunk.



MIÉRT INFORMATIKA?

A konfliktusdetektor két bemeneti értéket (Igen vagy Nem) dolgoz fel, és az Igen kimenetet csak akkor adja vissza, ha a két bemeneti érték különbözik. Ezt a logikai függvényt „kizáró vagy"-nak (XOR) nevezzük. Anna és Ben gépének első változata (két be- és egy kimeneti egység) egy egyszerűsített változata Frank Rosenblatt által 1957-ben leírt perceptronnak (mesterséges neuronnak). A kimeneti egység egy idegsejtet (neuront) mutat, amely képes a bemeneti jeleket feldolgozni és egy kimeneti jelet generál. Ezzel a perceptronnal megvalósíthatók az és (and) és a vagy (or) logikai műveletek, de a kizáró vagy (xor) nem.

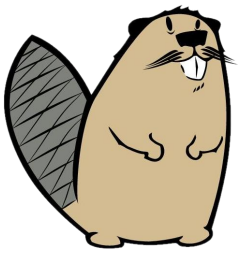
Ehhez szükség van egy másik kapcsolóegységekből álló rétegre, mint a feladat megoldásánál. Ezt csak az 1980-as években ismerték fel (pl. Rumelhart, Hinton & Williams 1986), majd (később) sikerült olyan mesterséges neurális hálózatokat programozni, amelyek az emberi agyhoz hasonlóan működnek, képesek például kameraképek kiértékelésére és tárgyak felismerésére.

WEBOLDALAK

https://gyires.inf.unideb.hu/GyBITT/19/ch02.html

https://hu.wikipedia.org/wiki/Boole-algebra_(informatika)#Kiz%C3%A1r%C3%B3_vagy

http://www.cs.toronto.edu/~hinton/absps/naturebp.pdf

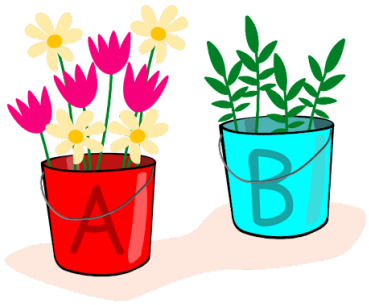


VIRÁGBOLT (2023-DE-02)

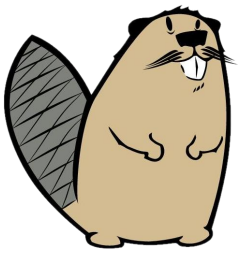
KISHÓD – KÖZEPES
BENJAMIN – KÖNNYŰ

Flórián virágcsokrokat árul. Minden csokrot az alábbi utasítások szerint köt meg. Minden egyes utasítást csak egyszer hajt végre a megadott sorrendben.

1. Flórián kivesz egy virágot az „A” vödörből.
2. Ha ez a virág egy margaréta , akkor Flórián vesz egy másik margarétát .
3. Flórián vesz egy ágat a „B” vödörből, és ezt a 3. lépést addig ismétli, amíg a csokra 4 növényből nem áll.

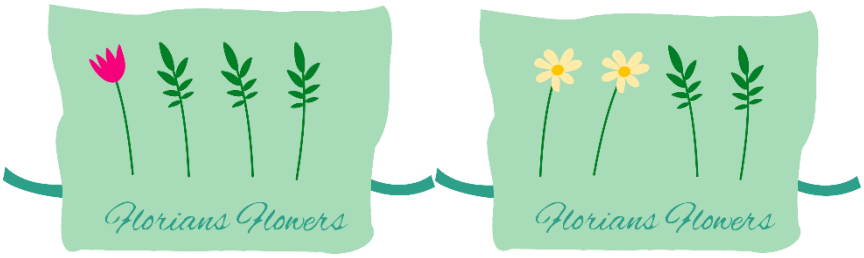


Hogy nézhet ki Flórián csokra?



Megoldás

Két helyes megoldás létezik:



A csokrok helyes megkötéséhez Flóriánnak követnie kell az utasításokat.

Miután Flórián kiválasztott egy virágot az „A" vödörből, döntenie kell a virágtól függően. Vagy vesz egy másik virágot (egy margarétát), vagy nem, és a 3. utasításhoz lépve vesz egy ágat .

Ezután ellenőrzi, hogy van-e már négy darab növény a csokorban.

Ha nem, akkor egy új ágat kell vennie, majd újra ellenőrizni a darabok számát.

Tehát, ha először egy margarétát választ, akkor választ egy másik margarétát is, majd vesz két ágat.

De ha először egy tulipánt vett, akkor a 3. ponttól kezdve követnie kell az utasításokat, és az első ág után addig kell ágakat választania, amíg nem lesz 4 növénye. Így összesen 3 ág kerül a csokorba az egy tulipán mellé.

MIÉRT INFORMATIKA?

A csokor megkötésére vonatkozó utasítások egyértelműek, és akár egy gép is elvégezhetné őket. Az informatikában ezt algoritmusnak nevezik. Ez a feladat néhány tipikus utasítást használ, amelyeket sok számítógépes programban használnak:

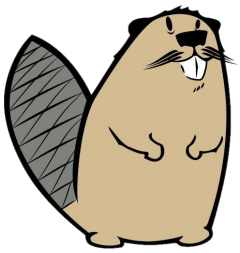
- Az első utasítás egy véletlenszerű kiválasztás egy objektumkészletből, azaz egy utasítás végrehajtása.

- A második utasítást _if-utasításnak_ vagy _elágazásnak_ nevezzük. Ebben az esetben két vagy több lehetőség közül kell választani egy feltétel teljesülése alapján.

- A harmadik utasítás viszonylag egyszerűnek tűnik, de egy számítógépes programban jól strukturáltnak kell lennie. A „Vegyünk egy ágat a B vödörből" utasítást többször meg kell ismételni, amíg a csokor 4 részből nem áll. Az utasítás tehát addig ismétlődik, amíg a „A csokor 4 részből áll." feltétel nem teljesül. Az ilyen utasítást _ciklusnak_ nevezzük.

Egy algoritmus többféleképpen ábrázolható. A feladatban az utasításokat mondatokként fogalmazzuk meg. De készíthetünk folyamatábrát is.

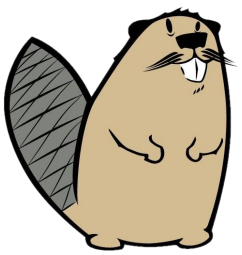
A virágkötészet egy mesterség. Hagyományok és szabályok vannak arra vonatkozóan, hogyan kellene egy csokor virágot vagy egy koszorút megkötni. Ez egy példa arra, hogy szabályok vagy



algoritmusok az élet számos területén előfordulnak, nem csak az informatikában. De az informatikával ellentétben a virágkötészetben el lehet térni a szabályoktól a művészeti hatás kedvéért.

WEBOLDAL

https://hu.wikipedia.org/wiki/Algoritmus

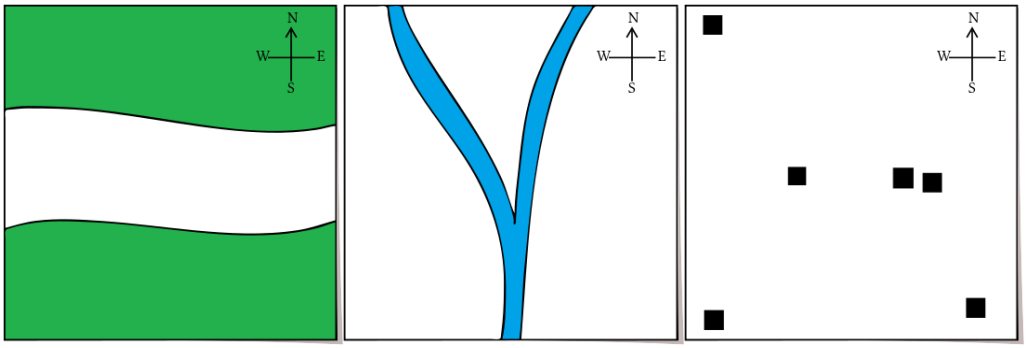


EMESE ÁLOMHÁZA (2023-DE-04)

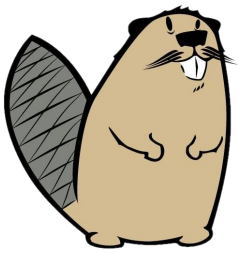
KISHÓD – NEHÉZ
BENJAMIN – KÖZEPES
KADÉT – KÖNNYŰ

Emesének három térképe van, amelyek mindegyike pontosan ugyanazt a területet mutatja: az egyik az erdőket, a másik a folyókat, a harmadik pedig a házakat. Emese álomháza az erdőben és a folyóparton van.

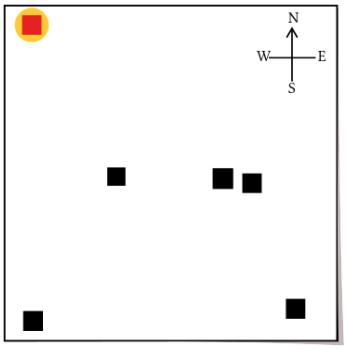
Erdők Folyók Házak



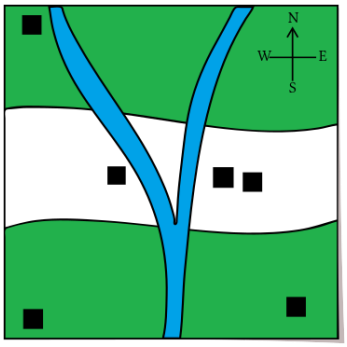
Melyik lehet Emese álomháza?



Megoldás



Az álomház megtalálásához mindhárom térkép információit ki kell értékelni. A kiválasztott háznak egy erdős területen és a folyóparton kell lennie, ami csak a bal felső sarokban lévő házra igaz. Ez könnyen észrevehető, ha a térképeket egymásra helyezve vizsgáljuk:



MIÉRT INFORMATIKA?

Ha az erdőkre, a folyókra és a házakra vonatkozó információk egyetlen térképen jelennének meg, akkor könnyen megtalálhatnánk a keresett házat.

A geoinformációs rendszer (GIS) a különböző információkat (pl. erdők, utak, országhatárok, benzinkutak, városházák, árterek stb.) gyűjti össze és jeleníti meg akár egyetlen térképen.

Az egyes információs rétegek szabadon figyelembe vehetőek, vagy figyelmen kívül hagyhatóak.

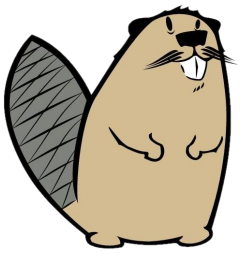
Például a katasztrófavédelem munkatársai összeállíthatják a menekülési tervekhez szükséges információkat és – akár egy programmal – ezeknek a figyelembevételével készíthetnek terveket, elemzéseket. Vagy egy elektronikus turistatérképen megjeleníthetőek a turistautak, a menedékházak, de a tömegközlekedési információk vagy a műemlékek is tetszés szerint.

A grafikus programokból is ismert a különböző képi információkat tartalmazó, több rétegből álló rajz. Fontos kérdés ezeken, hogy melyik réteg vagy mely objektumok milyen sorrendben láthatóak. A példában a házakat tartalmazó térképnek kell a legfelső rétegnek lennie, ahhoz, hogy a házakat ne takarják el az erdőterületek.

WEBOLDALAK

https://hu.wikipedia.org/wiki/F%C3%B6ldrajzi_inform%C3%A1ci%C3%B3s_rendszer

https://nat2012.nkp.hu/tankonyv/informatika_7/lecke_03_004

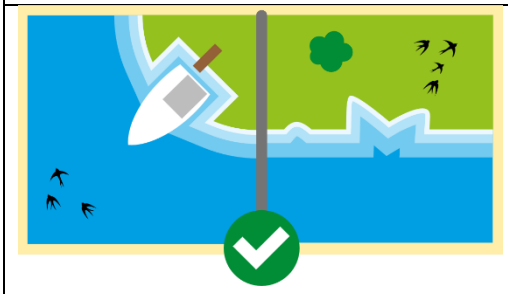
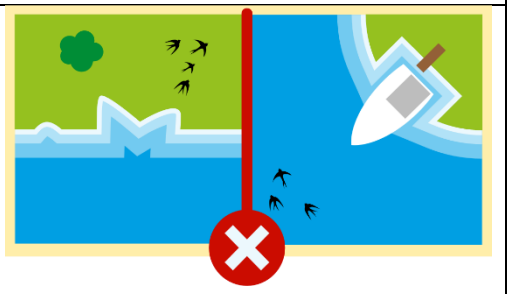


KÁRTYATÉRKÉP (2023-DE-06)

KISHÓD – KÖNNYŰ
BENJAMIN – KÖNNYŰ

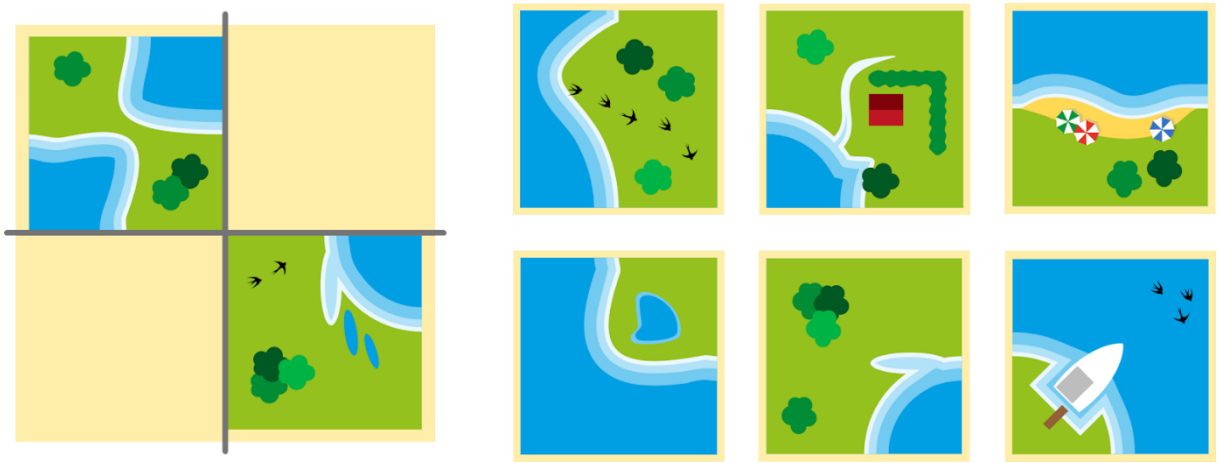
Edu új játéka víz és szárazföld területeket tartalmazó kártyákból áll. Amikor lerakja ezeket egymás mellé, a szomszédos kártyáknak úgy kell illeszkedniük egymáshoz, hogy a vízhez víz, a szárazföldhöz pedig szárazföld kerüljön.

Például a bal oldali képen helyesen lerakott kártyák láthatók, míg a jobb oldali képen a kártyák helytelenül vannak lerakva.

Helyesen lerakott kártyák	Helytelenül lerakott kártyák
	

Edu lerakott két kártyát, és további két kártyát szeretne hozzájuk lerakni.

Segíts neki: forgatás nélkül válaszd ki a hiányzó helyekre illő kártyákat.

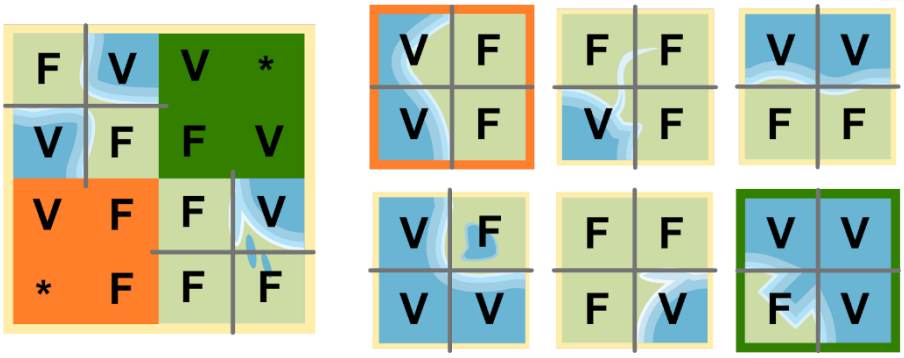


Megoldás

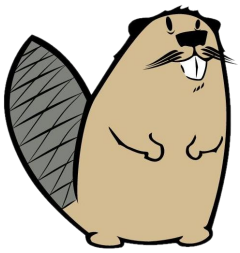
Azonnal látjuk, hogy ez a válasz helyes: minden beillesztett kártya illeszkedik a szomszédos kártyákhoz. Ha mind a hat kártyát megnézzük, azt látjuk, hogy csak a két kiválasztott kártya illeszkedik megfelelően (forgatás nélkül). Tehát ez az egyetlen helyes válasz.

Nézzük meg közelebbről Edu kártyáit. Először is, észrevehetjük, hogy a kártyák mindegyike 4 részre osztható, és minden oldal két-két külső széle vagy szárazföld vagy víz.

Ezeket "V"-vel (azokat a szakaszokat, amelyek külső szélei vizet mutatnak) és "F"-fel (azokat a szakaszokat, amelyek külső szélei szárazföldet mutatnak) jelölhetjük.



Azt is tudjuk, hogy két kártya akkor illeszkedik egymáshoz, ha a szomszédos szélei megegyeznek. Ezért a hiányzó kártyákra be is tudjuk írni a szükséges betűket. A hiányzó kártyák külső sarkaiban lévő két négyzet esetében nem törődünk a betűkkel, és beírunk egy "*"-ot. Mindkét mintának csak egy, a jobb oldalon lévő kártya felel meg. Összefoglalva, felfedeztük a kis kártyák egy különleges tulajdonságát, és ezt a felfedezést arra használtuk fel, hogy az F és V jelek használatával helyettesítsük őket. Ezzel a lépéssel jelentősen csökkentettük a kártyákon szereplő információt, hogy arra koncentrálhassunk, ami a feladat megoldásához szükséges.



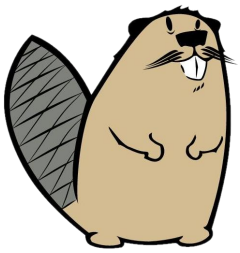
MIÉRT INFORMATIKA?

Az informatikusok sokszor hasonlóan dolgoznak: a valós tárgyból létrehoznak egy modellt, ami csak feladatban lényeges adatokat tartalmazza. A modellezés során létrehozott elvont modellt absztrakciónak is nevezzük. A számítógépeknek mindig a valóság modelljeivel kell dolgozniuk. Az ilyen modellek létrehozásakor ügyelni kell arra, hogy a valóság fontos tulajdonságait ne veszítsük el, de ne is tároljunk olyan információkat, melyek feleslegesek.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Absztrakci%C3%B3

https://hu.wikipedia.org/wiki/K%C3%B3d_(informatika)



ZEROBOT DILEMMA (2023-DE-08)

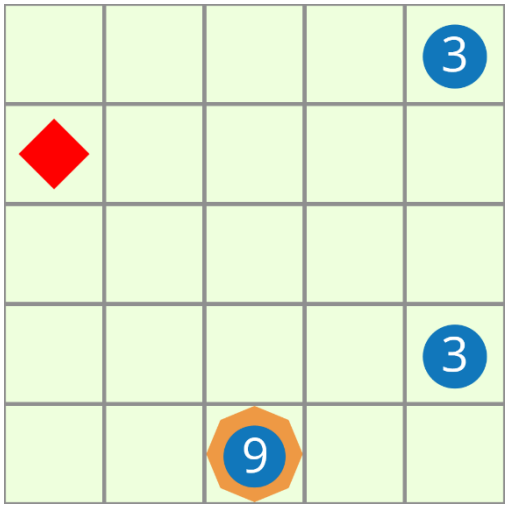
JUNIOR – NEHÉZ
SENIOR – KÖZEPES

A Zerobot üzemanyagcellája hidrogénnel működik. Biztonsági okokból a munkája során el kell fogyasztania az összes hidrogént minden tartályból. Azaz a munka befejeztével minden tartálynak üresnek kell lennie.

A Zerobot cserélhető hidrogéntartállyal közlekedik. A négyzetekből álló padlón felfelé, lefelé, jobbra és balra tud mozogni. Minden egyes alkalommal, amikor az egyik négyzetből a másikba lép, a tartály szintje 1-gyel csökken.

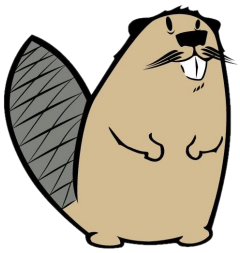
Néhány mezőn cseretartályok ( ) vannak. Amikor a Zerobot egy ilyen mezőre érkezik, kicseréli a tartályát a mezőn lévő cseretartályra, függetlenül attól, hogy melyik tartály mennyire van tele. Ez azt jelenti, hogy elveszi a cseretankot és a saját tankját a mezőre helyezi. Ezután továbbhalad.

A Zerobotnak vissza kell mennie a bázisállomására ( ). Mire odaér, minden tartálynak teljesen üresnek kell lennie. A Zerobot aktuális helyét és tartályának töltöttségi szintjét az ábrán a 9 szimbólum jelzi.



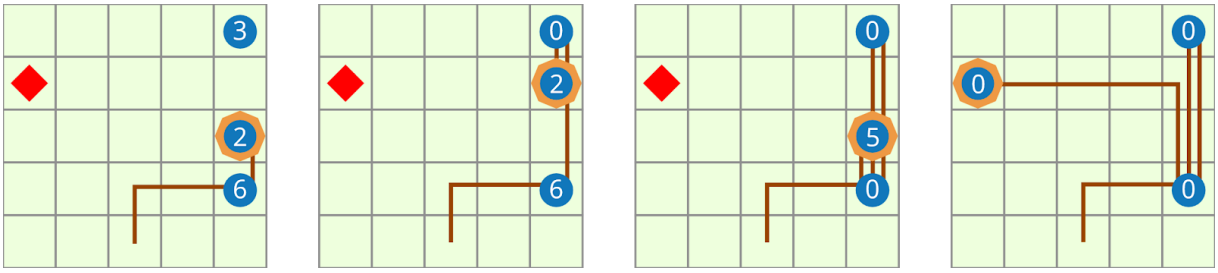
Hogyan kell a Zerobotnak mozognia?

A Zerobot mindig a legrövidebb úton halad a kijelölt célállomás felé és többször is cserélhet tartályokat.



Megoldás

Az ábrák azt mutatják, hogy a Zerobot 0 töltöttségi szint mellett pontosan 15 lépés alatt éri el a célját.



A Zerobotnak pontosan 15 lépést kell megtennie, hogy az összes rendelkezésre álló üzemanyagot, azaz 9 + 3 + 3 = 15 egységet felhasználja. Ennél tovább nem tud menni. Ahhoz, hogy az összes üzemanyagot megkapja, mindkét cseretartállyal rendelkező négyzetet meg kell látogatnia, az egyiket kétszer. Ha a Zerobot először a jobb felső cseretartályt látogatná meg, 17 lépést kellene tennie, hogy elérje a bázisállomását, ami nem lehetséges. Így a bemutatott mozgássorozat az egyetlen lehetséges megoldás.

MIÉRT INFORMATIKA?

A feladat az autonóm mobilitás néhány alapvető problémájával foglalkozik: Minden autonóm mobil robotnak (például egy önvezető autónak) figyelembe kell vennie, hogy tevékenységei megtervezésekor mennyi energia áll rendelkezésére üzemanyag vagy akkumulátortöltés formájában. Egyrészt gondoskodnia kell arról, hogy időben elérjen egy töltőállomást vagy benzinkutat, mielőtt az energiakészletei elfogynának. Másrészt keretfeltételeket kell figyelembe venni. A feladatban az egyik keretfeltétel az volt, hogy az energiakészletnek a végére teljesen el kell fogynia. A valóságban más keretfeltételekkel is számolni kell, például a töltőállomások elhelyezkedésével és elérhetőségével.

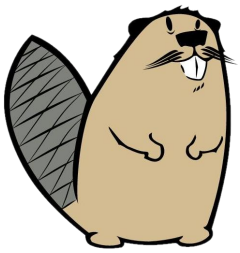
A mobilrobotok vezérlésére szolgáló szoftver olyan komponenseket tartalmaz, amelyek a feltöltés révén biztosítják az elegendő energiát (intelligens akkumulátortöltés-menedzsment).

Ezen kívül számítógépes programokat használnak a töltőállomások hatékony hálózatának megtervezésére és kezelésére is. Az informatikusok megoldásokat keresnek a töltőállomások elhelyezésének problémájára: a mobil robotok töltőállomásait úgy kell elhelyezni, hogy egy bizonyos minimális töltöttségi szinttel rendelkező robot el tudja érni a rendelkezésre álló töltőállomások egyikét. A töltőállomások és az önvezető autók közötti kommunikációra olyan protokollokat fejlesztettek ki, mint az OCPP (Open Charge Point Protocol).

WEBOLDALAK

https://transpack.hu/2022/01/03/robot-autonom-erzekelo-mir-ipar-4-0-iot-intelligencia-autorobot-intralogisztika/

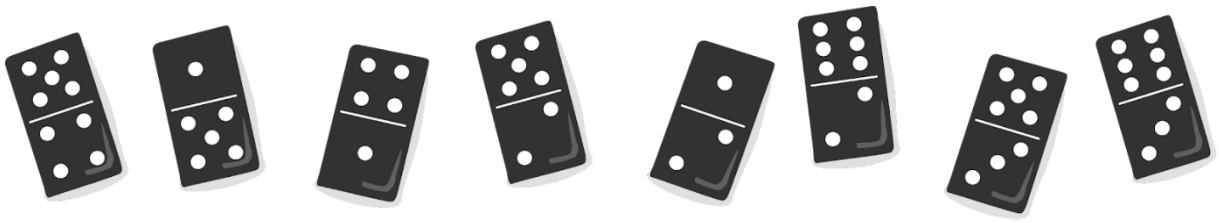
https://en.wikipedia.org/wiki/Open_Charge_Point_Protocol



DOMINÓ (2023-DE-09)

SENIOR – NEHÉZ

Egy dominónak két fele van. Mindkét felén 1-től 6-ig találhatóak pöttyök. Az alábbi 8 dominónk van:



Ezeket a dominókat a következő szabályok szerint kell egymás mellé, sorba lerakni:

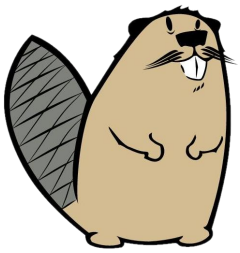
Mind a 8 dominót fel kell használni

Két, egymás melletti dominó szomszédos oldalán, mindig ugyanannyi pöttynek kell lennie

A rendelkezésre álló dominókból több ilyen sor is kirakható. A dominókon szereplő pöttyök nélkül egy megoldás így néz ki:

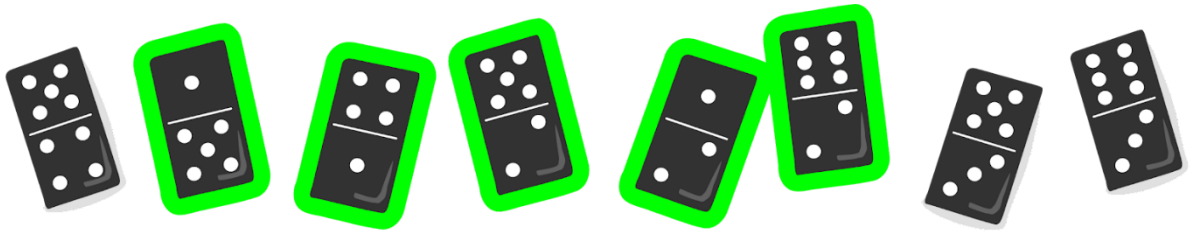
Észrevehető, hogy a sor elején és a végén jelölt helyekre, nem illeszthető be az összes dominó.

Válaszd ki az összes olyan dominót, amelyet elhelyezhetünk a sor elején és végén jelölt helyre

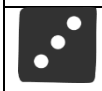
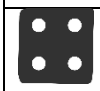
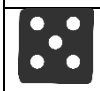
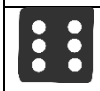


Megoldás

A nyolc dominóból öt kerülhet a sor elejére vagy végére.

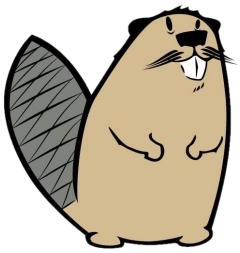


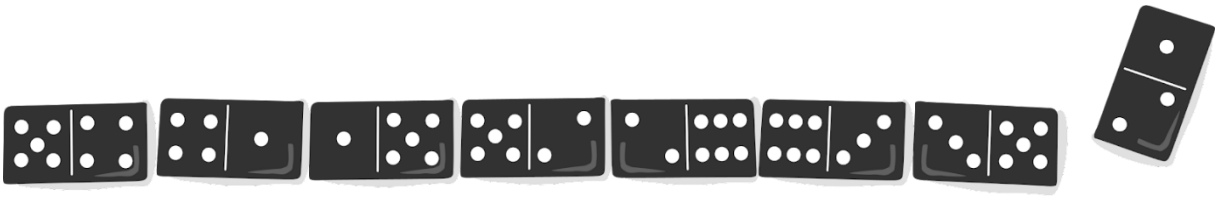
Kezdjük azzal, hogy megszámoljuk az egyes dominókon lévő pontok számát, és megvizsgáljuk, hogy az egyes pontok milyen gyakran, hányszor jelennek meg. A mi feladatunkban nyolc dominó van, ami 16 oldalra osztható. Az alábbi táblázat egy összefoglalót mutat, valamint egy extra oszlopot, amely megmutatja, hogy a pöttyök gyakorisága páros vagy páratlan:

Pontok	Megjelenés száma	Páros/páratlan
	3	páratlan
	3	páratlan
	2	páros
	2	páros
	4	páros
	2	páros

Figyeljük meg, hogy a 3, 4, 5 és 6 pöttyel ellátott oldalak páros gyakorisággal fordulnak elő, míg az 1 és 2 pöttyel ellátott oldalak páratlan gyakorisággal. Ez azért lényeges, mert két dominó szomszédos oldalainak mindig ugyanannyi pöttyel kell rendelkezniük, következésképpen párban kell előfordulniuk egy adott dominósorozatban. A 3, 4, 5 és 6-os dominók maradék nélkül párokba rendezhetők, míg az 1 és 2 dominók esetében ez nem így van.

A dominók sorrendjének szempontjából fontos megkülönböztetni a páros és páratlan gyakoriságú oldalakat. Különösen az 1-es és 2-es oldalakat (páratlan gyakoriságú oldalak) kell óvatosan elhelyezni, mivel ezek olyan helyzetekhez vezethetnek, amikor a sorozat nem folytatható a maradék dominók ellenére sem. Ha például úgy rendezzük el a dominókat, hogy az összes egy pöttyel rendelkezőt elhasználtuk, és a sorozat egy egy pöttyel rendelkező dominóval végződik, akkor a sorozat nem folytatható, még akkor sem, ha még vannak rendelkezésre álló dominók.



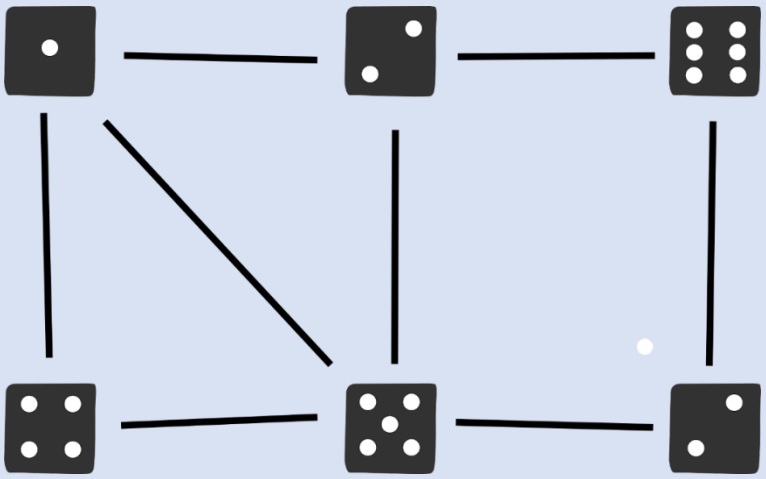


Ez a probléma azonban megoldható, ha szándékosan 1 pöttyel rendelkező dominót használunk szélső elemként. Ugyanez igaz a két pöttyel rendelkezőre is. A sorozat eleje és vége egyedi, mivel nem kell előző (az elején lévő dominó esetében) vagy következő (a végén lévő dominó esetében).

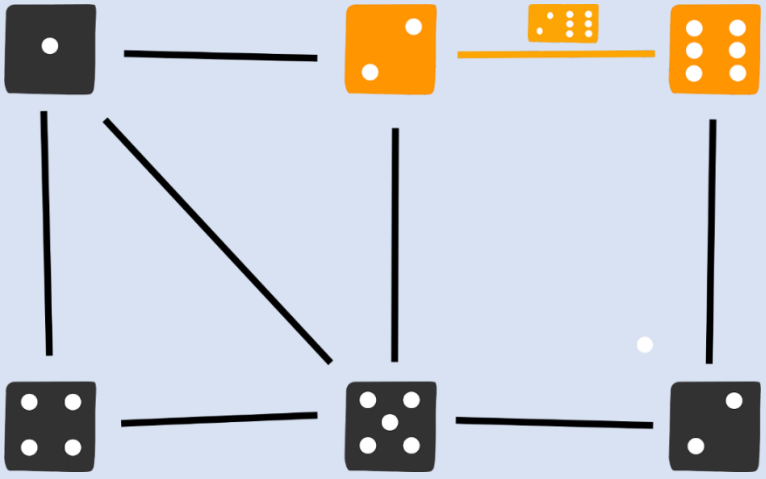
A nyolc dominó közül ötnek az egyik vagy mindkét felén az 1 vagy 2 pötty szerepel. Ezért a következő öt dominó tekinthető alkalmas szélső dominónak: (1,2), (1,4), (1,5), (2,5), (2,6).

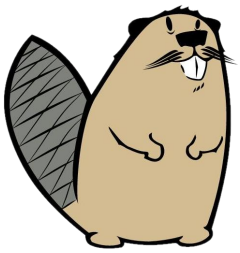
MIÉRT INFORMATIKA?

Számos olyan sor van, amely a szabályoknak megfelelően kirakható az adott nyolc dominó felhasználásával. Az ilyesfajta problémák jobb áttekinthetősége miatt, az informatikusok úgynevezett gráfokat használnak a probléma modellezéséhez:



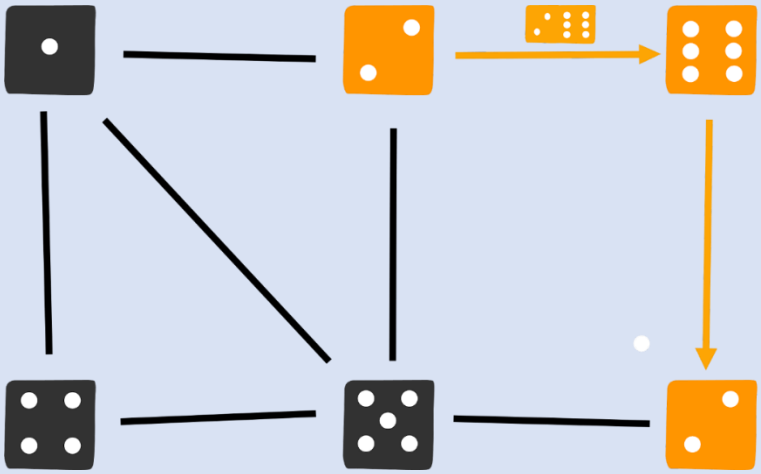
A fenti gráfon a négyzetek (úgynevezett csomópontok) jelölik a dominók lehetséges oldalait. A nyolc vonal (az úgynevezett élek), a nyolc dominót ábrázolja; minden vonal két számot (a dominó két felét) köt össze. Például a (2-6)-os dominót a következő él képviseli:





(Megjegyzés: bizonyos oldalak (pl. 5-6) között nincs vonal, mivel ez a kombináció nem szerepel a feladatban megadott dominók között.)

Ahhoz, hogy mind a nyolc dominó felhasználásával érvényes elrendezést kapjunk, olyan sorrendet kell találnunk, amelyben a pöttyök a szomszédos szegmensek között megegyeznek. Az első dominó elhelyezése után tehát már egyértelmű, hogy a második dominónak melyik számmal kell kezdődnie, mert két dominó szomszédos felének mindig ugyanannyi pöttyöt kell tartalmaznia. A gráfban ez abból látszik, hogy a dominók akkor és csak akkor helyezhetők egymás mellé, ha éleik ugyanabban a csomópontban találkoznak. Például a (2,6) és a (6,3) dominók egymás mellé helyezhetők, mivel mindkettőben szerepel a 6-os szám:

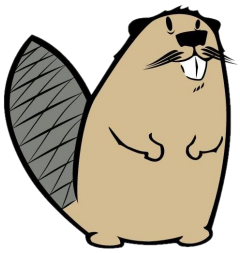


A dominók elrendezése úgy értelmezhető, mint egy útvonal (élek sorozata) a gráfon keresztül. Ennek az útvonalnak minden élét pontosan egyszer kell érintenie, hogy mind a nyolc dominót felhasználjuk, és egyiket se használjuk többször. Az olyan útvonalat, amely minden élt pontosan egyszer érint, Euler-útvonalnak nevezzük. Az elnevezés Leonhard Euler svájci matematikushoz köthető, aki a gráfelmélet feltalálója volt. Euler be tudta bizonyítani, hogy egy összefüggő gráfban akkor és csak akkor létezik Euler-útvonal, ha legfeljebb két olyan csomópont van, amelynek páratlan számú éle van.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Gr%C3%A1f

https://hu.wikipedia.org/wiki/Euler-k%C3%B6r



HÓDVÁR (2023-HU-37)

KADÉT – NEHÉZ
JUNIOR – KÖZEPES
SENIOR – KÖNNYŰ

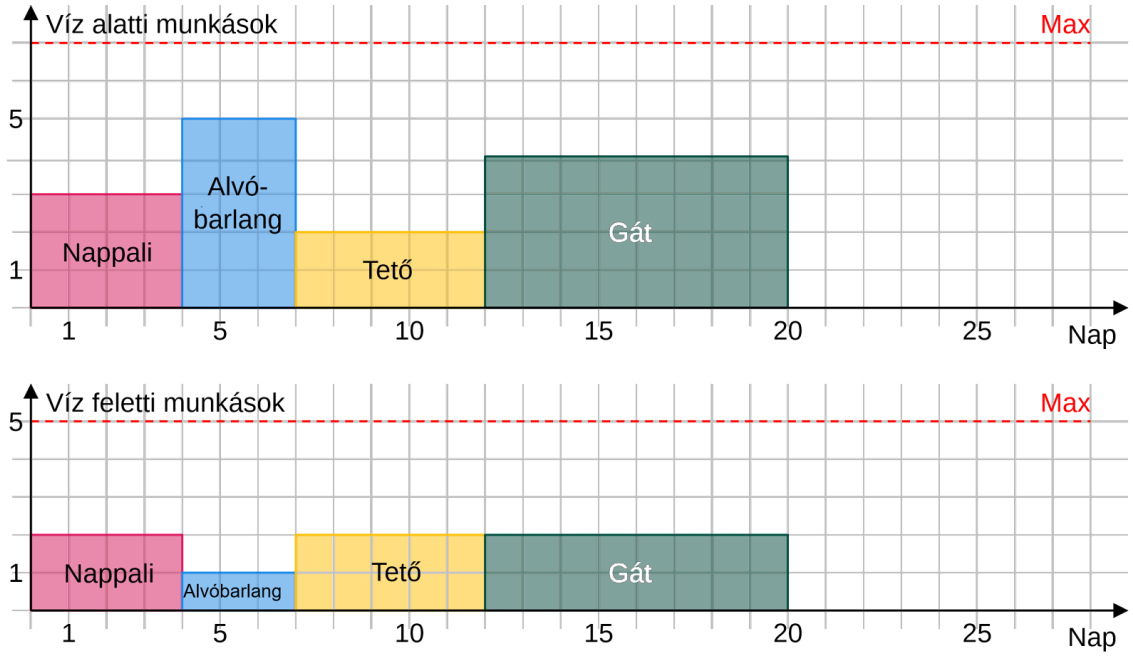
Minden hódvár 4 részből áll, amelyek mindegyike részben a víz alatt, részben a víz felett van. A hódvár építésénél minden munkás vagy csak a víz alatt, vagy csak a víz felett dolgozik. Minden résznél egyszerre folyik munka a víz alatt és a víz felett. A táblázatban minden egyes résznél látható, hogy mennyi ideig tart az építés, és hány munkásra van szükség a víz alatt és felett.

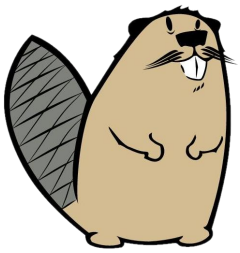
Szerkezet	Nappali	Alvóbarlang	Tető	Gát
Szükséges idő	4 nap	3 nap	5 nap	8 nap
Víz alatti munkások száma	3	5	2	4
Víz feletti munkások száma	2	1	2	2

A tetőt csak akkor lehet megépíteni, ha az alvóbarlang már elkészült. A többi rész építési sorrendje tetszőleges, akár egymással egy időben is építhetőek.

Készíts olyan ütemtervet a hódcsapat számára, amely lehetővé teszi, hogy a lehető leggyorsabban befejezzék a nagy várat, ha egyszerre legfeljebb 7 víz alatti és 5 víz feletti munkás áll rendelkezésre.

Mozdítsd el a részek elkészítését ábrázoló téglalapokat a tervben úgy, hogy a várépítés a lehető leghamarabb befejeződjön.





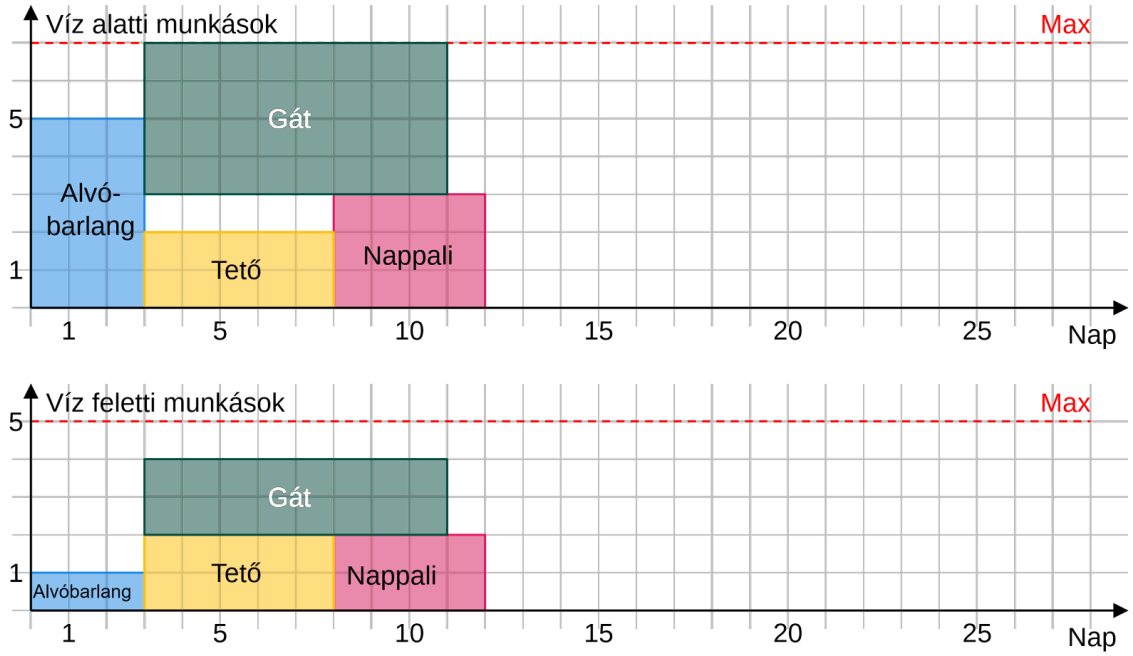
Megoldás

Az optimális, legrövidebb építési ütemezés megtalálásához a következő 2 megfontolást kell figyelembe venni:

1. Az alvóbarlang megépítését a tető megépítése elé kell tervezni. Mivel az alvóbarlanghoz 5 víz alatti munkás, a gáthoz 3, a lakótérhez pedig 4 víz alatti munkás szükséges, az alvóbarlang – a 7 víz alatti munkás korlátozásával – nem építhető meg egyszerre a gáttal és a nappalival sem. E megfontolás miatt optimális esetben először az alvóbarlangot kell megépíteni, és csak utána a másik három részt.

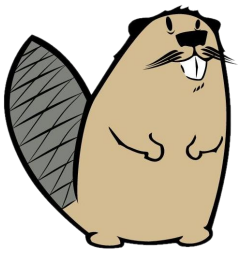
2. A gát és a nappali az alvóbarlang után egyszerre is megépíthető, vagy a két rész közül az egyiket a tetővel egy időben is meg lehet építeni. Nem lehet mind a 3 megmaradt részt egyszerre, mivel ezek együttesen 3+4+2=9 víz alatti munkást igényelnének. A legrövidebb építési időt úgy lehet elérni, ha a két rövidebb építési idejű szerkezetet (tető és nappali) egymás után építjük meg, és a hosszabb ideig tartó gátat velük egy időben építjük meg.

Így az optimális terv időtartama 12 nap. Egy lehetséges megvalósítás:



MIÉRT INFORMATIKA?

Az ütemtervek készítése egy olyan informatikai probléma, melyet számos alkalmazásban használnak. Egy-egy projekt részfeladatai között gyakran vannak időbeli függőségek, például az egyik részfeladat feldolgozása csak egy másik befejezése után kezdődhet meg. Ezenkívül a legtöbb projekttervet a lehető legrövidebb időn belül kell befejezni. A példában szereplő tervhez ábrázolt két diagram a Gantt-diagram egyik típusa, amelyet Henry Gantt (1861-1919) fejlesztett ki 1910 és 1915 között. Ezekben az összes erőforrás (a példában a különböző típusú munkaerő) időbeli felhasználását mutatják. A példánál nagyobb projektek esetében gyakran túl sokáig tart az összes lehetséges kombináció kipróbálása. Vannak gyorsabb módszerek is, de ezek gyakran az optimális terv helyett valamivel hosszabb építési idővel rendelkező tervet adnak. Ez a koncepció

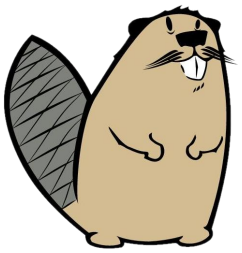


nem csak a nagyvállalati projekteknél alkalmazható, hanem a számítógépeknél is, amelyek folyamatai versengenek az erőforrásokért (számítási teljesítmény, memória-hozzáférés, külső eszközökhöz, például tárolóeszközökhöz, nyomtatókhoz vagy hálózati interfészekhez való hozzáférés).

WEBOLDALAK

https://support.microsoft.com/hu-hu/office/adatok-%C3%A1br%C3%A1zol%C3%A1sa-excel-programbeli-gantt-diagramon-f8910ab4-ceda-4521-8207-f0fb34d9e2b6

https://en.wikipedia.org/wiki/Gantt_chart



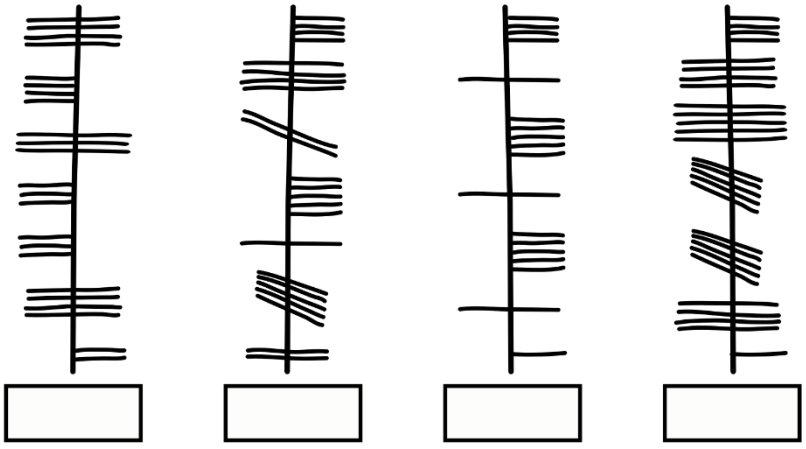
OGHAM KÓD (2023-IE-02b)

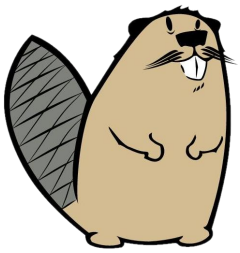
KADÉT – NEHÉZ
JUNIOR – KÖZEPES
SENIOR – KÖNNYŰ

Virág és Attila megismerkedtek a korabeli ír ábécével, az ogham-mal. Itt a betűk több kisebb vonalból állnak, melyek különbözőképpen vannak egy vonalra igazítva. Az írás során az egyes betűk között pedig nagyobb távolságokat hagynak és lentről felfele írják azokat.

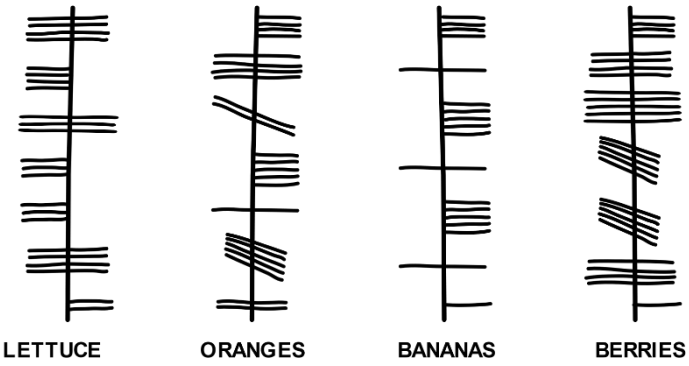
Virág le is írja az angolórán nemrég megtanult gyümölcsök nevét ogham ábécét használva: BANANAS, BERRIES, ORANGES, LETTUCE.

Segíts Attilának megtalálni az ogham ábécével leírt angol szavakat ! Húzd az angol szavakat az ogham ábécével leírt párjához!





Megoldás



A helyes megoldás többféleképpen is kideríthető.

Ha nem ismerjük az ábécét és szeretnénk betűről betűre végigmenni, a hasonló betűket is kereshetjük és hasonlíthatjuk.

Mivel lentről felfele sorakoznak a betűk és 3 szó is S-re végződik, ami a  jelet jelenti a szavak „tetején". Az első ábra lesz tehát a „lettuce" leírása, mivel ez az egy különbözik.

A maradék három szónál az első betűket nézve két esetben a „B" szerepel, tehát a második, a különböző lesz az „oranges".

A „bananas" háromszor is tartalmazza ugyanazt a betűt (A), ráadásul köztük N-ek szerepelnek. Ilyen mintát egyedül a harmadik ábránál találunk.

MIÉRT INFORMATIKA?

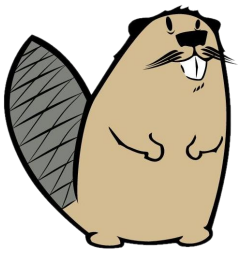
Ez a feladat példa egy ismert szöveg dekódolására egy ismeretlen írásmódban. Egy ismeretlen szöveg dekódolásakor az ember a gyakran használt szavak, betűk előfordulására tud építeni, és megpróbálja ezeket megtalálni a szövegben. Az ősi ábécéket és írásokat is így dekódolták. Például az egyiptomi hieroglifákat évszázadokig nem sikerült megfejteni, mígnem egy véletlen folytán megtaláltak egy hieroglifát és két ismert írást tartalmazó követ, a Rosetta-követ. Ez háromszor ugyanazt a szöveget tartalmazta. Ugyan különböző nyelveken írták (nem úgy, mint itt), de a nevek ugyanazok voltak. Így a hieroglifák lényeges elemeit sikerült megfejteni.

Az ősi írások megfejtése nem mindig ilyen egyszerű. A maja kultúra mintegy 650 írásjegyét még mindig nem sikerült teljesen megfejteni, és a mediterrán térségből származó lineáris A és lineáris B írásjelek sem jutottak el eddig a teljes megfejtésig.

Az informatikában is fontos a szövegek kódolása. A betűket azonban nem egyenként, vagy nem mindig ugyanarra a jelre kódolják, mivel az így kódolt szövegek meglehetősen könnyen dekódolhatók, például gyakori betűk vagy szótagok keresésével.

WEBOLDAL

https://hu.wikipedia.org/wiki/Ogham

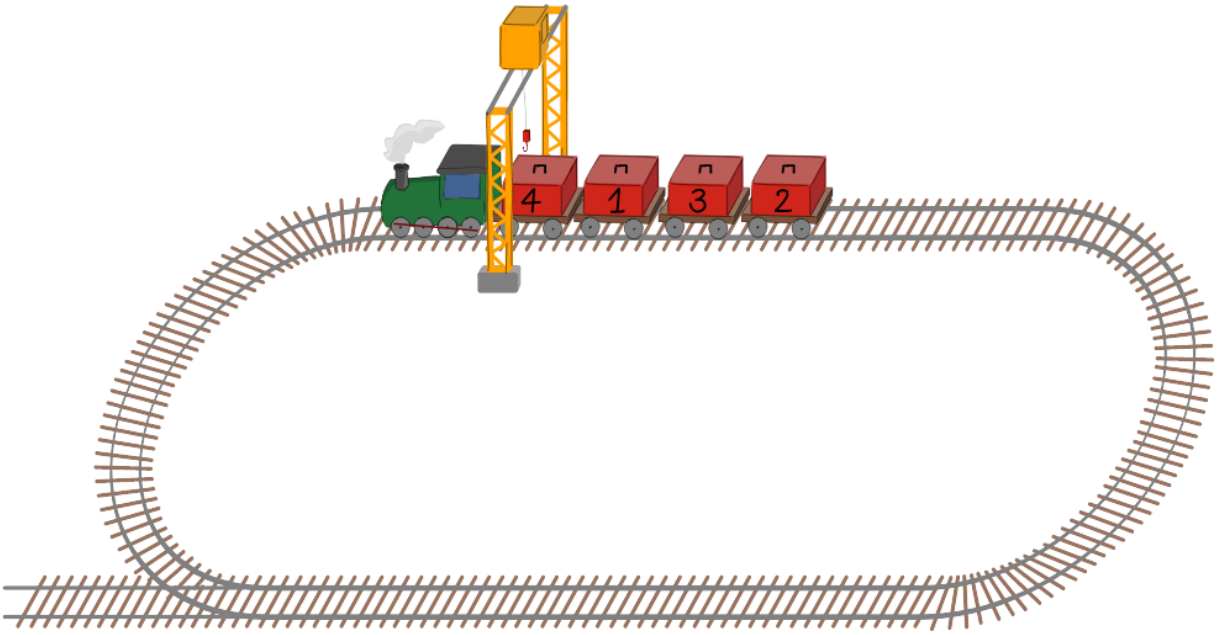


VONATRAKODÁS (2023-IN-03b)

BENJAMIN – NEHÉZŰ
KADÉT – KÖZEPES
JUNIOR – KÖNNYŰ

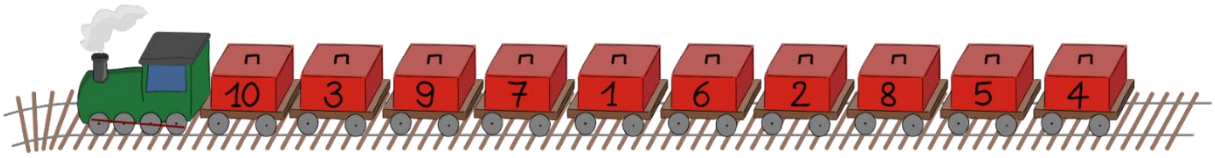
Egy tehervonat számozott konténereket tartalmazó vagonokból áll. A kirakodásra egyetlen darut használnak. A daru rögzített helyzetben van. Egy láda kiemeléséhez a konténert közvetlenül a daru alá kell vinni.

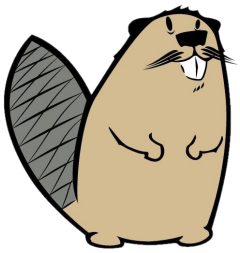
A konténereket a számozásuk sorrendjében kell kirakodni, az egyestől kezdve. A vonat csak előre haladhat. Miután teljesen áthaladt a daru alatt, körbe kell mennie, hogy újabb ládákat lehessen kirakodni.



Az ábrán lévő példában sorban az 1., 2., 3. és 4. konténert kell kirakodni. Ehhez három körre van szükség. Az első körben a 4-es konténert kihagyjuk, az 1-eset kirakodjuk, a 3-as konténert kihagyjuk és a 2-eset kirakodjuk. A második körben a 4-es konténert ismét kihagyjuk és a 3-asat kirakodjuk. Csak a harmadik körben kerülhet sor a 4-es konténer kirakodására.

Hány körre van szükség a következő vonat kirakodásához?





A helyes válasz: 7

A kirakodás előírt sorrendje: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Ha a fent leírt eljárást követjük, akkor az első körben az 1. és 2. konténereket, a második körben a 3. és 4., majd a következő körben az 5., majd a 6., majd egy körben a 7. és a 8., majd a 9. és az utolsó körben a 10. konténert lehet kirakodni. Ez hét kör.

Formálisan megfogalmazva, minden alkalommal, amikor a következő konténerszám nem hátrébb, hanem előrébb van a számsorban, egy további körre van szükség. Például, mivel a hármas a kettes előtt van, a kettes kirakodásakor a hármast kihagyják, így a vonatnak egy kört kell tennie, majd a hármast a daru alá kell vinnie. Az adott vonatban hat ilyen megfordított sorrendű számpár van (2,3), (4,5), (5,6), (6,7), (8,9) és (9,10), tehát összesen hét körre van szükség a megfelelő sorrendű kipakoláshoz.

MIÉRT INFORMATIKA?

A konténerek növekvő számsorrendben történő kirakodásával a vonat a konténereket a számok szerint rendezi. A rendezés fontos témakör az informatikában. Mivel a rendezni kívánt adatmennyiség igen nagy lehet, fontos figyelembe venni a rendezés hatékonyságát. Ebben a rendezési eljárásban, amelyben a rendezni kívánt sorozatot (a vonatot) a benne lévő elemekkel (a konténerekkel) mindig csak egy irányban lehet végignézni, a rendezés hatékonysága nemcsak az elemek számától, hanem a szükséges körök számától is függ.

A szükséges áthaladások számát a példában azon számpárok számából tudjuk kiszámolni, amelyek a rendezett sorrendben közvetlenül követik egymást, de a vonaton fordított sorrendben szerepelnek. Ez az előválogatás mértéke. A különböző listák előválogatásának más mérőszámai is lehetnek, például az inverziószám (ellentétes sorrendben álló elemek száma), a hiányzó elemek száma vagy az előválogatott részsorozatok száma.

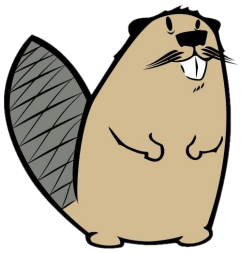
Az itt használt rendezési eljárás nem tartozik az olyan jól ismert nevű általános rendezési eljárások közé, mint a buborékrendezés vagy a gyorsrendezés. A példában levő rendezésre adott számolás csak az 1, 2, 3, 4, 5 ... konténerszámok permutációival működik, azaz ha a dobozszámok duplikálódnak egy vonaton, vagy ha egy dobozszám hiányzik, ez a számolási mód nem működik.

A sorozatok hosszától függő rendezések hatékonyságáról elmondható, hogy a megteendő körök száma lineárisan nő a vonat hosszával. Az összehasonlító műveletek és a cseremű veletek minden egyes körnél előfordulnak. Összességében a rendezési hatékonyság a vonat hosszával négyzetesen növekszik.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Algoritmus

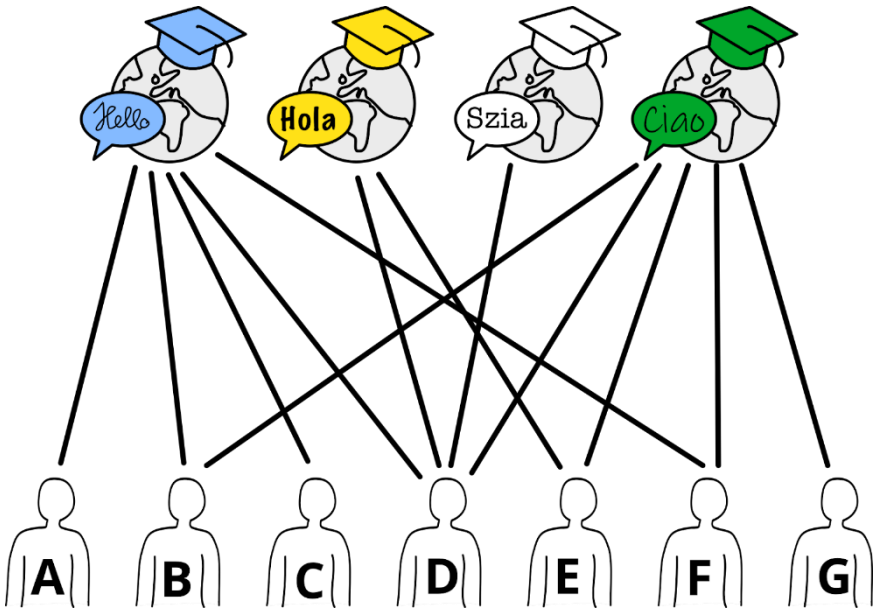
https://hu.wikipedia.org/wiki/Rendez%C3%A9s_(programoz%C3%A1s)



NYELVISKOLA (2023-IR-02)

KADÉT – NEHÉZ
JUNIOR – KÖZEPES
SENIOR – KÖNNYŰ

Egy nyelviskola négy nyelvtanfolyamot tervez. A rendelkezésre álló tanárok közül tudjuk, hogy ki melyik nyelveket taníthatja, lásd az ábrát.



Minden tanfolyamot el kell indítani, és minden tanfolyamon csak egy tanár taníthat. Azokat a tanárokat, akik nem tartanak tanfolyamot, máshová osztják majd be.

Az alábbi mondat közül melyik az, amely biztosan igaz?

A) Ha B, F és G tanár nem áll rendelkezésre, az egyik tanfolyamot törölni kell.

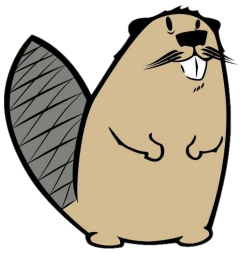
B) D tanár fog a spanyol 

tanfolyamon tanítani.

C) Az E tanár fog az olasz 

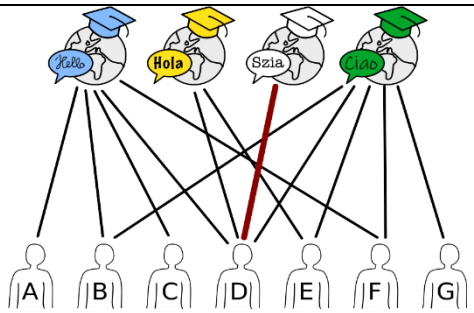
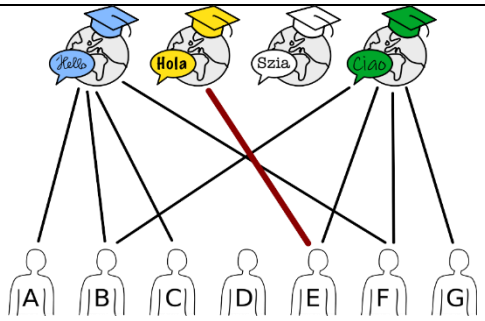
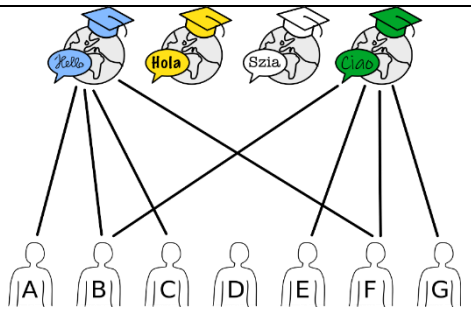
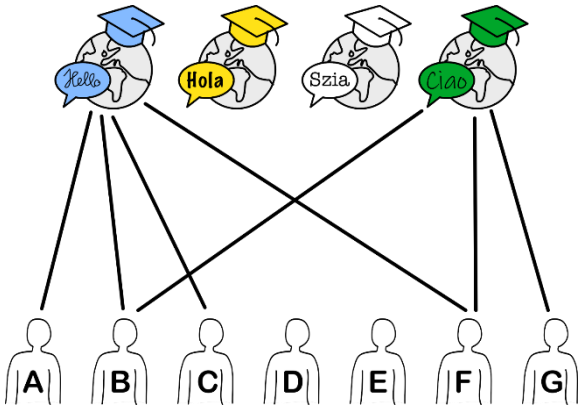
tanfolyamon tanítani.

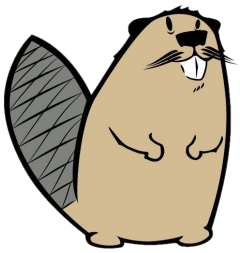
D) 4 olyan tanár van, aki nem fog nyelvtanfolyamon tanítani.



A helyes válasz az A)

Nézzük végig a tanfolyamokból kiindulva logikusan a lehetőségeket. Közben "távolítsuk el" a már kizárt vagy felhasznált összekötéseket a tanfolyamok és a tanárok között.

D az egyetlen tanár, aki taníthat a magyar tanfolyamon. Így a tanfolyam foglalt, és D nem taníthat más tanfolyamokon és ebből adódik, hogy a B) válasz hamis.	
Mivel D így már tanít, E az egyetlen tanár, aki taníthat a spanyol tanfolyamon. Így a tanfolyam foglalt, és E nem taníthat más tanfolyamokon, ezért a C) válasz szintén hamis.	
Az utolsó két fennmaradó tanfolyamhoz pedig szabadon választhatunk tanárt a megmaradtak közül. Mivel minden tanfolyamhoz több független, a másik nyelvet nem tanító tanár is tartozik, több lehetőségünk is van a megoldásra. A D válasz nem helyes, mert 2 tanfolyamra 5 tanár jut. Azaz, ha minden tanfolyamon tanít tanár, csak 3 (5-2) olyan marad, akinek nem jut tanfolyam	
Az A válasz igaz, mert ha "eltávolítjuk" B, F és G - tanárokat, akkor 4 tanár marad 4 tanfolyamra, azaz minden tanárnak kell tanfolyamot tartania. Az előző lépésekben már megmutattuk, hogy D a magyart és E a spanyol tanfolyamot tartja. A és C között kellene az olasz és az angol tanfolyamot elosztani, de egyikük sem tanít olaszt, így ez nem lehetséges.	

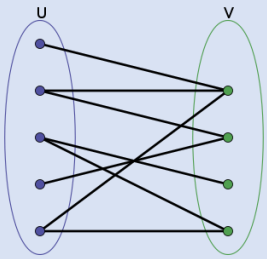


Ezt az "okoskodást" kezdhettük volna a tanfolyamok felől is: a magyar tanfolyamot csak D tarthatja meg, ekkor viszont a spanyol tanfolyamhoz már csak E rendelhető hozzá.

MIÉRT INFORMATIKA?

A gráf egy absztrakt fogalom, amely élekkel (vonalakkal) összekapcsolt csomópontokból (pontokból) áll. A gráfok egy speciális fajtája a kétrészes gráf, amely olyan gráf, amely csomópontjai 2 külön részhalmazra oszthatók úgy, hogy csak a különböző részhalmazok csomópontjai között vannak élek.

Feladatunk tehát egy kétrészes gráffal ábrázolható (lásd a kétrészes gráf tipikus ábráját a bal oldalon), amelynek első részhalmazát a kurzusok, második részhalmazát pedig a tanárok alkotják.



A kétrészes gráfok nagyon jól alkalmazhatók a feladatmegosztási problémák modellezésére, sőt megoldására is.

Most, hogy a konkrét problémánkat a megfelelő formális struktúrába helyeztük, előnyünkre válik, hogy sok ilyen jellegű problémának a megoldására már kész módszerek, algoritmusok állnak a rendelkezésünkre. Ezután ezeket a problémákat az ismert algoritmusokkal meg lehet oldani. Ilyen módon a feladat könnyedén megoldható Python nyelven is, ahol elérhető a probléma megoldására egy programozási könyvtár. Konkrétan ez azt jelenti, hogy valaki létrehozott egy olyan algoritmust, amely az adott probléma megoldására lett kifejlesztve és ezeket az eljárásokat kész módszerként a rendelkezésünkre bocsátja.

A feladat Python nyelven történő megoldása az alábbi linken érhető el: https://www.coding4you.at/dachu_2023/ir02/ir02_graphs.py

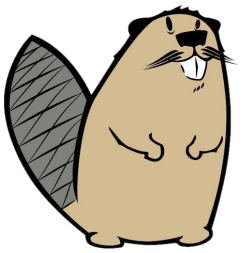
Egy szintén érdekes, kétrészes gráffal ábrázolható és megoldható probléma az úgynevezett házassági probléma. Itt egy házasságra hajlandó férfiakból álló halmaz áll szemben egy nem feltétlenül azonos számú házasságra hajlandó nőkből álló halmazzal. Az eljárás célja, hogy az összes férfi vagy az összes nő (attól függően, hogy melyik halmaz a kisebb) megházasodjon, figyelembe véve a férfiak (vagy nők) kívánságait. Philip Hall angol matematikus 1935-ben a házassági tételben fogalmazta meg azokat a feltételeket, amelyek mellett egy ilyen felosztás lehetséges.

A probléma egy változatában nem erről a teljes hozzárendelésről van szó, hanem arról, hogy minél több sikeres párosítást érjünk el.

WEBOLDAL

https://hu.wikipedia.org/wiki/P%C3%A1ros_gr%C3%A1f

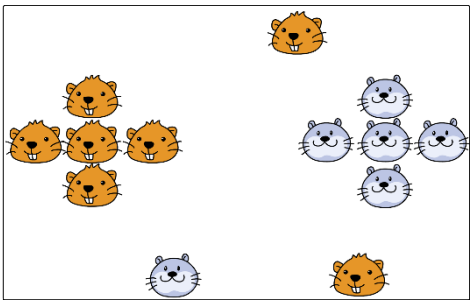
https://www.uni-miskolc.hu/~matka/Dokumentumok/Hozzarendelesi_feladat.pdf



HÓDOK ÉS VIDRÁK (2023-KR-01)

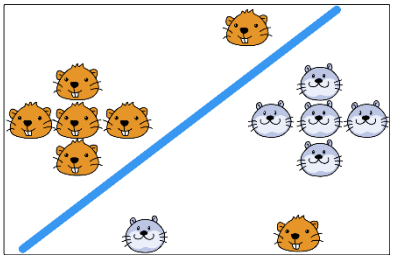
KISHÓD – KÖZEPES
BENJAMIN – KÖNNYŰ

A hódok  és a vidrák  szeretnék egymás között határvonallal felosztani a területet, amin élnek úgy, hogy a terület egyik felén csak hódok, a másikon csak vidrák éljenek.

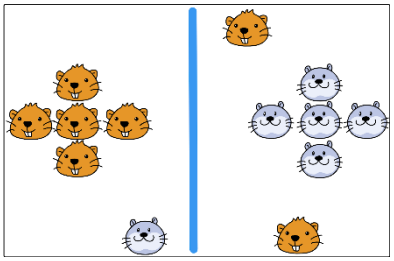


Melyik felosztásnál kell a legkevesebb állatnak elköltöznie a korábbi lakhelyéről?

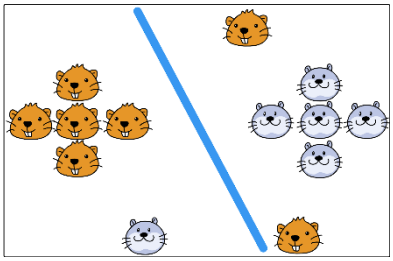
A)



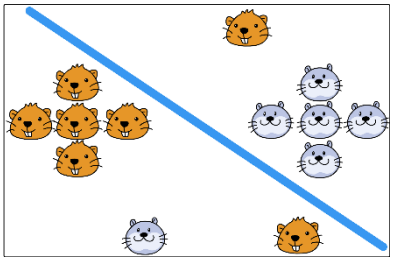
B)

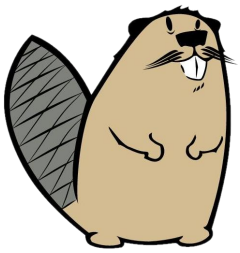


C)

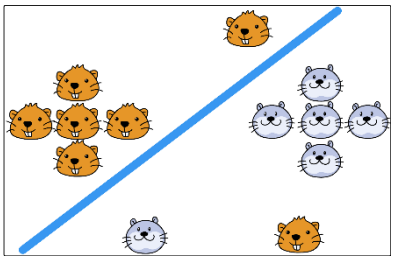


D)





A helyes válasz az A)



Tekintettel a hódok és vidrák jelenlegi elhelyezkedésére, a legjobb megoldás a határvonal megrajzolására a mellékelt képen látható. Ebben az esetben csak egyetlen egy hódnak kell költöznie, a jobb alsó oldalon.

MIÉRT IINFORMATIKA?

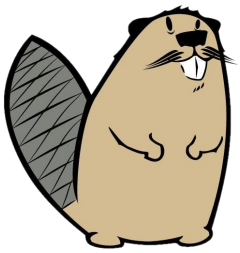
Az osztályozás és kategorizálás az informatikában olyan folyamatokat jelent, amelyek során adatokat vagy objektumokat csoportokba rendelünk közös tulajdonságaik alapján. Ez segíti az adatok hatékony kezelését és keresését.

Az osztályozás lehetővé teszi az adatok hierarchikus vagy szekvenciális rendezését, míg a kategorizálás segít a logikai csoportok vagy kategóriák létrehozásában. Az informatikában gyakran alkalmazzák az osztályozást és kategorizálást adatbázisok, keresőrendszerek és adatfeldolgozás során az adatok jobb szervezése és könnyebb hozzáférése érdekében. Az osztályozás és kategorizálás nélkül az adatok kezelése és értelmezése sokkal nehezebb és időigényesebb lenne.

Ebben a feladatban azt kellett megoldani, hogy a legkevesebb állatot elköltöztetve alakítsuk ki a hódok és vidrák csoportját. A csoportok megalkotása a lehető legkevesebb lépésből az osztályozás alapvető problémája.

WEBOLDALAK

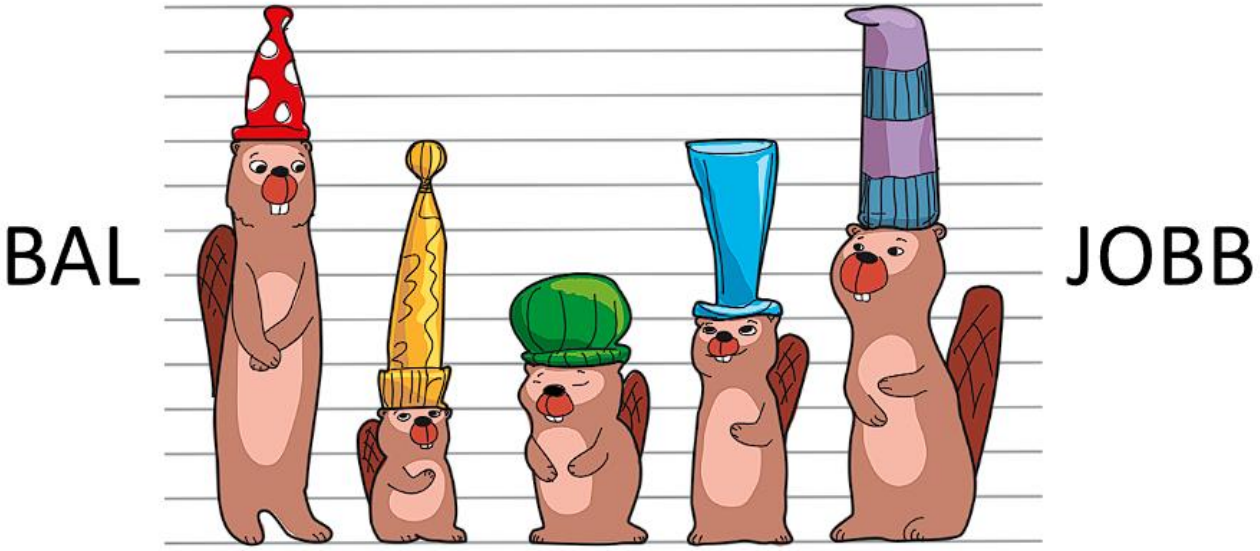
https://hu.wikipedia.org/wiki/Oszt%C3%A1lyfelbont%C3%A1s
https://hu.wikipedia.org/wiki/Oszt%C3%A1lyoz%C3%A1s_(k%C3%B6nyvt%C3%A1rtudom%C3%A1ny)



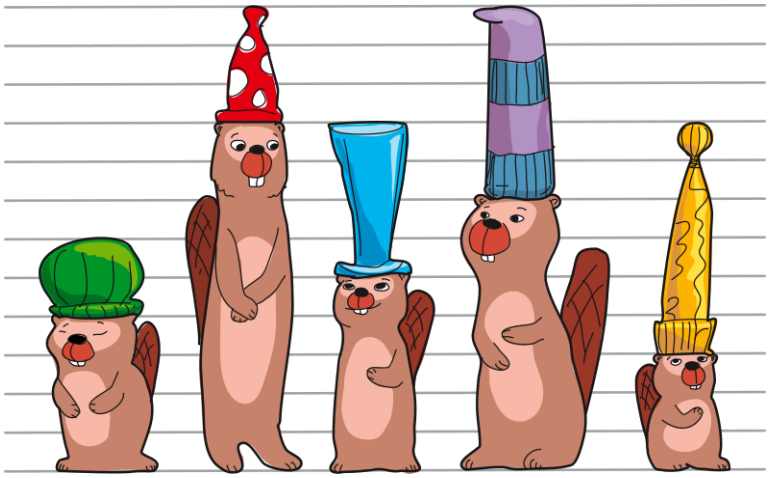
EMELEM KALAPOM (2023-LT-01)

KISHÓD – KÖNNYŰ

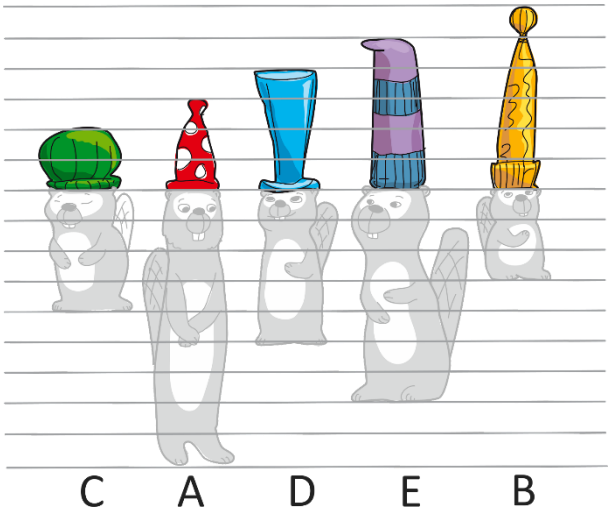
Rendezd a hódokat a kalapjuk magassága szerint balról jobbra növekvő sorrendbe.



A helyes válasz a:



Ha csak a kalapokra figyeljünk, akkor sokkal könnyebb a sorbarendezés.

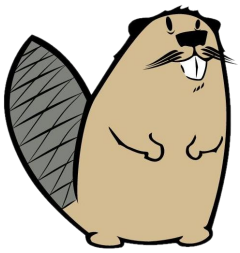


MIÉRT INFORMATIKA?

Sok dolog a környezetünkben, annak érdekében, hogy könnyebben megtaláljuk őket, rendezett vagy számozott: eszközök méret szerint, szócikkek egy lexikonban vagy szavak a szótárban, a névsor a naplóban…

Lehet számok szerint (mint például a méret) vagy ábécérendben rendezni. Színeket a hozzájuk tartozó színhullámhossz (szám) vagy a szín neve (betűrend) szerint is lehet rendezni. Mindenesetre bármilyen tulajdonság alapján lehet rendezni, amelyhez egyértelmű "kisebb/nagyobb, mint" viszonyt (rendezési viszony) lehet meghatározni. Ahhoz, hogy például egy kupac almát rendezzünk, először is szükség van arra, hogy megadjuk a rendezés alapját jelentő tulajdonságot, mint például a színt, nevet vagy méretet.

Ebben a feladatban a "Kalapos hódok" tulajdonságait kellett megfigyelned, és fel kellett ismerned, hogy nem csak egy mérettel rendelkeznek, hanem minden hódnak és a kalapnak is van külön-külön mérete. Ennek megfelelően, a kalapos hódokat három különböző tulajdonság alapján lehetne rendezni. Számunkra csak a kalapok mérete volt a fontos, ez adta meg, hogy mi alapján rendezzük a hódokat növekvő sorrendbe.



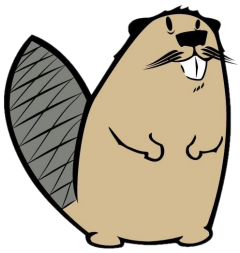
A különböző dolgok tulajdonságai és az azok alapján sorba rendezése, segítenek nekünk a mindennapi életben abban, hogy gyorsabban megtaláljunk dolgokat és gyorsabban, hatékonyabban férjünk hozzá a szükséges adatokhoz.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Rendez%C3%A9s_(programoz%C3%A1s)

https://hu.wikipedia.org/wiki/Rendezett_halmaz

https://hu.wikipedia.org/wiki/Keres%C5%91algoritmus



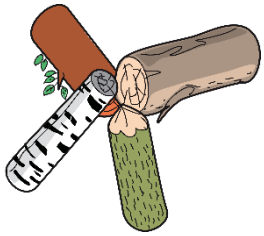
FOTÓ (2023-LT-02)

BENJAMIN – KÖNNYŰ

A hód egy fényképet készített:

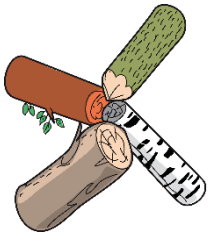
Melyiket készíthette a 4 kép közül?

A

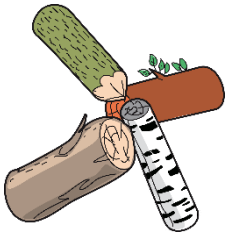


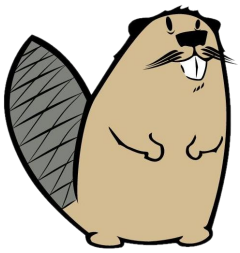
B

C

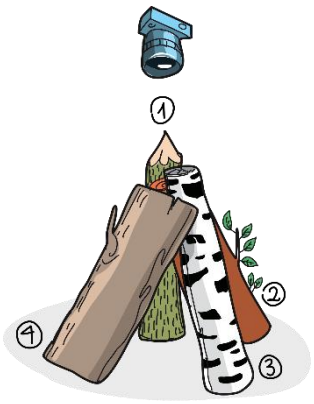


D





A helyes válasz a D)

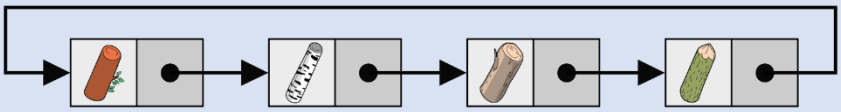


Ahhoz, hogy megtudjuk, melyik a helyes kép, meghatározzuk a fatörzsek helyzetét. Kiválasztunk egy fatörzset (pl. az 1. fatörzset, a hegyes végűt), és meghatározzuk, hogy melyik fatörzs van tőle balra. Ezt megtesszük az összes többi fatörzzsel is, és összehasonlítjuk az egyes fatörzsek helyzetét a fényképen látható helyével.

Kezdjük a hegyes fatörzzsel (1.). Ez a 2. és 4. barna fatörzs között van: Az A) és C) válaszok tehát nem lehetségesek. Kizárhatjuk a B választ is, mert ott a hegyes fatörzs (1.) a vastag barna fatörzstől (4.) jobbra van. Csak a D) válaszban van a hegyes fatörzs (1.) a vastag barna fatörzstől (4.) balra. A hód tehát csak a D) képet készíthette.

MIÉRT INFORMATIKA?

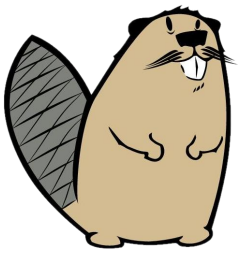
Ebben a hód feladatban a fatörzsek sorrendjét vesszük figyelembe. Ami néhány elem (itt négy fatörzs) esetén a szomszédos párok összehasonlításával lehetséges. Az ennél sokkal több elemű problémáknál már nem lenne ilyen könnyű, érdemes tehát automatizálni a megoldást. Ehhez segítséget adhat, ha az elemeket egy megfelelő adatszerkezetben, például egy összekapcsolt, láncolt listában tároljuk:



Egy **láncolt listában** minden egyes adatelemet egyetlen cellában tárolunk. Minden cella tartalmaz egy (vagy kettő) linket is a lista következő (és előző) cellájára. Ha az utolsó cella hivatkozást tartalmaz az első cellára, akkor az egy körkörös adatszerkezet. Ez azért fontos a példában, hogy bármelyik fatörzsből kiindulva végigmehessünk a listán.

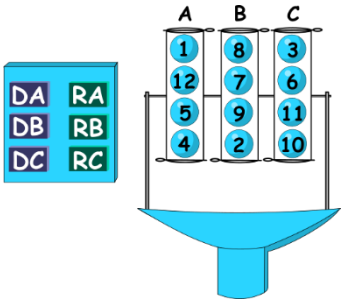
WEBOLDALAK

https://hu.wikipedia.org/wiki/L%C3%A1ncolt_lista

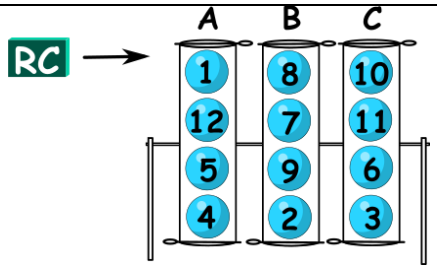
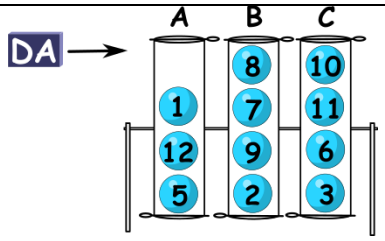


FORGÓ CSÖVEK (2023-ME-03b)

KADÉT – NEHÉZ
JUNIOR – KÖZEPES
SENIOR – KÖNNYŰ



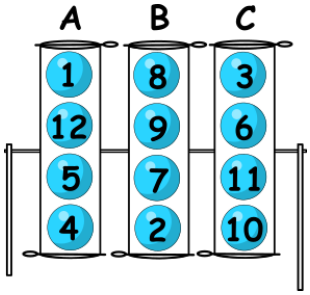
Hód Huba új játékot kapott a születésnapjára. A játékban három cső van, amelyekben egyenként 4 labda van. Huba megnyomhatja a 6 gomb egyikét a vezérlőpulton. Az RA, RB és RC gombok az A, B és C csövet 180 fokkal elforgatják (fejjel lefelé). A DA, DB és DC gombok a megfelelő csőből egy-egy labdát a tárolóegységbe dobnak.

Például, ha megnyomja az RC gombot, a C cső elfordul fejjel lefele:	Ezután, ha a DA gombot megnyomja, az A csőből kiesik a tartályba a 4-es (legalsó) golyó:
	

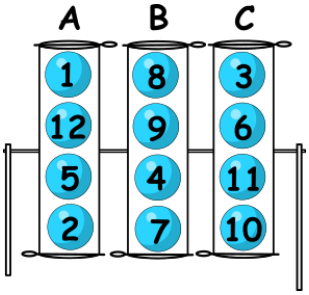
Huba az összes labdát növekvő sorrendben szeretné a csövekből a tárolóegységbe juttatni, de néha a labdák helyzete a csövekben ezt lehetetlenné teszi.

Az alábbi helyzetek közül melyik esetben lehetetlen elérni Huba célját?

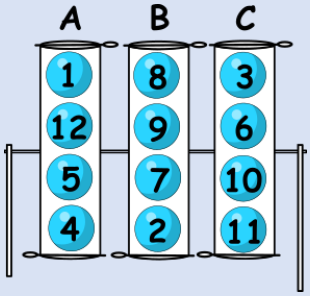
A)



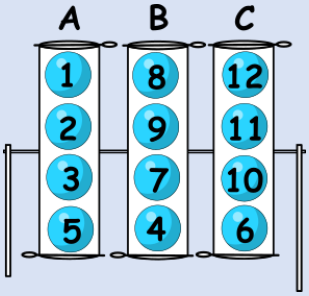
B)

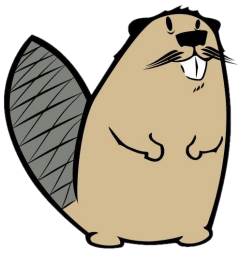


C)

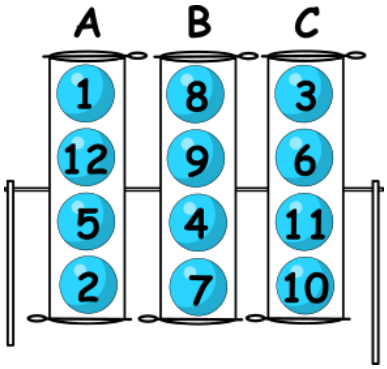


D)





A helyes válasz a B)



Tudjuk, hogy a labdák helyes sorrendje a végén a következő: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12. A 4-es labdát nem lehet a 7-es és a 9-es labda előtt a tárolóba dobni, mert a 4-es labda a 7-es és a 9-es labda között van (valamelyiket előtte kellene kidobni).

Az egyes csövekben lévő labdák sorrendjére alulról felfelé haladva az alábbi feltételek valamelyikének kell teljesülnie, hogy Huba növekvő sorrendben dobhassa le a labdákat:

1. növekvő sorrend
2. csökkenő sorrend
3. növekvő sorrend, majd csökkenő sorrend.

Ez az A, a B és a D válaszlehetőség esetén is fennáll.

Az A. lehetőség: Az A csőben lévő labdák növekvő sorrendben [4, 5, 12], majd csökkenő sorrendben [12, 1] helyezkednek el. A B csőben lévő labdák növekvő sorrendben [2, 7, 9], majd csökkenő sorrendben [9, 8] helyezkednek el. A C csőben lévő labdák növekvő [10, 11], majd csökkenő [11, 6, 3] sorrendben helyezkednek el.

C. lehetőség: Az A csőben lévő labdák növekvő [4, 5, 12], majd csökkenő [12, 1] sorrendben helyezkednek el. A B csőben lévő labdák növekvő [2, 7, 9], majd csökkenő [9, 8] sorrendben helyezkednek el. A C csőben lévő labdák csökkenő sorrendben [11, 10, 6, 3] helyezkednek el.

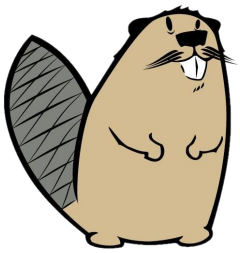
A D. lehetőség: Az A csőben lévő labdák csökkenő sorrendben [5, 3, 2, 1] helyezkednek el. A B csőben lévő labdák növekvő [4, 7, 9], majd csökkenő [9, 8] sorrendben helyezkednek el. A C csőben lévő labdák növekvő sorrendben [6, 10, 11, 12] helyezkednek el.

MIÉRT INFORMATIKA?

Ezt a feladatot rendezési problémának nevezzük. A rendezési algoritmusok a számítástechnika alapvető algoritmusai közé tartoznak. A rendezéshez számos algoritmus létezik, ezek egyike az összefésüléses rendezés. Az összefésüléses rendezés egy hatékony, általános célú és összehasonlításon alapuló rendezési algoritmus. Az összefésüléses rendezés során az adattömböt rekurzív módon 2 részre osztjuk, majd mindkét részt rendezzük, végül pedig összefésüljük őket. Az összefésülő rendezés egy változata a 3-rendezett sor összefésülése, ahol a tömb 2 részre osztása helyett, 3 részre osztjuk a tömböt.

A csövek példák egy "kétvégű sor" absztrakt adatszerkezetre. A kétvégű sor lehetővé teszi az elemek beszúrását és eltávolítását mindkét végéről (de csak a végéről).

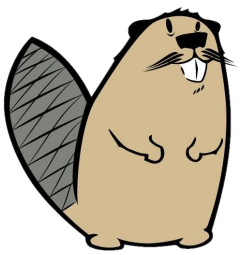
A csövekben lévő labdákat tekinthetjük a programunk bemeneti adatainak. A megadott konfigurációk három olyan tesztesetet írnak le, amelyre a programunk megoldást ad, és egy olyan tesztesetet, amelyre nem.



WEBOLDALAK

http://adatszerk1.elte.hu/downloads/eloadas/Eloadas2.pdf

https://hu.wikipedia.org/wiki/%C3%96sszef%C3%A9s%C3%BCl%C3%A9ses_rendez%C3%A9s



HÍDÉPÍTÉS (2023-NZ-01)

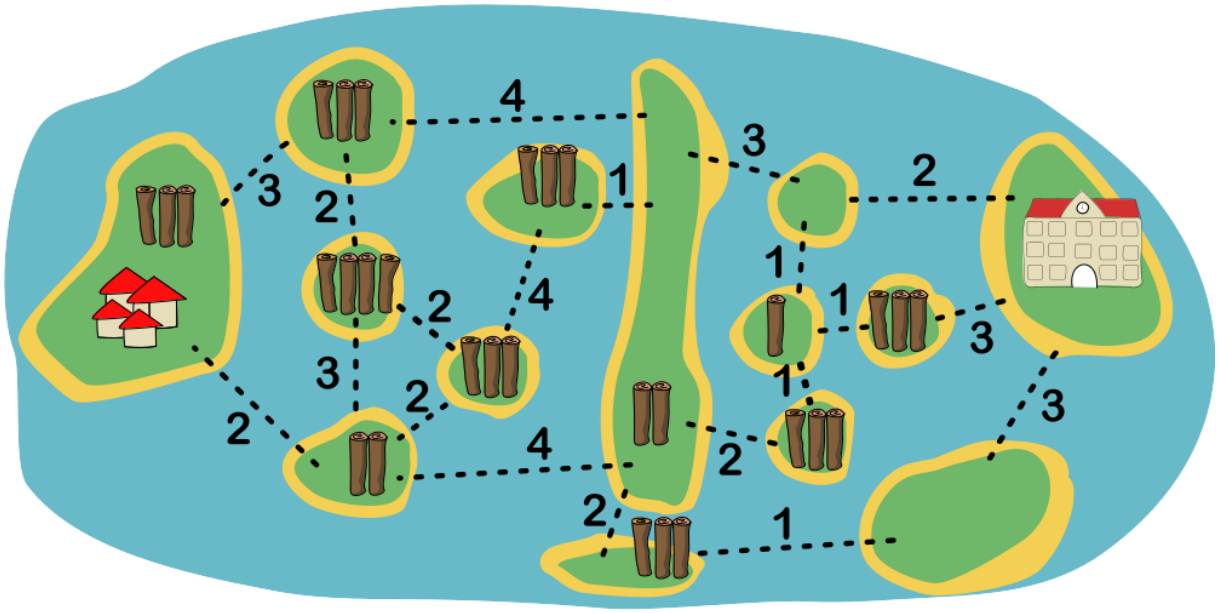
JUNIOR – NEHÉZ
SENIOR – KÖZEPES

Bob hídépítő következő feladata, hogy hidakat építsen ahhoz, hogy a falujából  a diákok eljuthassanak az új iskolába  .

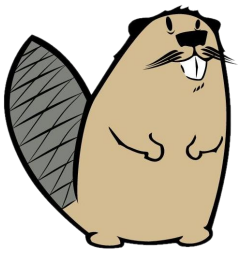
Sajnos Bob nem tud úszni, ő is csak a hidakon tud közlekedni a szigetek között.

A híd építéséhez Bobnak mindig meghatározott számú rönkre van szüksége. A falujában csak 3 rönköt tud felvenni, hogy elkezdje az építést. Bármelyik híddal összekötött szigetről további rönköket szedhet fel. Az alábbi térkép mutatja, hogy hány rönk szükséges egy híd építéséhez az adott átkelőnél és hogy hány rönk található az egyes szigeteken.

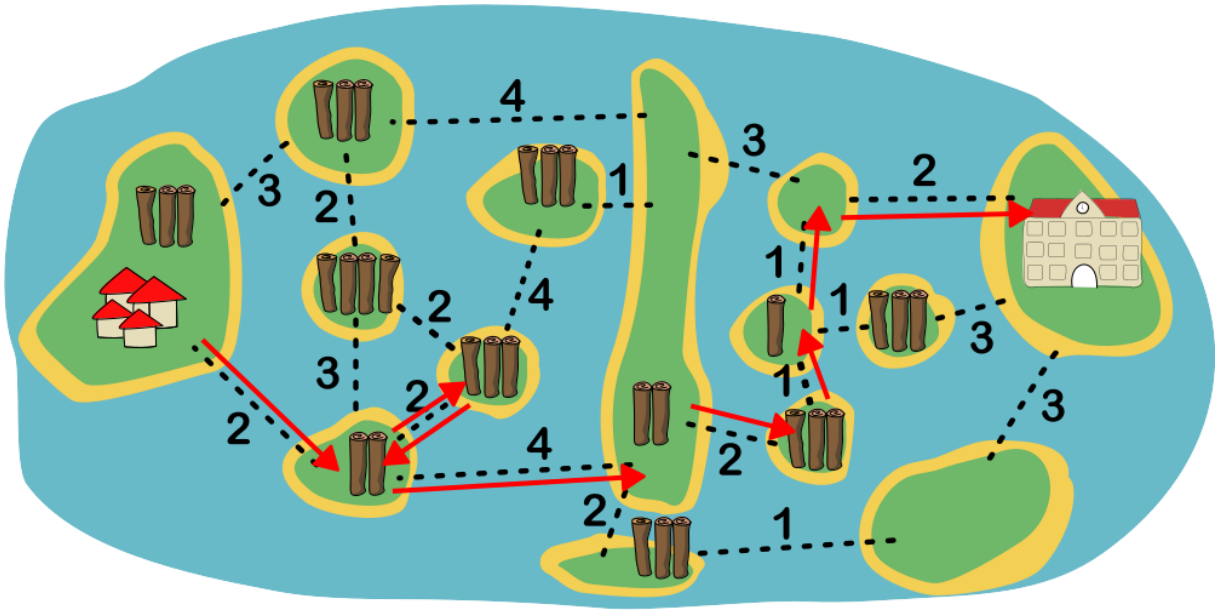
Minden rönk csak egyszer használható fel, de nem kell az összes rendelkezésére álló rönköt felhasználnia.



Melyik hidakat építse meg Bob, hogy a lehető legkevesebb rönköt használja fel, és hogy a diákok eljuthassanak a faluból az iskolába?



A megoldás:



A helyes útvonal megvalósítható (minden lépésnél a felvett rönkök száma >= felhasznált rönkök), ha minden lépésnél feljegyezzük a felvett és felhasznált rönkök összesített számát: (3,2)(5,4)(8,4)(8,8)(10,10)(13,11)(14,12)(14,14).

Bizonyíthatjuk, hogy ez a minimális út, ha megkeressük az összes olyan utat, amely a falut az iskolával köti össze, és amely <= 14 használt rönkkel rendelkezik, és megmutatjuk, hogy csak ez az eset megvalósítható. Először is vegyük észre, hogy minden útnak át kell haladnia a középen lévő nagy szigeten, így a problémát két részre oszthatjuk. Másodszor, a faluból a központba vezető útnak legalább 6 rönkfára, a központból az iskolába vezető útnak pedig legalább 5 rönkre van szüksége (ezeket nevezzük alsó korlátnak).

Először a bal oldali részt oldjuk meg a jobb oldali alsó korlát segítségével. Ez azt jelenti, hogy megkeressük az összes olyan megvalósítható utat, amely <= 9 rönköt használ (mivel a bal oldali rész bármely megoldása >= 10, összesen legalább 15 rönköt használna fel).

A bal oldalon a következő útvonalak vannak, amelyek a középponthoz kapcsolódnak és <= 9 rönköt használnak (felszedett rönkökkel, felhasznált rönkökkel ábrázolva):

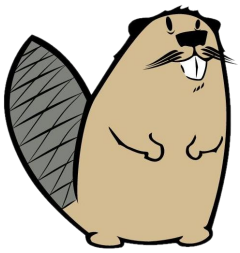
{A,C}=(6,7),{A,B,C}=(10,9),{O,Q}=(5,6),{O,H,Q}=(9,9),{O,P,Q}=(8,8),{O,P,J,F}=(11,9)

A megvalósíthatatlan megoldások eltávolításával (amelyeknél a felvett rönkök száma < a felhasznált rönkök) a következő marad:

{A,B,C}=(10,9),{O,H,Q}=(9,9),{O,P,Q}=(8,8),{O,P,J,F}=(11,9)

Ezután megoldjuk a jobb oldalt a bal oldali 8 használt farönk alsó korlátját használva, tehát megkeressük az összes olyan útvonalat, amely <= 6 farönköt használ:

{D,E}=(2,5),{D,G,E}=(3,6),{K,L,G,E}=(6,6),{R,S,T}=(5,6)



Most kombináljuk a bal és a jobb oldali rész megoldásait, hogy megtaláljuk azokat a megvalósítható párokat, amelyeknek <= 14 összesen felhasznált rönkje van, és csak {O,P,Q}=(8,8)+{K,L,G,E}=(6,6) → (14,14) találunk, mivel az összes többi pár felvett rönkje < felhasznált rönk.

MIÉRT INFORMATIKA?

A földtömegek és a potenciális hidak egy gráf csomópontjaiként és éleiként ábrázolhatók. Az egyes szárazföldi tömegeknél felszedett és az egyes hidak építéséhez felhasznált rönkök száma hozzárendelhető a megfelelő csomópontokhoz és élekhez. A Brian által bejárt útvonal az élek listájaként ábrázolható az építés sorrendjében.

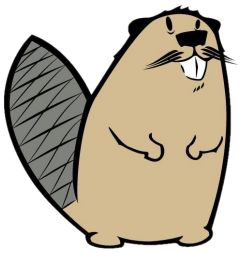
Ennek a problémának a megoldása általánosságban erőforrás-korlátozással rendelkező útvonaltervezési algoritmusok segítségével végezhető el. Az összes érvényes útvonal kimerítő keresése működne, de a gráf méretéhez képest exponenciális időt venne igénybe.

Alternatív megoldásként a probléma irányított oszd meg és uralkodj kereséseként is megfogalmazható, ahogyan azt a példamegoldás is teszi. Ez a megközelítés a kényszeroptimalizálás olyan technikáit használja, mint az útvonal-konzisztencia és az elágazás-és-korlátozás. Ezek a technikák képezik az alapját a legnehezebb valós világbeli erőforrás-elosztási problémák, rejtvények (mint a Sudoku!) és a játékot játszó mesterséges intelligenciák megoldásainak..

WEBOLDALAK

https://www.cs.bme.hu/~kiskat/algel/alg3-handout.pdf

https://hu.wikipedia.org/wiki/Divide_et_impera_(informatika)

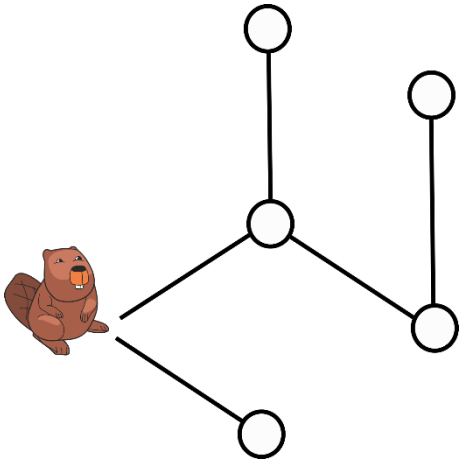


TAMÁS ÉS BARÁTAI (2023-PE-02)

BENJAMIN – NEHÉZ
KADÉT – KÖZEPES
JUNIOR – NEHÉZ

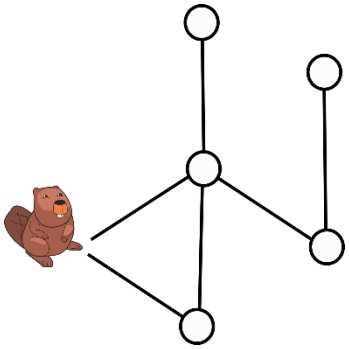
A faluban, ahol Tamás lakik összesen hat ház van, köztük Tamásé is. Gyalogosan ugyanannyi időbe telik végigmenni minden egyes útszakaszon, amely az egyik háztól közvetlenül a másikig vezet.

Tamás készített egy térképet azokkal a legrövidebb útvonalakkal, amelyek az ő házától a többiek házához vezetnek. A térképen nem jelölte be a falu összes útszakaszát, hanem csak azokat, amelyek tőle indulva a legrövidebb útvonalakba tartoznak.

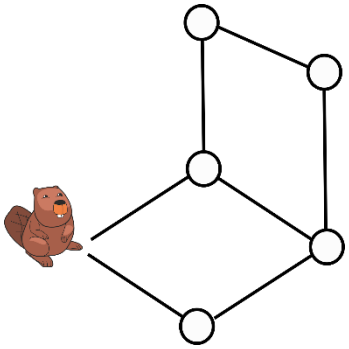


Több lehetséges térkép létezik a falu összes útszakaszáról, amelyekből Tamás ugyanazt a saját térképet rajzolhatta. **Az alábbi lehetőségek közül melyik nem lehet a falu térképe?**

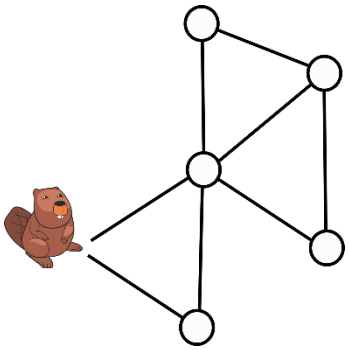
A)



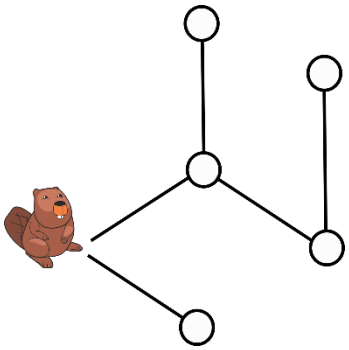
B)

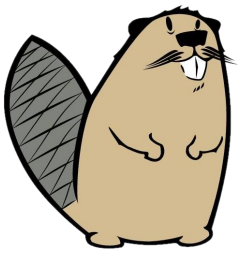


C)



D)





A helyes válasz a C)

A saját térképe szerint Tamásnak három útszakaszra van szüksége ahhoz, hogy eljusson a jobbra legtávolabbi házhoz. Ha a C a falu térképe lenne, akkor Tamás rövidebb úton, mindössze két útszakasszal is eljuthatna ahhoz a házhoz. Emiatt a C nem lehet a falu térképe.

Minden más térkép esetében nincs rövidebb útvonal, amely Tamás házától a többi házhoz vezetne.

MIÉRT INFORMATIKA?

Tamás informatikus, aki a térképét gráfként rajzolta meg. A gráfok csúcsokból (itt házak) állnak, amelyeket élek (itt az útszakaszok) köthetnek össze. Alkalmasak a valóság modellezésére, például egy város útvonalakkal összekötött pontjainak ábrázolására.

Tamás tudja, hogy a gráfokkal kapcsolatban sok algoritmus létezik, például a szélességi bejárás, amellyel az olyan feladatokat lehet megoldani, mint például "Keresd meg a legrövidebb utat a barátodhoz". Valószínűleg a személyes falutérképét egy nagyobb gráfon, a falu teljes térképén szélességi bejárással hozta létre.

A gráfelméletben azt mondjuk, hogy Tamás térképe a falu teljes gráfjának részgráfja. Tamás részgráfjának két különleges tulajdonsága van:

- Minden csúcs közvetlenül (egy élen keresztül) vagy közvetve (több élen keresztül) kapcsolódik egymáshoz, azaz összefüggő gráf.

- Bármely két csúcs között mindig pontosan egy útvonal van.

Az ilyen tulajdonságokkal rendelkező gráfot az informatikában feszítőfának nevezik. Ebben a gráfban Tamás házának helye a fa gyökere. A gyökérből Tamás az összes többi csúcsot (a többi házat) egyetlen útvonalon érheti el.

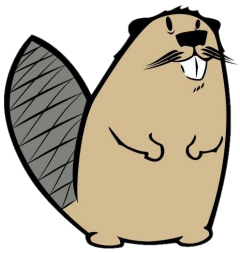
A gráfelméletnek számos valós alkalmazása van, elsősorban a hálózatokkal (közlekedési hálózatok, távközlési hálózatok...) kapcsolatban, például a navigációs rendszerekben az útvonalak kiszámításában.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Fesz%C3%ADt%C5%91fa

https://hu.wikipedia.org/wiki/Gr%C3%A1felm%C3%A9let

https://hu.wikipedia.org/wiki/Gr%C3%A1f



POSTFIX JELÖLÉS (2023-RO-02)

JUNIOR – NEHÉZ
SENIOR – KÖZEPES

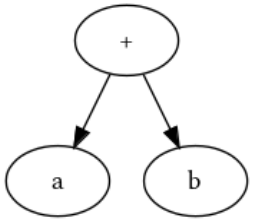
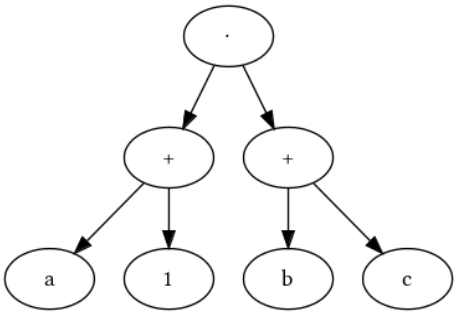
Egy matematikai kifejezés a következőkből áll

- egy operátorból, például "+", "−", " ·" vagy "**:**",
- és az operandusokból, amelyek lehetnek számok, mint például 1, 2, ... vagy kifejezések, mint például (1 + 2).

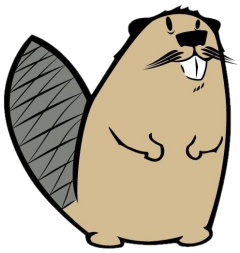
Egy matematikai kifejezés szerkezete tehát jól ábrázolható az operátorok és operandusok diagramjaként. Az ilyen diagramot szintaxisfának is nevezik.

Egy szintaxisfából viszont előállítható egy matematikai kifejezés postfix jelölése. Ebben a jelölésben először az operandusokat írjuk le, majd az operátort.

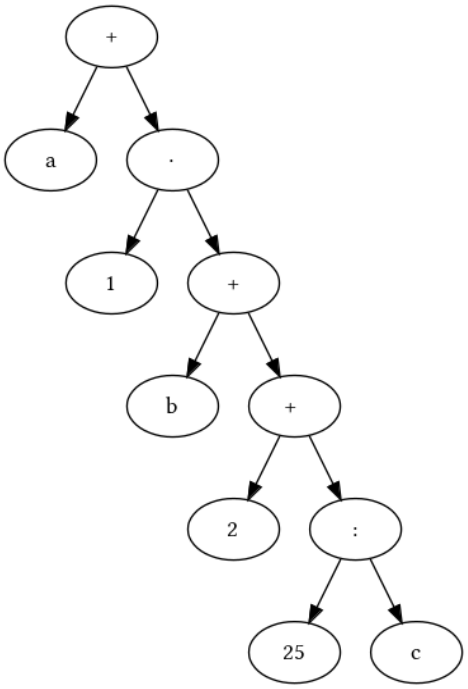
A táblázatban a következő két matematikai kifejezésből származó szintaxisfák és postfix jelölések láthatóak:

Matematikai kifejezés	`a + b`	`(a + 1) · (b + c)`
Szintaxisfa		
Postfix jelölés	`a b +`	`a 1 + b c + ·`

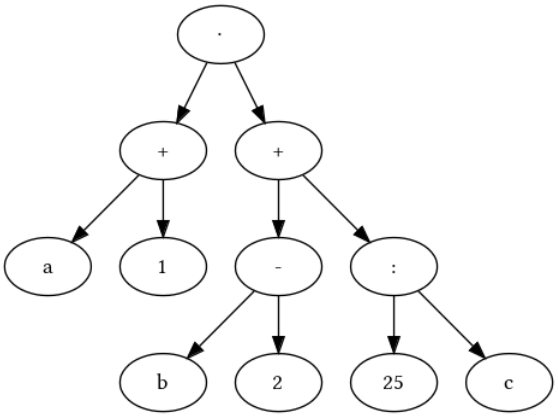
A következő oldalon lévő szintaxisfák közül, melyik írható le " a 1 + b 2 + · 25 c : + "postfix jelöléssel?



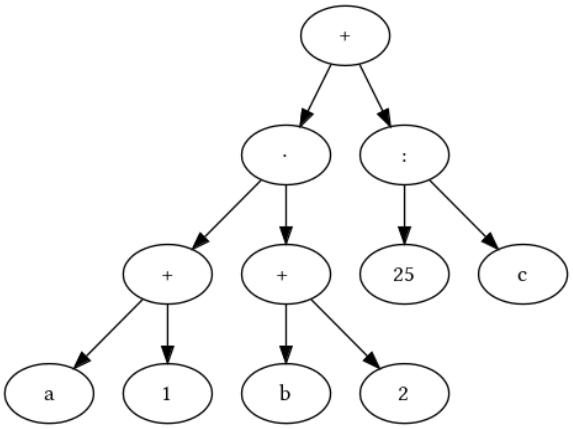
A)



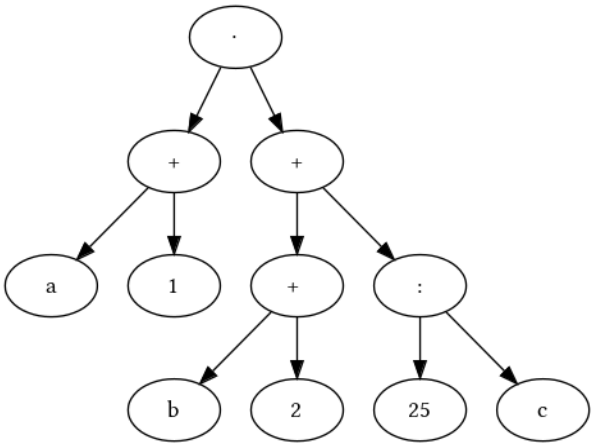
B)

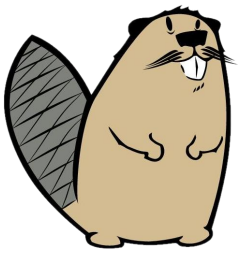


C)



D)





A helyes válasz a C)

Amikor a postfix jelölést visszaalakítjuk a szintaxisfára, először az utolsó karaktert, a "+" operátort kell megnéznünk. Ennek a fa tetején kell lennie, mert ez a legutolsó karakter. Ez azt jelenti, hogy csak az A) vagy a C) válasz lehet jó megoldás.

A "+" operátorhoz két operandus tartozik, egy bal és egy jobb oldali. A jobb oldali operandus esetében az egyetlen lehetséges következő karakter a fában a ":", mert ez az utolsó előtti karakter az utótagú jelölésnél. Ez azt jelenti, hogy csak a C) válasz lehet helyes.

De vajon helyes-e? Természetesen lépésről lépésre tovább lehetne konvertálni a postfix jelölést. De egyszerűbb a szintaxisfát lépésről lépésre átalakítani a postfix jelölésbe:

A legalsó három "legkisebb", egyenként 3 elemből álló részfa a 1 +, b 2 + és 25 c : lesz.

E három "legkisebb" részfa bal oldali kettője lesz a felső "+" bal oldali operandusa, így a bal oldali részfa most a 1 + b 2 + -. A harmadik "legkisebb" részfa már a jobb oldali operandus.

Így a válasz posztfix jelölése a 1 + b 2 + - 25 c : +, ami a feladat kifejezése.

MIÉRT INFORMATIKA?

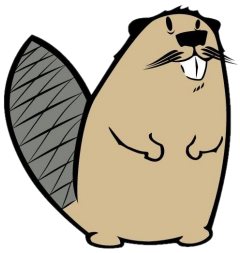
A postfix jelölést, fordított lengyel formának (RPN) is nevezik és gyakran használják matematikai kifejezések vagy általában kifejezések félreérthetetlen megfogalmazására (pl. programozási nyelvekben). Ha a C) válasz szintaxisfájára "normál" jelölést írnánk, akkor zárójeleket kellene használnunk: (a + 1) - (b + 2) + 25 : c. Erre a postfix jelölésnél nincs szükség. A postfix jelölés formát Jan Łukasiewicz (1878-1956) vezette be először, de még prefix jelölésként (az operátor a két operandus előtt áll, ahogyan azt f(x, y) függvényekből vagy programozásból ismerhetjük függvénynév(argument1, argument2, argument3) néven).

Korábban sokan ismerték és használták ezt a jelölést, különösen az első tudományos zsebszámológépekben, amelyekkel gyorsan, megbízhatóan és mindenekelőtt, zárójelek nélkül lehet bonyolult matematikai kifejezéseket kiszámítani. Még ma is van egy elszánt közösség, amely postfix jelöléssel ellátott zsebszámológépeket (például a HP 35-ösöket) használ😊.

A számítógépekben többek között egy programozási nyelv kifejezéseinek elemzésekor használják.

WEBOLDAL

https://hu.wikipedia.org/wiki/Ford%C3%ADtott_lengyel_jel%C3%B6l%C3%A9s

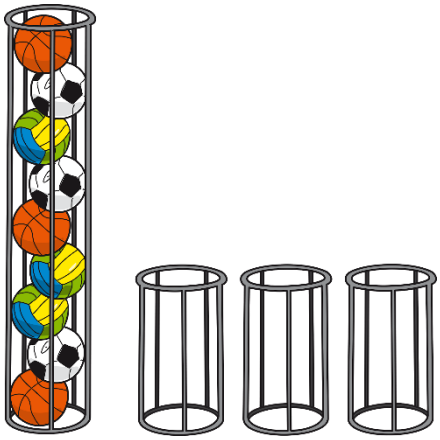


LABDARENDEZÉS (2023-SA-01)

KISHÓD – KÖZEPES
BENJAMIN – KÖNNYŰ

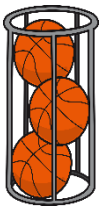
Sára előtt egy nagy tárolóban három különböző típusból összesen 9 labda van. Minden labdát a vele azonos típusúakkal egy tárolóba szeretne átrakni.

Ehhez Sára egyesével kiveszi a labdákat a tárolóból, és sorban a kisebb tárolókba teszi őket.



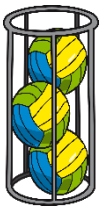
Ha a labdák eredetileg a fenti kép szerint vannak elhelyezve, melyik kisebb tárolót fogja Sára először megtölteni?

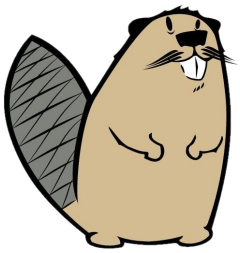
A)



B)

C)

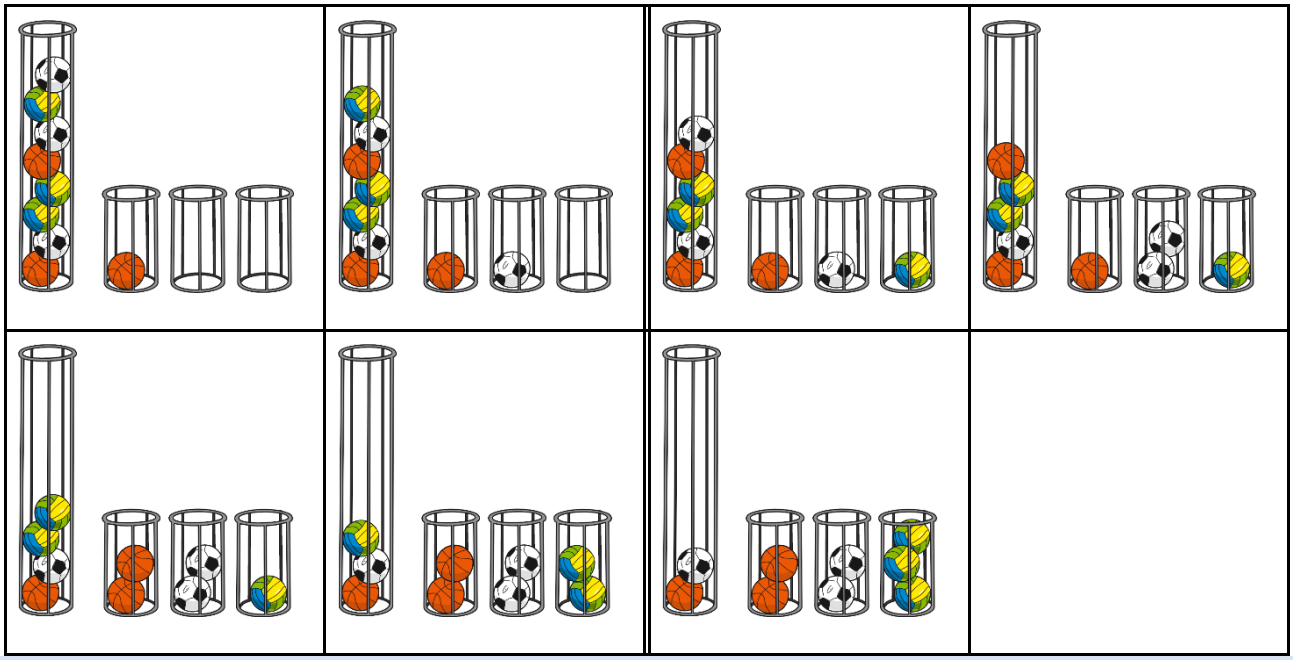




A helyes megoldás a C)

A nagy tárolóból Sára egyenként tudja kivenni a labdákat, fentről lefelé haladva. Az elsőnek megtöltött kisebb tároló akkor telik meg, amikor az adott típus harmadik labdáját teszi bele.

A harmadik ugyanolyan típusú labda, amelyik először fog előkerülni a nagy tárolóból, az a röplabda (C lehetőség).



MIÉRT INFORMATIKA?

Sára a labdákat a tároló tetejéről tudja csak kivenni. Az informatikában ezt az adatszerkezetet veremnek nevezik. A verem egy lineáris adatszerkezet, amelyet adatok tárolására használnak: csak a verem tetejére tudunk elemeket elhelyezni (push) vagy onnan kivenni (pop).

Számos példát tudunk a való életben mutatni a verem szerkezetre: például egy tányér- vagy könyvhalom, ahol a tetején lévő elemet vehetjük el először, míg az alján lévő elem lesz az utolsó elem, amelyet levehetünk, és mindig a tetejére tesszük a következő elemet.

De az informatikusok is szeretnek veremmel dolgozni és sokszor használják például kifejezések kiértékelésére, a helyességük, helyes sorrendjük ellenőrzésére.

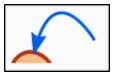
WEBOLDAL

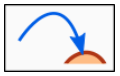
https://hu.wikipedia.org/wiki/Verem_(adatszerkezet)

RÉPAÜLTETÉS (2023-SK-02)

KISHÓD – NEHÉZ
BENJAMIN – KÖZEPES
KADÉT – KÖNNYŰ

A nyuszirobot a következő utasításokat tudja végrehajtani:

 balra ugrik a következő dombra

 jobbra ugrik a következő dombra

 elültet egy magot azon a dombon, amin éppen van

A nyuszirobot a következő utasításokat hajtja végre az ábrák sorrendjében:

Ezzel magot ültetett valahol ezen a négy dombon:

Nem tudjuk, hogy melyik dombról indult, de azt tudjuk, hogy egy dombon soha nem ültet el egynél több magot.

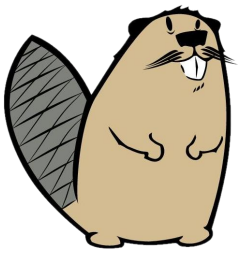
Melyik rajz mutathatja a dombokat az elültetett magokkal, ha a nyuszirobot az utasításokat már végrehajtotta?

A) 

B) 

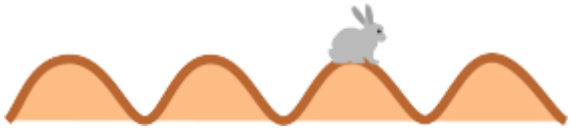
C) 

D) 



A helyes válasz a D)

A nyuszirobotnak balról a harmadik dombról kellett indulnia, különben nem tudna a megadott utasítások szerint ugrani, azaz egyszer jobbra, majd háromszor balra.



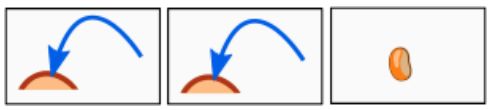
Az első két utasítás végrehajtása után:

A nyuszirobot a legjobboldalibb dombon elültet egy magot:

Majd végrehajtva a következő két utasítást:

A magok helye a következő:

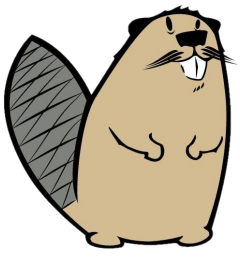
Majd kétszer balra ugrik és elülteti az utolsó magot is:

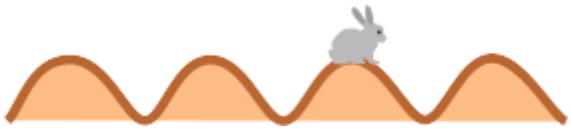


A magok így a D) válaszban megjelölteknek megfelelően helyezkednek el:

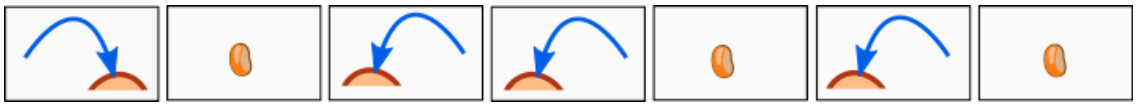
Az A) válasz akkor lenne helyes:

ha a nyuszirobot innen indulna





és a következő utasításokat hajtaná végre:



A B) válaszhoz:

a nyuszirobotnak innen kellene indulnia:

és az utasítások például ezek lennének:

A C) válasz esetén:

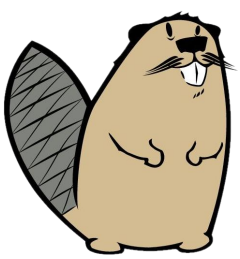
a nyuszirobot például az első dombról indulna:

és a következő utasításokat hajtaná végre:

MIÉRT INFORMATIKA?

A számítógépeket is utasításokkal programozzák, akárcsak a robotnyulat. Egy erre használt számítógépes program általában több utasításból áll.

Esetünkben a robot utasítássorozatát képkockák (ikonok) segítségével adtuk meg. A program kimenete (eredménye) nem csak a kiindulási helyzettől (bemenet), hanem az utasítások sorrendjétől is függ.

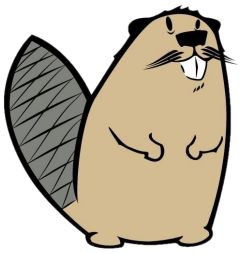


Ez a hód feladat mutat egy példát a robotok mezőgazdasági felhasználására is. Ezek a robotok nemcsak ültetni, hanem öntözni, beporozni, permetezni is tudnak.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%B3g%C3%A9p-programoz%C3%A1s

http://technika.gmgi.hu/uploads/termek_1501/robotok_a_mezogazdasagban_18_10.pdf



KÖZEL VAGY TÁVOL (2023-SK-07)

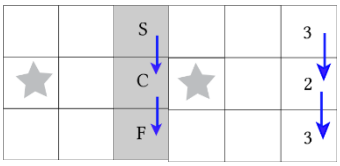
BENJAMIN – NEHÉZ
KADÉT – KÖZEPES
JUNIOR – KÖNNYŰ

Anna egy kincset rejtett el Daninak egy négyzetekből álló játéktérben.

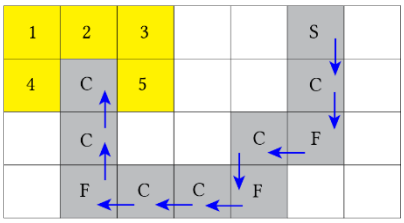
Kezdetben Dani az S négyzeten áll. Egyszerre egyetlen lépést tehet vízszintesen vagy függőlegesen egy szomszédos négyzetre. Minden megtett lépése után Anna megmondja neki, hogy közelebb ("K") vagy távolabb ("T") van-e a kincshez. A kincs távolsága a megteendő lépések száma (vízszintesen és függőlegesen), azon az úton, amelyiken a legrövidebben juthatunk el a kincshez.

A következő 3*3 négyzetrácsot tartalmazó példában Anna a kincset a csillaggal jelzett négyzetben rejtette el.

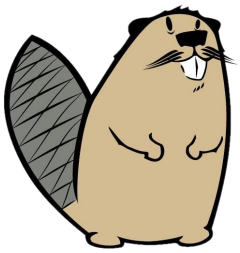
Dani az S-nél indul és két lépést tesz a nyilak mentén. A bal oldali képen látható, hogy Anna mit mond neki minden lépés után. A jobb oldali kép mutatja Dani kincshez viszonyított aktuális távolságát.



Most Dani egy nagyobb, 4*7 négyzetrácsot tartalmazó játéktéren van és a nyilak mentén halad. Anna megmondja neki, hogy a kincs az öt sárga négyzet valamelyikében van. A képen az is látható, hogy Anna az egyes lépéseknél mit mond neki.

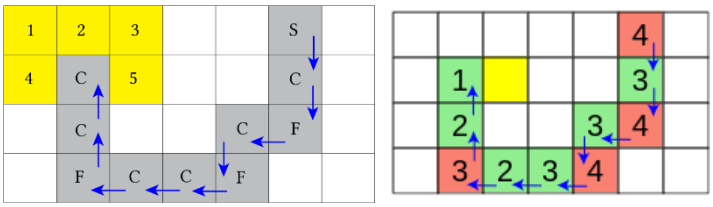


Melyik sárga négyzetben található a kincs?



A helyes válasz az ötös (5) számú sárga négyzet

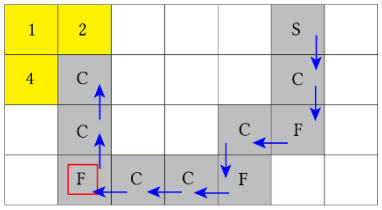
Ellenőrizhetjük a válasz helyességét, ha Anna jelzéseit követve (K vagy T) minden négyzetbe – amit Dani meglátogat – beírjuk a megoldástól, vagyis az ötös sárga négyzettől való távolságot:



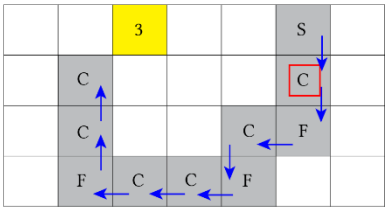
Minden olyan lépésben, amelyben a távolság nőtt, Anna visszajelzése valóban T volt. Azokban a lépésekben, amelyekben a távolság kisebb lett, Anna visszajelzése mindig K volt. Anna visszajelzései és a kincs tényleges távolságai megegyeznek. Ebből arra következtetünk, hogy a kincs az ötössel jelölt négyzetben lehetett.

A másik négy sárga négyzet esetében mindig találunk legalább egy olyan visszajelzést Annától, amely ellentmond a távolságnak.

Ha a kincs az egyes, kettes vagy négyes négyzetben van, akkor a negyedik sor második oszlopában (az ábrán jelölt helyen) Dani közelebb került a kincsekhez, bár Anna azt állítja, hogy távolabb. A kincs nem lehet e három négyzet egyikében sem.

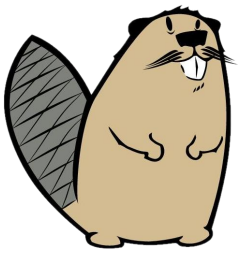


Ha a kincs a hármas négyzetben van, akkor a második sor hatodik oszlopában (az ábrán jelölt helyen) Dani távolabb került a kincstől, bár Anna azt állítja, hogy közelebb. Így a kincs nem lehet ebben a négyzetben sem.



MIÉRT INFORMATIKA?

Ebben a hód feladatban egy célt adnak meg, és arra kérnek minket, hogy oldjuk meg a feladatot megadott nyomok listája alapján. Ez a forgatókönyv a Reinforcement Learning (RL) nevű gépi tanulási technikára hasonlít. Az RL rendszerek alapvető összetevői egy szereplő és egy játéktér, amin mozogva a szereplő valamely stratégia alapján döntéseket hoz, és egy visszajelzés, amit minden egyes döntés után kap. A visszajelzés az aktuális állapotot vagy annak változását méri.



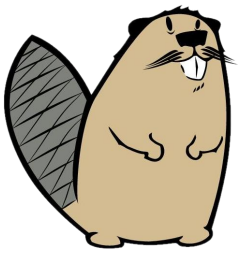
Ebben a feladatban a játéktér a négyzetrács az elrejtett kinccsel. A nyilak azt mutatják, hogy a játékos, Dani milyen döntéseket hozott. A visszajelzés pedig megfelel Anna "K" és "T" visszajelzésének.

Két pont közötti távolságon általában a pontok közötti szakasz hosszát értjük, ez az euklideszi távolság. Ebben a feladatban a távolságot más szemszögből nézzük és a Manhattan távolságot számoljuk ki. Az elnevezés onnan származik, hogy New York Manhattan nevű kerületében minden utca vagy vízszintesen, vagy függőlegesen fut, és az ezeken az utcákon megtett út hossza mutatja a két ház közti távolságot.

WEBOLDALAK

https://hu.wikipedia.org/wiki/T%C3%A1vols%C3%A1g

https://enaris.org/material/hu/Reinforcement%20Learning/index.html



HEGYMÁSZÁS (2023-TW-04)

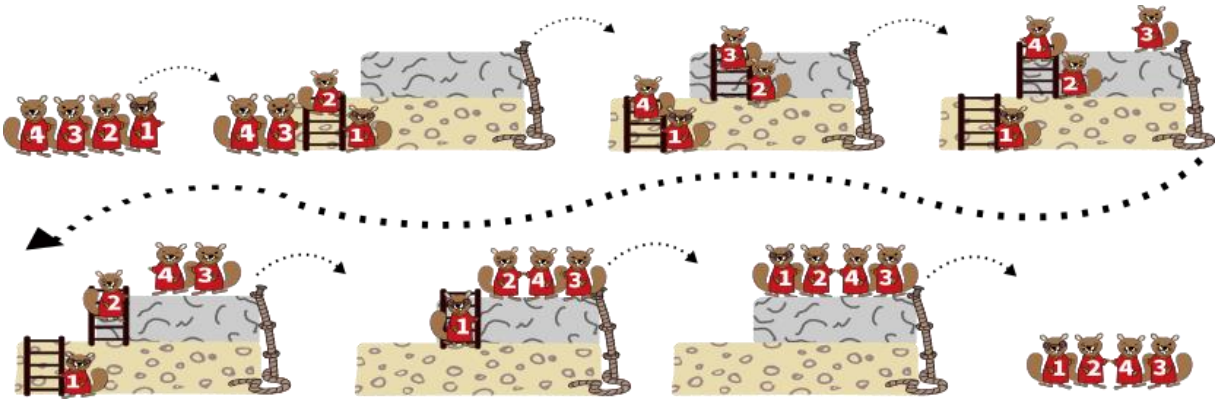
JUNIOR – NEHÉZ
SENIOR – KÖZEPES

A hódok a következő túrájukra gyakorolnak. Felsorakoznak egymás után és libasorban gyalogolnak. Minden hódnál van egy létra. Amikor egy hód elsőnek ér el egy megmászandó sziklafalhoz, megtámasztja a létráját, hogy a többiek felmászhassanak rajta, szépen sorban.

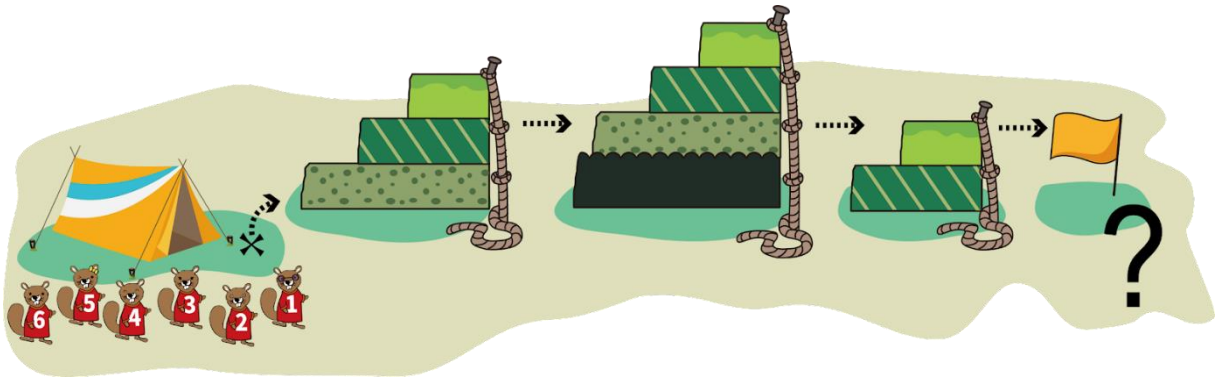
Minden hód, aki a létráját a sziklafalnak támasztotta, a helyén marad és tartja a létrát mindaddig, amíg az összes többi hód, akinek ő tartott létrát fel nem mászik a hegy tetejére. Minden hód, aki létrát tartott, mindvégig a saját létráját használva, utoljára mászik fel a hegy tetejére. Így folytatják a mászást, amíg fel nem érnek mindannyian a hegy tetejére.

Amikor befejezik a mászást, egy kötélen lecsúsznak a hegyről, a sorrendjüket megtartva.

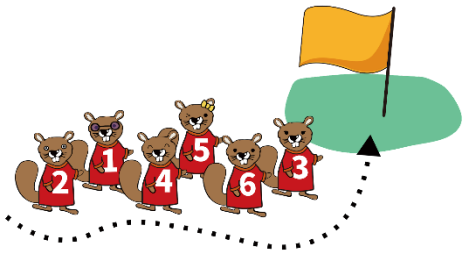
Az alábbi ábra azt mutatja meg, hogyan mászik fel négy hód a két sziklafalon át a hegyre.



Hat hód kirándulni indul a fent leírt módon a következő ábrán látható terepen. Határozd meg a hódok, túra végére kialakuló sorrendjét a zászlótól nézve!

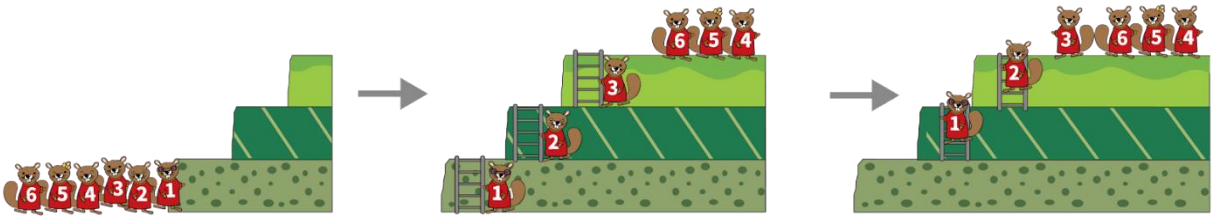


A helyes sorrend a zászlótól nézve: 365412

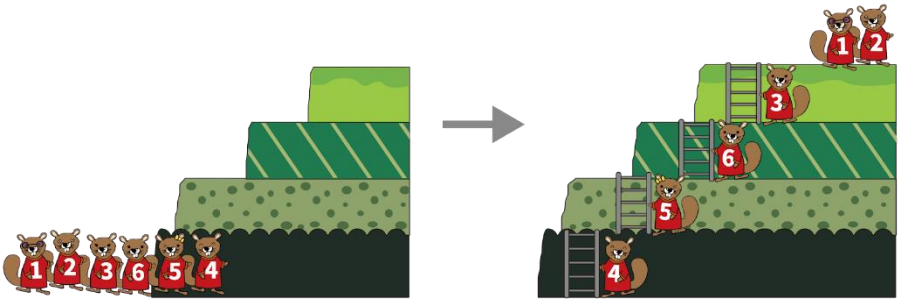


A hódok eredeti sorrendje: 123456. Ebben a sorrendben érkeznek meg az első hegyhez, amely 3 szintből (sziklafalból) áll. Az 1-es hód az 1-es szinten tartja a létrát, a 2-es hód a 2-es szinten, a 3-as hód a 3-as szinten, míg a 4-es, 5-ös és 6-os hódok ebben a sorrendben másznak fel a hegy tetejére.

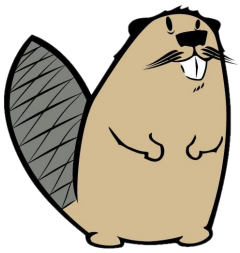
Miután a többi hód felért a hegy tetejére, a 3., a 2. és az 1. hód ebben a sorrendben mászik fel, amíg mindannyian fel nem érnek a hegy tetejére. A sorrendjük a mászás után 456321.

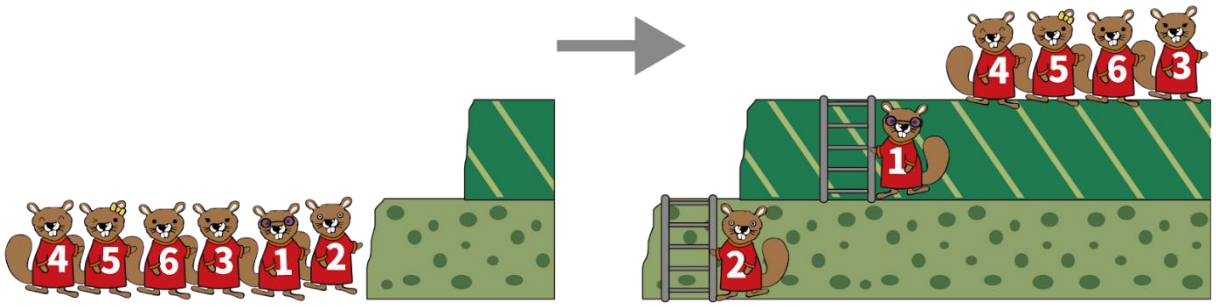


Miután lecsúsznak a hegy végén levő kötélen, a hódok a 456321 sorrendben érkeznek meg a második hegyhez, amin 4 sziklafal (szint) van. A sorban első hódként a 4. számú hód az 1. szinten tartja a létrát, az 5. sorszámú hód a 2. szinten, a 6. sorszámú hód a 3. szinten, a 3. sorszámú hód a 4. szinten. Így a 2. sorszámú és az 1. sorszámú hód ebben a sorrendben mászik fel a második hegy tetejére. Miután a 2. és az 1. felért a csúcsra, a 3., a 6., az 5. és a 4. mászik fel, ebben a sorrendben, amíg mindannyian fel nem érnek a második hegy tetejére. A sorrend most 213654.

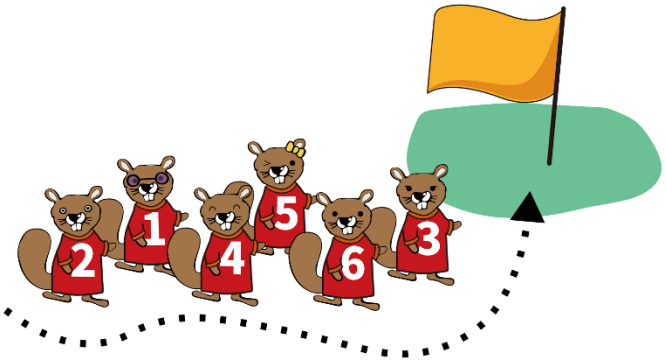


A hódok lecsúszás után tehát a 213654 sorrendben érkeznek az utolsó hegyhez. Ez a hegy2 szintes. A sorban első hódként a 2. hód az 1. szinten tartja a létrát, az 1. hód pedig a 2. szinten, míg a többi hód a 3654 sorrendben mászik fel a hegy tetejére. Miután a többi hód felért a hegy tetejére, az 1. és a 2. hód ebben a sorrendben mászik fel.





Az utolsó hegy megmászása után a hódok sorrendje tehát 365412. Ez azt jelenti, hogy a kötélen való lecsúszás után a táborba érkező hódok sorrendje ugyanez, vagyis 365412.



MIÉRT INFORMATIKA?

Ahogyan a hódok ebben a feladatban a hegyekre felmásznak, az informatikában a verem és a sor adatszerkezetek felépítését mutatja. A hódok a létrákat a verem műveletével állítják fel, és a hegyre sorban másznak fel.

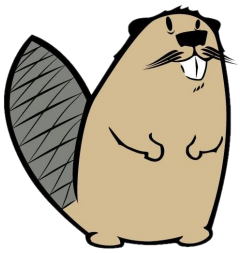
A verem és a sor kétféle adatszerkezet. Az adatok tárolása egy veremben pont olyan, mint ahogyan a valóságban egy gödörbe tesszük be és vesszük ki a dolgokat: egy objektumot úgy adunk hozzá a veremhez, hogy a verem tetejére helyezzük. Az egyetlen elérhető objektum az, amelyik a verem tetején van, azaz az utolsónak behelyezettet tudjuk elsőként kivenni. Ezt a sorrendet úgy hívjuk, hogy LIFO (last in first out), ami annyit jelent, hogy "utoljára be, elsőnek ki".

Másrészt az adatok sorban történő tárolása olyan, mint amikor az emberek sorba állnak például a pénztárnál: az első személy a sorban lesz az első, aki fizethet. Más szóval, ha egy sorból egyesével vesszük ki az adatokat/tárgyakat, akkor az első sorba helyezett adat/tárgy lesz az első, amelyik kikerül a sorból, és az utolsónak elhelyezett lesz az utolsó (a sor végén). Ezt a sorrendet úgy is hívjuk, hogy FIFO (First in first out), ami annyit jelent, hogy "először be, elöször ki".

WEBOLDALAK

https://hu.wikipedia.org/wiki/Verem_(adatszerkezet)

https://hu.wikipedia.org/wiki/Sor_(adatszerkezet)



SZÁMZÁR (2023-UA-01)

JUNIOR – NEHÉZ
SENIOR – KÖZEPES

Bob ajtaján egy számzár van. A nyitáshoz be kell írnia egy kódot, ami jelenleg öt számjegyből áll és így néz ki: 0 2 4 3 1

Bob titkosítva jegyzi fel a kódot: n >> c azt jelenti, hogy a kódban a c számjegy bal oldalán pontosan n számjegy van, amely nagyobb, mint c. Például a kódban 1 >> 3, azaz a 3-as számjegy bal oldalán pontosan egy számjegy van (nevezetesen 4), amely nagyobb, mint a 3. A jelenlegi kódot tehát a következőképpen jegyezte le:

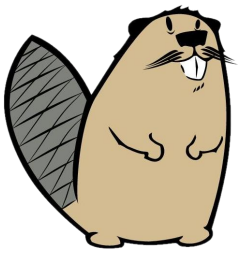
0 >> 0; 3 >> 1; 0 >> 2; 1 >> 3; 0 >> 4

A csak öt számjegyből álló kód azonban túl gyenge Bob számára. Ezért kitalál egy új kódot, amely a 0-tól 7-ig terjedő számjegyekből áll. Az új kódot így jegyzi le:

3 >> 0; 2 >> 1; 4 >> 2; 4 >> 3; 1 >> 4; 1 >> 5; 1 >> 6; 0 >> 7

Mi az új kódja? Húzd a számokat a helyükre!

0, 1, 2, 3, 4, 5, 6, 7



A helyes kód a: 74105623

A kód meghatározásához nézzük meg egyesével Bob jelöléseit:

			0				

`3 >> 0`: A 0-tól balra pontosan 3 számjegy van, amely nagyobb a 0-nál, tehát a 0-nak a kód negyedik helyén kell állnia (hiszen a 0 a legkisebb)

`2 >> 1`: Az 1-től balra pontosan 2 olyan számjegy van, amely nagyobb, mint 1. Az 1-es számjegynek tehát a kód harmadik helyén kell állnia.

		1	0				

`4 >> 2`: A 2-től balra pontosan 4 számjegy van, amely nagyobb, mint 2. Mivel a 2-nél kisebb 1 és 0 számjegyek már a harmadik és negyedik helyet elfoglalják, a 2-nél nagyobb számjegyeknek az első, második, ötödik és hatodik helyen kell állnia. A 2-es számjegynek tehát a kód hetedik helyére kell kerülnie.

		1	0			2	

`4 >> 3`: A 3-tól balra pontosan 4 nagyobb számjegy van, mint a 3, ezért a 3-as számjegynek a kód nyolcadik, azaz utolsó helyén kell állnia.

		1	0			2	3

`1 >> 4`: A 4-től balra pontosan 1 számjegy van, amely nagyobb, mint 4. A 4-es számjegynek tehát a fennmaradó számjegyek közül a második helyen kell lennie; ez a kód második számjegye.

	4	1	0			2	3

`1 >> 5`: Az 5-től balra pontosan 1 számjegy van, amely nagyobb, mint 5. Az 5 számjegynek tehát a fennmaradó számjegyek közül a másodiknak kell lennie; ez a kód ötödik számjegye.

	4	1	0	5		2	3

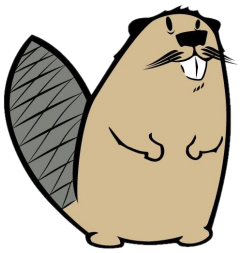
`1 >> 6`: A 6-tól balra pontosan egy számjegy van, amely nagyobb, mint 6. A 6-os számjegynek tehát a fennmaradó számjegyek közül a másodiknak kell lennie; ez a kód hatodik számjegye.

	4	1	0	5	6	2	3

`0 >> 7`: Nincs 7-nél nagyobb számjegy. A 7-es számjegynek az utolsó szabad pozícióban, azaz a kód első pozíciójában kell lennie.

7	4	1	0	5	6	2	3

Természetesen másik számjegytől is kezdhettük volna a vizsgálódást. Az eredmény ugyanez lenne, de ott időnként a pontos hely még nem meghatározható, csak a sorrend alakul folyamatosan.



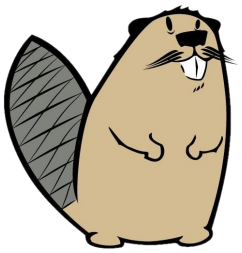
MIÉRT INFORMATIKA?

A feladatot megoldhattuk volna úgy is, hogy előállítjuk az összes lehetséges kódot és megnézzük, melyikre igazak a feltételek (illetve, hogy melyikkel nyitható a zár). A legtöbb számkód feltörésére – főleg, ha nem állnak rendelkezésre a szabályok – ezt a módszert szokták használni. Ezt „nyers erő" (brute force) módszernek is szokták nevezni.

Ez természetesen sokkal hosszabb és fáradságosabb munka, mint amit mi használtunk a titkosítási szabály ismeretében: a logikus levezetés és a számok „helyezgetése" a szabályaink értelmezésével. Tulajdonképpen egy rendezést végzünk a számokon, kihasználva, hogy minden számjegy csak egyszer fordulhat elő a kódban, és a szabályok megadják az összefüggéseket: melyik számjegy hova kerül a többihez képest.

WEBOLDAL

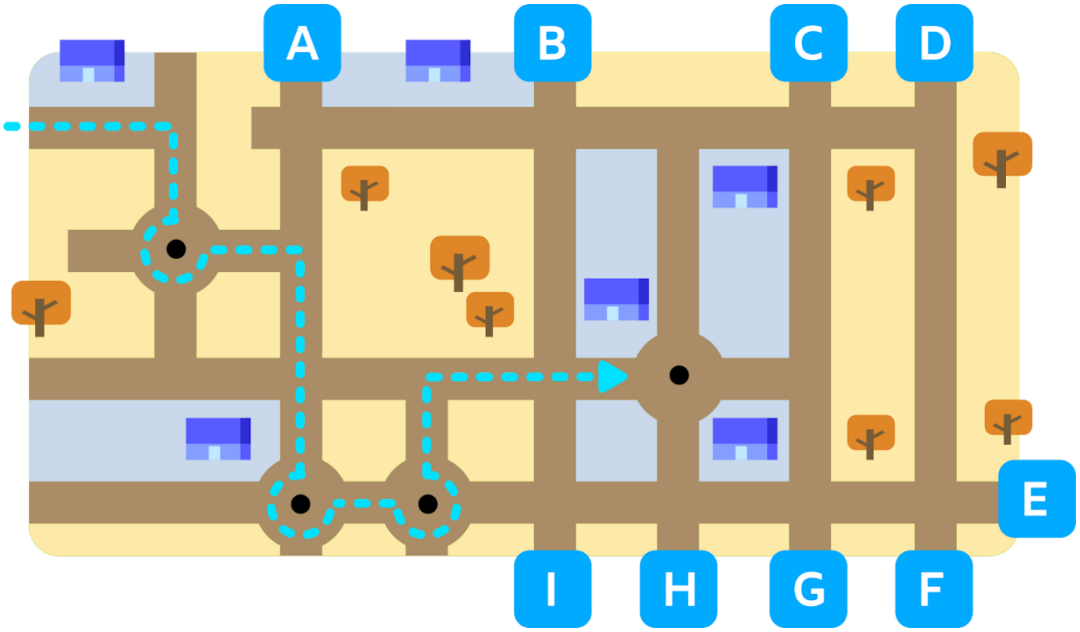
https://hu.wikipedia.org/wiki/Brute_force-t%C3%A1mad%C3%A1s



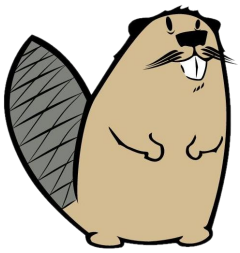
ÖNVEZETŐ AUTÓ (2023-US-03)

KISHÓD – NEHÉZ
BENJAMIN – KÖZEPES
KADÉT – KÖNNYŰ

Tomi az önvezető autójával utazik. Az autó nemrégiben frissített szoftverében egy komoly hiba van, ami miatt az autó minden alkalommal, amikor ugyanolyan típusú kereszteződéshez ér, ugyanabba az irányba (ugyanúgy) fordul. Ahogy azt a kép is mutatja. Például az autó minden körforgalomban a harmadik kijáratot választja.



Melyik betűhöz jut el Tomi, ha folytatja útját az önvezető autójával?



A helyes válasz az F)

A képből megállapítható, hogy a navigációs rendszer a következő szabályokat követi az egyes kereszteződés-típusok esetében:

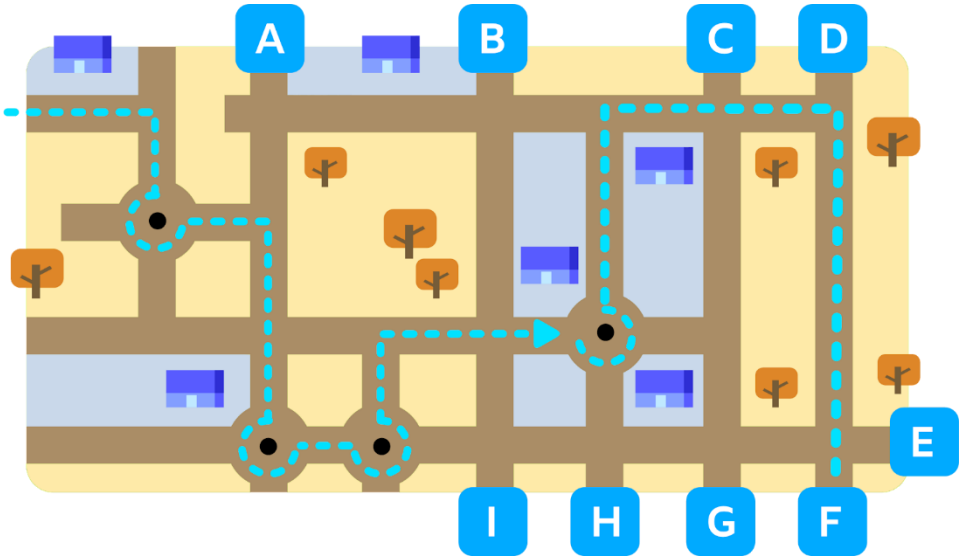
A T alakú kereszteződésben az autó mindig jobbra fordul.

Körforgalomban az autó mindig a harmadik kijáratot választja, azaz a kereszteződést az eredeti irányhoz képest balra hagyja el.

A kétirányú, + alakú kereszteződésben az autó mindig egyenesen halad tovább.

A következő kereszteződés egy körforgalom, tehát az autó a harmadik kijáraton megy ki, az ábrán felfelé. A következő egy T-alakú kereszteződés, tehát az autó jobbra fordul. A következő, + alakú kereszteződésben egyenesen megy tovább, majd a T alakú kereszteződésben – a térkép jobb felső sarkánál – újra jobbra fordul. Végül az autó egyenesen megy át a + alakú kereszteződésen, és eljut az F-fel jelölt pontba.

Az alábbi térképen a leírt útvonal látható.

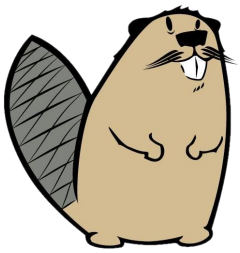


MIÉRT INFORMATIKA?

Az informatikában sokszor előforduló helyzet, hogy egy rendszert – előre meghatározott szabályokat követő – algoritmus irányít. A rendszer jövőbeli viselkedése az algoritmus elemzésével és a követett szabályok pontos meghatározásával jósolható meg. A vizsgált rendszer lehet egy valóságos rendszer vagy egy szoftverrendszer is.

Az autonóm biztonsági rendszerek kiemelt fontosságúak az önvezető autókban. Ezek a rendszerek nemcsak a vezető nélküli vészmegállást teszik lehetővé, hanem a vezető felügyelete alatti autonóm sávtartást is biztosítanak. Ezek a funkciók segítenek a járműnek biztonságosan reagálni különböző helyzetekre, még akkor is, ha a vezető nem aktív módon felügyeli az autót.

Az önvezető autók terjedését jelenleg a mesterséges intelligencia hiánya gátolja, különösen a megfelelő biztonsági szint elérése terén. A fejlett mesterséges intelligencia kulcsfontosságú a



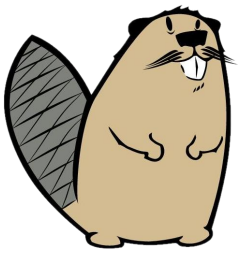
járművek helyes értelmezésében, döntéshozatalban és reakciókban, és ennek fejlesztése szükséges a teljes körű önvezetés biztosításához.

WEBOLDALAK

https://hu.wikipedia.org/wiki/%C3%96nvezet%C5%91_aut%C3%B3

https://hu.wikipedia.org/wiki/Szoftver

https://hu.wikipedia.org/wiki/Mesters%C3%A9ges_intelligencia



KINCSESLÁDÁK (2023-UY-01)

BENJAMIN – KÖZEPES
KADÉT – KÖNNYŰ

Egy szigeten három kincsesláda található: Az egyik láda a hegyek lábánál, a második egy pálmafa alatt, a harmadik pedig a tengerparton van.

Kora reggel az összes kincsesláda üres volt, de valamikor a nap folyamán megjelent Hódszakáll kalóz és az egyik ládát megtömte arannyal.

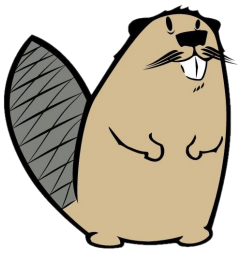
Három turista – Anita, Brigitta és Cecil – ugyanazon a napon, de különböző időpontokban fedezte fel a szigetet. Mindegyikük készített egy-egy fotót, amelyen egy vagy két láda látható.

Tudjuk, hogy:

- Az egyik turista azelőtt készítette el a fotót, hogy Hódszakáll az aranyat az egyik ládába tette volna.
- A másik két turista azután készítette a fotót, amikor már Hódszakáll elhagyta a szigetet.

	Anita fotóján csak a **tengerparton lévő kincsesládát** látjuk. Ez a láda üres.
	Brigitta fotóján két kincsesládát látunk, egyet a **pálmafa alatt** és egyet a **tengerparton**. Mindkettő üres.
	Cecil fotóján is két kincsesládát látunk, egyiket a **pálmafa alatt**, a másikat a **hegy lábánál**. Mindkettő üres.

Hódszakállnak szerencséje volt, mert egyik turista sem találta meg az aranyát. De melyik kincsesládában lehetett az arany?



Az arany a hegy lábánál levő kincsesládába került

ivel az arany három lehetséges helyen lehet, sorban ellenőrizhetjük mindegyikre, hogy a képek megfelelnek-e a történetnek.

1. lehetőség: Az arany a hegy lábánál levő kincsesládában volt.

A kincsesládát a hegyek lábánál csak Cecil fotóján látjuk, üresen. Cecil tehát még Hódszakáll megérkezése előtt elkészítette a fényképét. Anita és Brigitta fotóján csak a pálmafa alatt és a tengerparton látjuk a ládákat, ahol mindkettő üres. Mindkét fotó Hódszakáll látogatása után is készülhetett. A történet szerint az egyik turista még azelőtt készítette a fényképét, hogy Hódszakáll megtöltötte volna a dobozt, a két másik turista pedig utána. A történet és a mi gondolataink arról, hogy mikor készült a három fotó, nem mondanak ellent egymásnak.

2. lehetőség: Az arany a pálmafa alatti kincsesládában volt.

Brigitta és Cecil fotóin azt látjuk, hogy a pálmafa alatti kincsesláda üres. Ha abban van a kincs, akkor mindkét fotó azelőtt készült, hogy Hódszakáll megtöltötte volna a ládát arannyal. Tudjuk azonban, hogy csak egy turista készítette a fényképét Hódszakáll szigetre látogatása előtt. Ebből az ellentmondásból arra következtetünk, hogy nem a pálmafa alatti ládában volt a kincs.

3. lehetőség: Az arany a tengerparton levő kincsesládában volt.

Anita és Brigitta fotóin azt látjuk, hogy a tengerparton lévő kincsesláda üres. Mindkét fotó tehát azelőtt készült, hogy Hódszakáll megtöltötte volna a ládát arannyal. Azonban tudjuk, hogy csak egy turista készítette a fényképét Hódszakáll látogatása előtt. Ebből az ellentmondásból arra következtetünk, hogy a tengerparton nem lehetett elásva aranykincs.

Mivel az aranynak a három láda valamelyikében kellett lennie, és csak az első lehetőség engedi ezt meg, arra következtethetünk, hogy az arany valóban a hegyek lábánál lévő kincsesládában volt.

MIÉRT INFORMATIKA?

Ez a hódfeladat a logikai érvelésről szól.

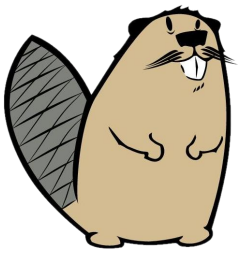
A három fényképet és a helyzetről szerzett ismereteinket használjuk fel arra, hogy megtaláljuk a három lehetőség közül miért lehet igaz vagy miért lehet hamis valamelyik.

Az ilyen logikai érvelésekben gyakran fontos szerepet játszanak az ellentmondások.

Ha logikus gondolkodással két olyan állítás is következik a helyzetből, amelyek nyilvánvalóan nem lehetnek egyszerre igazak, akkor kijelenthetjük, hogy a feltételezés, amelyből kiindultunk, hamis.

A logika rendkívül fontos szerepet játszik az informatika számos területén:

Például az olyan programnyelv, mint a Prolog, ahelyett, hogy egy jól definiált parancssorozatot dolgozna fel, állításokból indul ki (például: az egyik turista korábban készítette a fotót, melyik fotón mi látható, ...), és a megoldást (pl. az arany a hegyek lábánál lévő kincsesládában van) következtetések segítségével a kiinduló állításokból vezeti le. A levezetés itt nem számok összeadását vagy szorzását jelenti, hanem a meglévő ismeretekből automatikusan új állítások készítését. Ezt nevezzük logikai programozásnak.



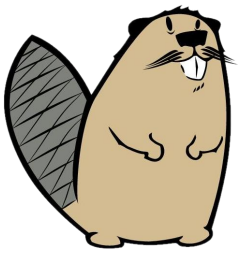
Az informatikában számos területen használunk logikai összefüggéseket: adatbázisok lekérdezéséhez bizonyos szempontok szerint; programozáskor az elágazások és ciklusok feltételeinek meghatározásához; hardver- és szoftvertervezésben és a szoftverellenőrzésekben is.

Mivel az aranynak a három doboz valamelyikében kell lennie, és csak a harmadik lehetőség vezethető le a kiinduló állításokból, arra következtethetünk, hogy az arany a hegyek lábánál lévő kincsesládában lehetett.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Funkcion%C3%A1lis_programoz%C3%A1s

https://hu.wikipedia.org/wiki/Prolog



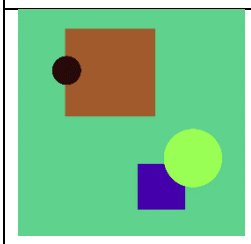
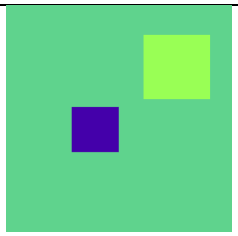
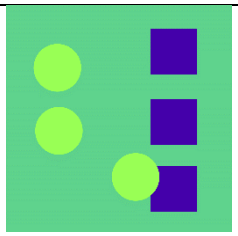
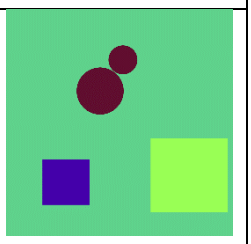
VÉLETLEN KÉPEK (2013-DE-02)

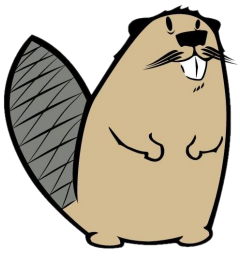
SENIOR – NEHÉZ

Egy gyárban a következő módszerrel készítenek csomagolópapírokat: A nyomtatógép színes köröket és négyszögeket tervez, majd ezeket papírlapokra nyomtatja. A gép minden alkalommal a következő utasításokat hajtja végre:

1) Tervezz egy kört, színezd ki egy véletlenszerűen kiválasztott színnel és nevezd el K-nak
2) Ismételd meg véletlenszerűen sokszor a következő 4 utasításból álló blokkot
   a) Tervezz egy véletlenszerűen kiválasztott színű és nagyságú négyszöget, majd nevezd N-nek
   b) Véletlenszerűen határozd meg K méretét NAGYnak vagy KICSInek
   c) Nyomtasd K-t egy véletlenszerűen kiválasztott helyre a papírlapon
   d) Nyomtasd N-t egy véletlenszerűen kiválasztott helyre a papírlapon.

Melyik papírlapot NEM ezzel a nyomtatógéppel készítették?

A	B	C	D
			



A helyes válasz az A)

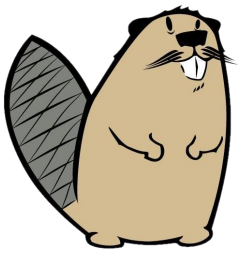
A nyomtatógép programja első lépésben kiválaszt egy kört, és a későbbiekben a körnek csupán a méretén változtat. Tehát a papírlapon az összes körnek ugyanolyan színűnek kell lennie. Ezen kívül a gép legalább egy kört nem egy négyszögre nyomtat, hiszen a nyomtatás sorrendje kör – négyszög – kör – négyszög. A program lefutása után a gép ugyanannyi kört nyomtat, mint négyzetet. Ugyanakkor a nyomtató készíthette a B papírlapot is, ha véletlenül a négyzeteket pont a körökre nyomtatta. Az is lehet, hogy a körök véletlenül a háttérrel azonos színt kaptak. A C és D képen ugyanannyi kör és négyzet található. A köröknek mindkét papírlapon azonos színe van, és legfeljebb két különböző méretűek. Ezeket a papírlapokat a nyomtatógép készíthette.

MIÉRT INFORMATIKA?

Napjainkban igen sok film használ számítógépek által automatikusan előállított képeket. Ilyenkor a véletlennek fontos szerep jut, ugyanis a tervezők legtöbbször csak absztrakt szabályokat határoznak meg a képek formálásához. Az összetett mesterséges struktúrák apró részleteit (fák ágait és leveleit, állatok bundájának színét és szőrszálait) számítógépek alkotják meg, az alkotás módja nincs meghatározva, de összhangban működik a megadott szabályokkal.

WEBOLDAL

https://hu.euronews.com/next/2019/07/29/mi-az-a-veletlenszam-es-miert-nehez-letrehozni



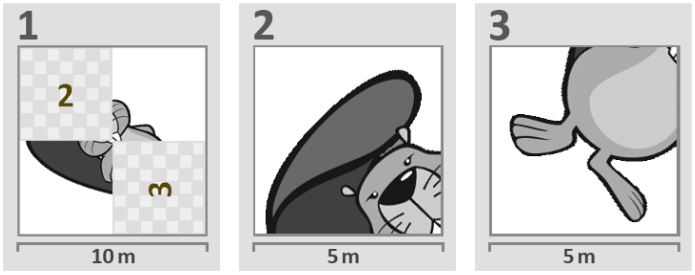
REKURZÍV FESTÉS (2016-AT-06)

SENIOR – NEHÉZ

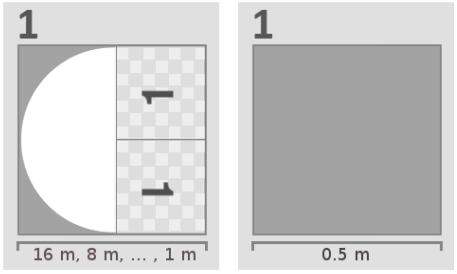
Tina és Ben segítenek egy különleges kiállítás előkészítésében az Informatikai Múzeumban. Az egyik kiállítóterem padlójára kell egy 16x16 méteres képet festeniük.

A művészektől egy utasításkártyát kapnak, amelyen a híres festőkártya-nyelven utalás szerepel a képelemekre, méretekre és forgatásokra. Némelyik utasításkártyán szerepelnek számozott mezők, amelyek további utasításkártyákra utalnak.

Itt egy példa egy korábbi projektből. Ha az ember helyesen hajtja végre ezt a három utasításkártyát, egy hód képe lesz látható:



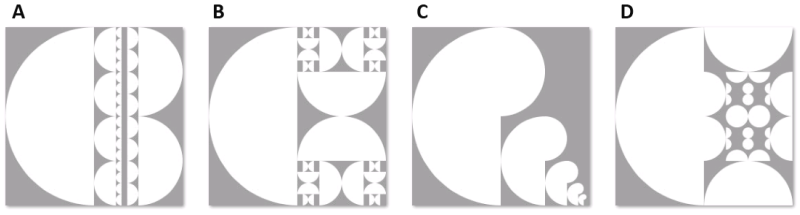
A különleges kiállításra Tina és Ben a következő festési utasításkártyákat kapják:

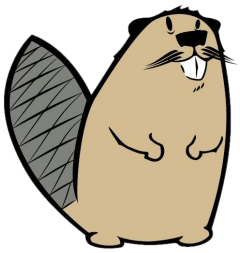


Ben ráncolja a homlokát. „Ez meg mit jelentsen? A bal oldali kártya önmagára utal vissza, ezen kívül mindkét kártyának ugyanaz a száma!"

Tina nevet: „Megküzdünk ezzel is! Először csak a bal oldali kártyát használjuk. A jobb oldali kártya később majd felvilágosít róla, mikor kell abbahagynunk a festést."

Hogyan fog kinézni a kiállítóterem padlója?





A helyes válasz a B)

A bal oldali utasításkártya arra utasít, hogy a padló bal felét egy félkör felülettel kell kitölteni, melynek kerek oldala balra mutat. A padló jobb felén ugyanezt az utasításkártyát kétszer kell alkalmazni. A képelemek fekvése meg kell feleljen az egyes számok irányának.

Az egyik képelemnél az egyes 90°-kal balra van forgatva. Ezért a képelemet szintén balra kell forgatni, a félkör görbülete befelé mutat. A másik képelemnél az egyes 90°-kal jobbra van forgatva. Ezért a képelemet szintén jobbra kell forgatni, a félkörfelület görbülete befelé mutat.

Egyedül a „B" válasznál áll fenn ez a helyzet.

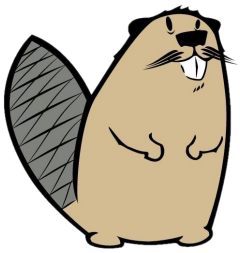
MIÉRT INFORMATIKA?

Az informatikában az olyan utasításokat, amelyek önmagukra hivatkoznak, „rekurzívnak" nevezik. A kifejezés a latin „recurrere" (magyar: visszafutni) szóból ered. A rekurzió egy hatásos elgondolás. Némely összetett feladat megoldására megfogalmazható rövid és átlátható rekurzív utasítás.

Egy rekurzív utasításnak tartalmaznia kell egy feltételt, amely meghatározza, mikor kell megtörni (abbahagyni) a rekurziót. Máskülönben a rekurzió mindaddig tovább dolgozik, amíg valamely erőforrás ki nem merül. Vagy a számítógép memóriája, vagy a felhasználó türelme.

WEBOLDAL

https://hu.wikipedia.org/wiki/Rekurzi%C3%B3



BONTSD RÉSZEKRE(2017-RU-05)

SENIOR – NEHÉZ

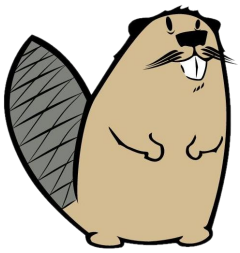
Egy különleges szövegkódoló rendszer minden betűt a 0 és 9 közötti számjegyekből készített kóddá alakít. A sajátossága az, hogy semelyik kód nem kezdődhet egy másik betű kódjával.

Például: ha az X betű kódja a 12, akkor az Y-t kódolhatja 2-ként. Így sem a 12 nem kezdődik 2-vel, sem a 2 12-vel. A Z kódja lehet a 11, mivel sem a 12 sem a 2 nem kezdődik 11-gyel és a 11 sem kezdődik 2-vel vagy 12-vel. A 21 már nem engedélyezett a Z kódolására, hiszen 2-vel kezdődik, ami viszont már az Y-hoz tartozó kód.

A BEBRAS szót a rendszer így kódolta:

1 2 1 1 2 2 3 3 3 2 1

Helyezz elválasztó vonalakat az egyes betűk kódjai közé!



A teljes BEBRAS szó kód felosztása csak a következő lehet:

1 – 21 – 1 – 22 – 33 – 321

A számsor elején kezdjük a szétbontást. Amennyiben a B-t 12-ként kódoljuk, az E betű szükségképpen az 1-es kódot kapja (ekkor utána ismét 12, mint B betű jön). Ez viszont ellentmond a kódkezdő szabálynak, vagyis a B betűt kódoló 12 az E betűt kódoló 1-essel kezdődne.

Nem lehet, hogy a B betűt több, mint 2 számjeggyel kódoljuk (121, 1211, 12112, stb.), mert még egyszer meg kellene ismétlődnie és ehhez nincs elég számunk. Ezért a B betűt kizárólag az 1-es kóddal láthatjuk el.

Az E betű után ismét B következik, ezért az E-t csak 2-re vagy 21-re kódolhatjuk (vagy 211223332-re). Nem lehet 2, mert akkor a kódolt szöveg BEBB-bel kezdődne. Nem lehet 211223332 sem, hiszen akkor az egész szó csak BEB lenne. Következésképp az E betűt a 21-es számként kódolhatjuk csak. Így tudjuk, hogy 1 21 1 a BEB kódolása. Még a hátralevő 2233321 kódrészt kell feldarabolnunk.

Tekintsük a szó végén álló S betűt. Nem lehet a kódja 1, sem 21, mivel ezek a számsorok már foglaltak a B és E betűk kódolásánál. Következésképp ezek lehetnek a lehetséges kódok az S-hez: 321, 3321, 33321, 233321 és 2233321. S-t nem kódolhatjuk 2233321-ként, hiszen akkor a szó mindössze BEBS lenne.

Ugyanilyen okból nem lehet 233321 sem, mert akkor is hiányozna még az R vagy az A betű (ha csak egy-egy számjeggyel kódolnánk is). Ha S-t 3321-ként kódoljuk, az RA párost 223-ként kell. Azonban az R nem lehet 2, és az A sem lehet csak 3, hiszen más kódszavak már kezdődnek ezekkel a számokkal. Ezek alapján az S csakis 321-ként kódolható. Az RA párosra marad a középső számsor: 2233. A már ismert okok miatt R-t 22-ként, A-t 33-ként kódoljuk.

MIÉRT INFORMATIKA?

A feladatban szereplő kódolási technikát előtag kódnak (prefix kód) nevezzük. Egy előtag egy karaktersorozat, mellyel egy másik karaktersorozat kezdődik. Egy előtag-kód esetében a kód előtagja nem lehet egy másik kód. Vagyis nem kezdődhet egy kód egy másik kóddal.

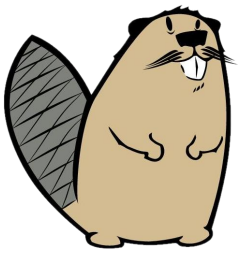
Az előtag kódoláskor a kódszavak különböző hosszúságúak. Az előtag kód előnye, hogy nincs szükség elválasztó jelekre a kódszavak között. Mindig felismerhető, hogy melyik helyen kezdődik a következő kódszó. Ha a gyakran előforduló betűket rövid kódszavakkal jelöljük, akkor egy szöveget hatékonyan kódolhatunk, sőt, ha nagyobb szövegről van szó, helytakarékosan tárolhatjuk azt.

A Huffman-kódolás egy módszer hatékony előtag kódolás keresésére. Szélesen elterjedt és olyan adatformátumok használják, mint pl. a JPEG és az MP3.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Titkos%C3%ADt%C3%A1s

https://hu.wikipedia.org/wiki/Huffman-k%C3%B3dol%C3%A1s

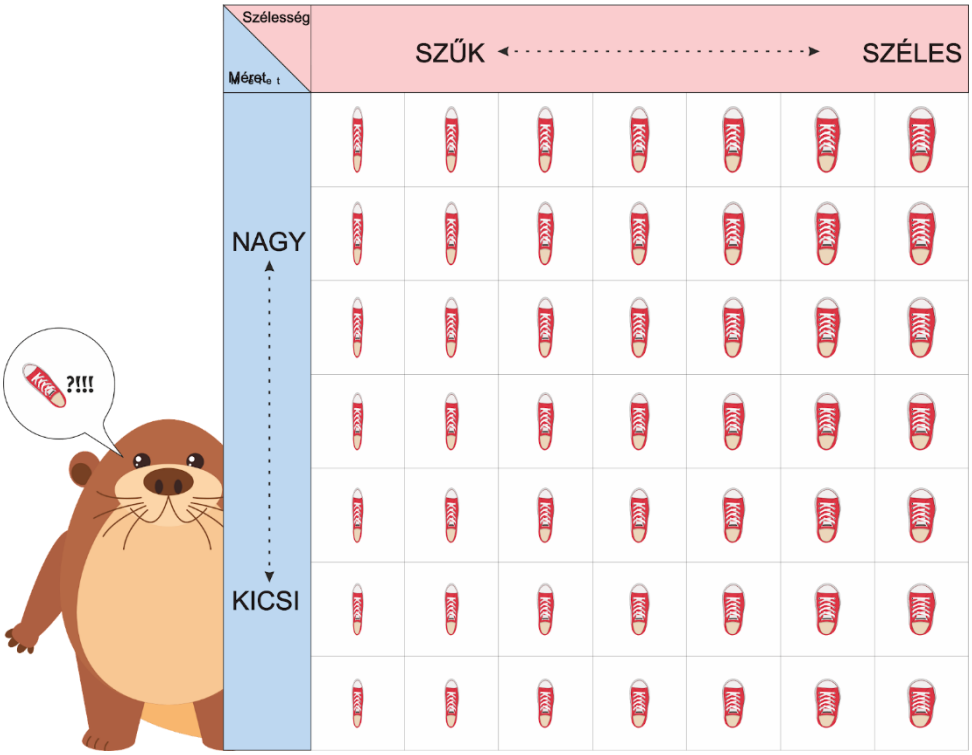


CIPŐVÁSÁRLÁS (2019-KR-04)

SENIOR – NEHÉZ

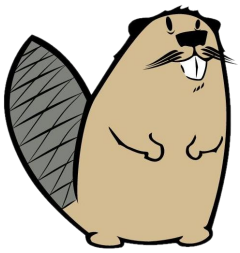
Kati, a hód elment a cipőboltba, hogy vegyen magának egy pár cipőt. A boltban a cipők a képen látható módon vannak elrendezve: méret és szélesség szerint is növekvő sorrendben. A cipők méret és szélesség szerint mind különbözőek, a legkisebb és legkeskenyebb cipő balra lent, a legnagyobb és legszélesebb cipő pedig jobbra fent található.

Mivel Kati feledékeny, nem tudja a saját cipőméretét, így addig kell a cipőket próbálgatnia, amíg meg nem találja a megfelelő méretűt és szélességűt.



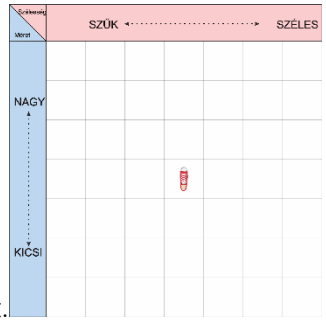
Kati kitalált egy olyan módszert, amely garantálja, hogy a lehető legkevesebb próbával megtalálja a rá illő cipőt.

Mennyi az a legkevesebb próba, amennyivel Kati biztosan megtalálja a megfelelő cipőt?



A helyes válasz a 2)

Katinak szerencséje lehet, és elsőre megtalálhatja a cipőt.



Azonban 2 cipő felpróbálása után biztosan megtalálhatja a megfelelőt.

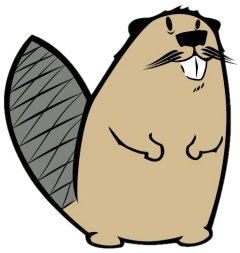
- A próbát a középen lévő cipővel kell kezdenie:

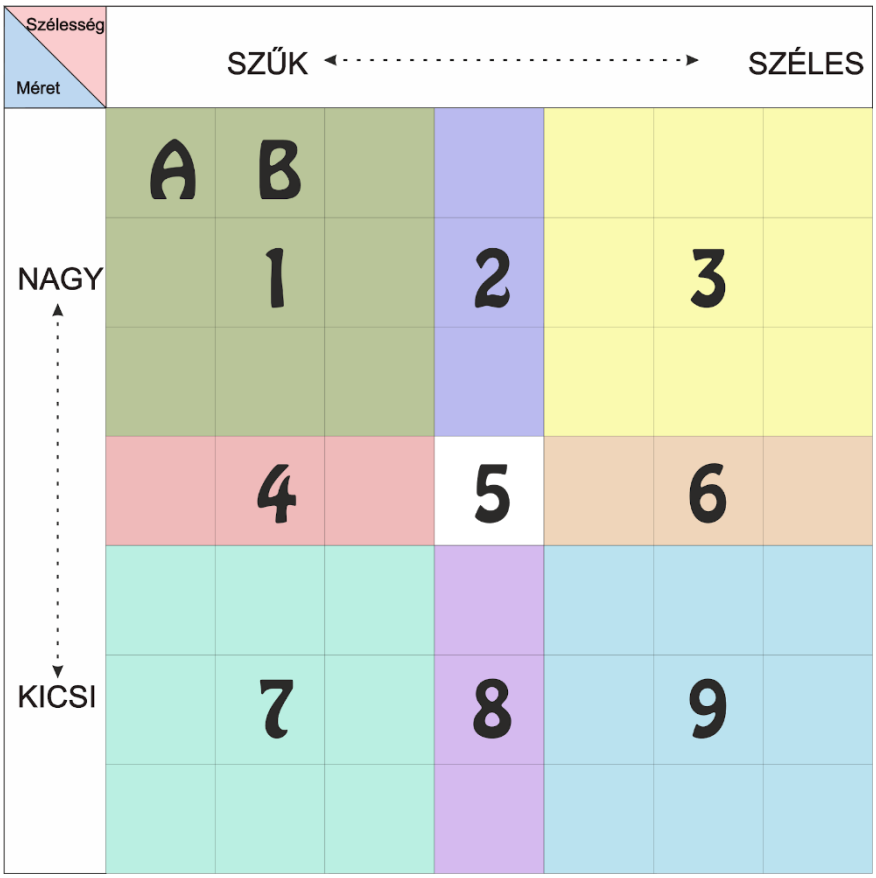
- A cipő vagy
    - a megfelelő méretű, vagy kisebb, vagy nagyobb és
    - vagy megfelelő szélességű vagy keskeny vagy széles lesz.

Ez alapján Kati már tudni fogja, hogy a neki megfelelő cipőt hol keresse:

- Ha a cipő pontosan illik rá, akkor megtalálta a megfelelő cipőt
- Ha a cipő kisebb és szélesebb, mint ami neki kell, akkor az 1-essel jelölt részben lévő cipők közt lesz a megfelelő.
- Ha a cipő kisebb, de a szélessége pont jó, akkor a 2-essel jelölt részben lévő cipők közt lesz a megfelelő.
- Ha a cipő nagyobb és keskenyebb, akkor a 9-essel jelölt részben lévő cipők közt lesz a megfelelő.

És így tovább.





Tegyük fel, hogy a cipő, amelyet Kati felpróbált, kisebb és szélesebb, mint ami neki kell. Tehát nagyobb és keskenyebb cipőt kell felpróbálnia, amit az 1-essel jelölt részben fog megtalálni.

Most felpróbálja az 1-essel jelölt rész közepén lévő cipőt.

- Ha a cipő illik rá, akkor megtalálta a megfelelő cipőt.
- Ha a cipő még mindig kisebb és szélesebb, akkor az A helyen lévő cipő lesz a megfelelő.
- Ha a cipő kisebb, de a megfelelő szélességű, akkor a B helyen lévő cipő lesz a megfelelő.
- És így tovább.

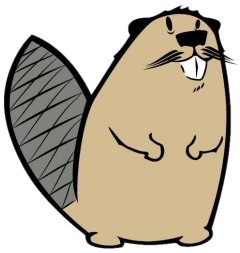
Így Katinak legfeljebb 2 cipőt kell felpróbálnia, hogy megtalálja a neki pontosan megfelelő szélességű és megfelelőméretű cipőt.

Ha más helyről veszi le kiindulásként az első próbához a cipőt, akkor lehet, hogy több cipőt kell felpróbálnia

MIÉRT INFORMATIKA?

A probléma a bináris keresésről szól, amellyel egy kívánt értéket találhatunk meg egy kétdimenziós rendezett térben.

A fenti feladatban a cipőboltban lévő cipők két dimenzióban, a méret és a szélesség szerint is növekvő sorrendben vannak elrendezve. Így használható a bináris keresés algoritmusa, ami



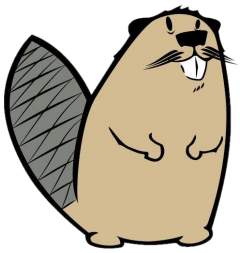
rendezett adathalmazokon működik, egy adott elem gyors megtalálására. A bináris keresés minden alkalommal mindkét dimenzióban felére csökkenti a keresett elemet tartalmazó részt.

A bináris keresési algoritmus használható akkor is, amikor egy számot kell kitalálni például 1 és 100 közt, és visszajelzést kapunk arról, hogy a szám nagyobb vagy kisebb, mint amire épp rákérdezünk.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Bin%C3%A1ris_keres%C3%A9s

https://hu.wikipedia.org/wiki/Keres%C5%91algoritmus



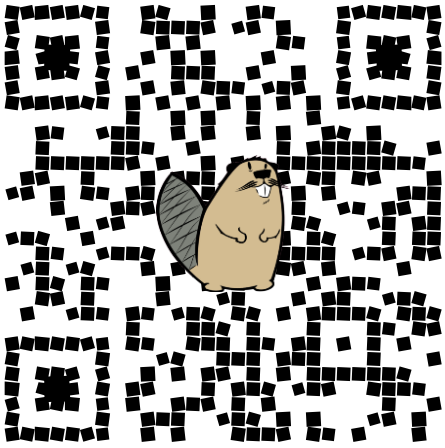
TÁMOGATÓINK, KÖSZÖNETNYÍLVÁNÍTÁS

Köszönjük a nemzetközi Bebras kezdeményezés országainak, kiemelten a DACH-csapatnak (Németország, Ausztria, Svájc),

az ELTE IK hallgatóinak, illetve

a kapcsolattartó tanároknak szervezői munkájukat.

A verseny kérdései Kutatási célokra csak a szervezők megkérdezésével, megkeresésével és kutatási munkájuk ismeretével, valamint idézésével használhatóak fel.



ELTE | IK
INFORMATIKAI KAR