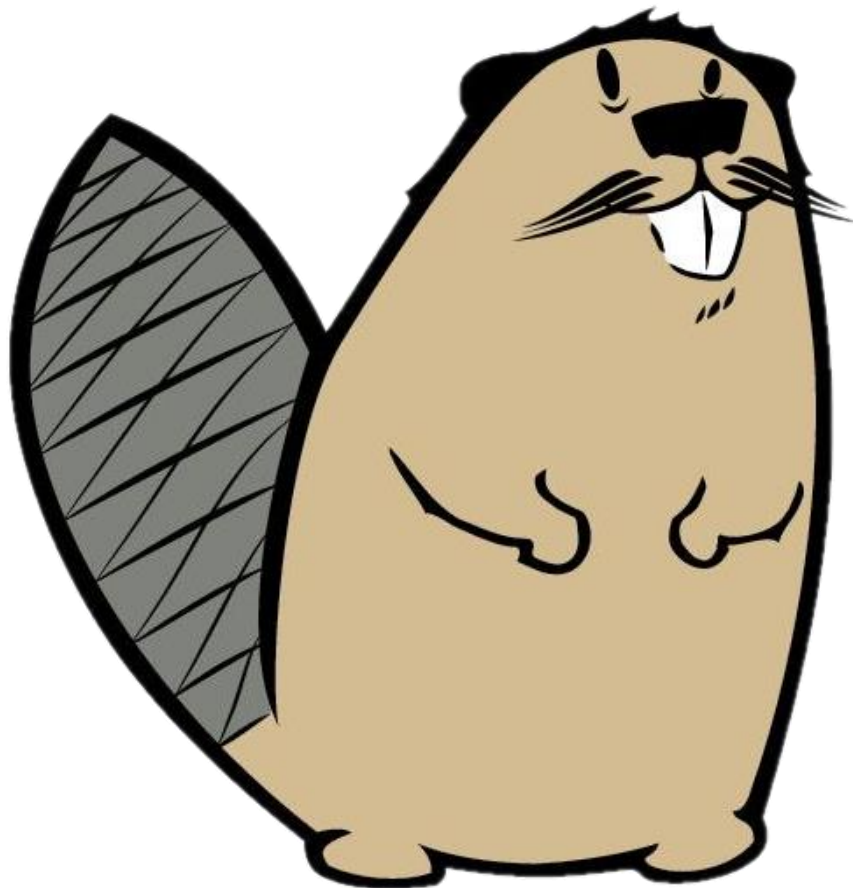


HÓDÍTSD MEG A BITEKET!

INFORMATIKAI GONDOLKODÁST TÁMOGATÓ, NEMZETKÖZI BEBRAS
KEZDEMÉNYEZÉS MAGYAR MEGVALÓSULÁSA



2022

MI IS AZ E-HÓD

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

A nemzetközi Bebras, melyhez 2021-ben már 50 ország kapcsolódott, 2015-ben elnyerte az Informatics Europe „Best Practices in Education” díját.

A kezdeményezés alapja Dr. Valentina Dagiene litván professzor által életre hívott 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 2022-ben tizenkettedik alkalommal, öt korcsoportban vehettek részt a diákok 4. osztálytól 12. osztályig.

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

Az alábbi dokumentumban a 2022-es magyar verseny feladatai és megoldásai találhatóak.

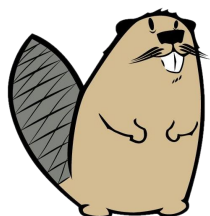
További információkért látogasson el a [HTTP://E-HOD.ELTE.HU/](http://e-hod.elte.hu/) weboldalra, vagy írjon email-t az info@e-hod.elte.hu címre.

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 14:00 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 válaszolják 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 továbbítani a szervezők felé;
- A verseny célja: minél több pont összegyűjtése helyes válaszok megjelölésével, helytelen válaszok esetén 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

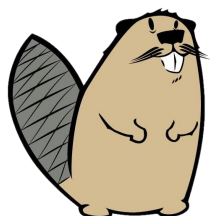
A Kishód korcsoportban 12, minden más korcsoportban 18 feladatot kell megoldani három nehézségi szinten. Minden helyes válasz pontot ér, minden helytelen válaszáért pontlevonás jár.

Nem megválaszolt kérdés esetében az összpontszám változatlan marad.

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

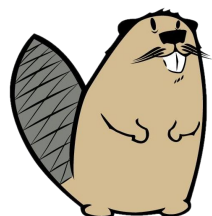
	Könnyű	Közepes	Nehéz
Helyes válasz	6 pont	9 pont	12 pont
Helytelen válasz	-2 pont	-3 pont	-4 pont

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

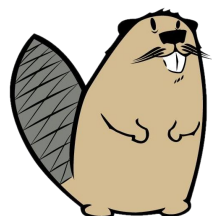


TARTALOMJEGYZÉK

Mi is az E-HÓD.....	2
Részvétel.....	2
Szabályok.....	3
Értékelés, Pontozás.....	3
Tartalomjegyzék.....	4
Feladatok.....	6
Kifestő (2022-AT-01a).....	7
Padlóminta (2022-AT-02)	9
Lista (2022-AT-04).....	11
Kapcsold fel az izzót! (2022-AU-03).....	13
Négy kártya (2022-BE-02).....	17
Varázsfölde (2022-CA-02)	21
Tömör megjelenítés (2022-CA-04).....	24
Csavarok és anyák (2022-CA-06)	27
Greedy trollok (2022-CH-04)	30
Ételrejtekhely (2022-CH-08).....	33
Kincsek a zsákban (2022-CH-14).....	36
Mari szomszédai (2022-CY-01).....	38
Ugróiskola (2022-CZ-04b).....	41
Szív grafika (2022-DE-02)	43
Okoszemüveg (2022-DE-03).....	46
Kerek hangár (2022-DE-05)	48
Filmnézés (2022-DE-07)	50
Méhkaptár (2022-FR-02).....	53
Antivírus (2022-HU-02).....	56
Vigyázz, galóca! (2022-HU-04)	58
Erdei party (2022-IE-02).....	61
Művészet hagyományosan (2022-IN-01)	64
Kagylók és kövek (2022-IT-02).....	66

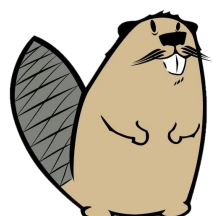


Hol van a hód (2022-JP-02).....	69
Hódburger recept (2022-KR-03)	71
Hódbajnokság (2022-KR-06).....	73
Ösvény a nagymamához (2022-LT-08)	75
Amőba (2022-LV-03)	78
Mária kincset keres (2022-MK-01).....	80
Csomagolás (2022-NL-03)	83
Vírus (2022-NZ-01).....	85
Teknős és nyúl (2022-PH-03)	88
Kreatív hímzőprogram (2022-SK-02)	90
Matrózláncok (2022-SK-03).....	92
Titkosírás 8 (2022-SK-04)	94
Szőnyegminta (2022-TR-02).....	96
Hódgát (2022-TW-06).....	99
Rendezés (2022-UA-03b)	101
Kedvenc cukorka (2022-IT-01b)	105
Szavak (2022-US-06).....	107
Varázslóiskola (2022-UZ-03).....	109
Hatszögfestés (2022-VN-05a)	112
Hatszögfestés (2022-VN-05b).....	114
Támogatóink, köszönetnyilvánítás	116



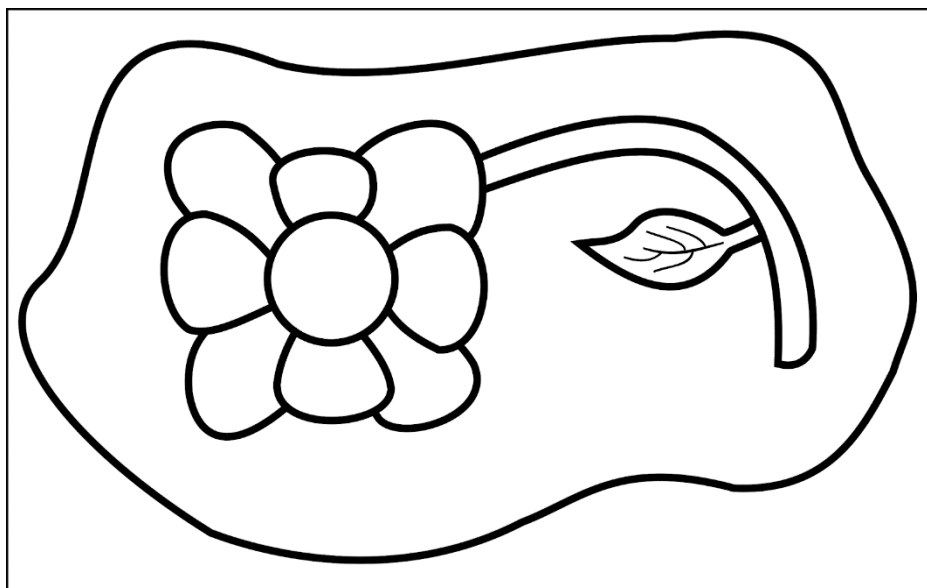
FELADATOK

- Kishód:** 2022-AT-01a, 2022-CH-14, 2022-DE-02, 2022-FR-02a
2022-IE-02, 2022-IN-01, 2022-KR-03, 2022-LT-08
2022-PH-03, 2022-SK-03, 2022-UK-02, 2022-VN-05a
- Benjamin:** 2022-AU-03, 2022-CA-06, 2022-CH-08, 2022-CH-14, 2022-CY-01, 2022-CZ-04b
2022-DE-02, 2022-FR-02a, 2022-HU-02, 2022-JP-02, 2022-KR-03, 2022-PH-03,
2022-SK-02, 2022-SK-03, 2022-SK-04, 2022-TR-02, 2022-TW-06, 2022-VN-05a
- Kadét:** 2022-AT-04, 2022-AU-03, 2022-CA-06, 2022-CH-04, 2022-CH-08, 2022-CY-01
2022-DE-03, 2022-DE-05, 2022-DE-07, 2022-HU-02, 2022-HU-04, 2022-KR-03
2022-LV-03, 2022-SK-02, 2022-SK-03, 2022-SK-04, 2022-TR-02, 2022-VN-05b
- Junior:** 2022-AT-04, 2022-AU-03, 2022-CA-04, 2022-CA-06, 2022-CH-04, 2022-CY-01
2022-DE-03, 2022-DE-05, 2022-DE-07, 2022-HU-04, 2022-IT-02, 2022-LV-03
2022-MK-01, 2022-NL-03, 2022-SK-04, 2022-US-06, 2022-UZ-03, 2022-VN-05b
- Senior:** 2022-AT-02, 2022-AT-04, 2022-BE-02, 2022-CA-02, 2022-CA-04, 2022-CH-04
2022-DE-03, 2022-DE-05, 2022-DE-07, 2022-IT-02, 2022-KR-06, 2022-LV-03
2022-MK-01, 2022-NL-03, 2022-NZ-01, 2022-UA-03b, 2022-US-06, 2022-UZ-03

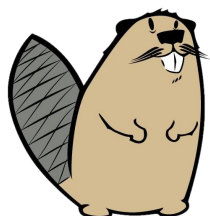


KIFESTŐ (2022-AT-01a)

KISHÓD – KÖNNYŰ



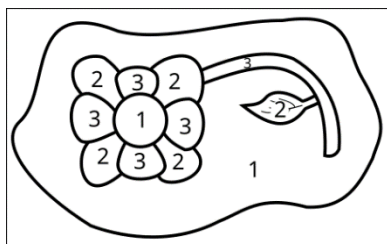
Fesd ki a képet zöld, kék és sárga színekkel úgy, hogy az azonos színűre festett területek ne érintkezzenek egymással!



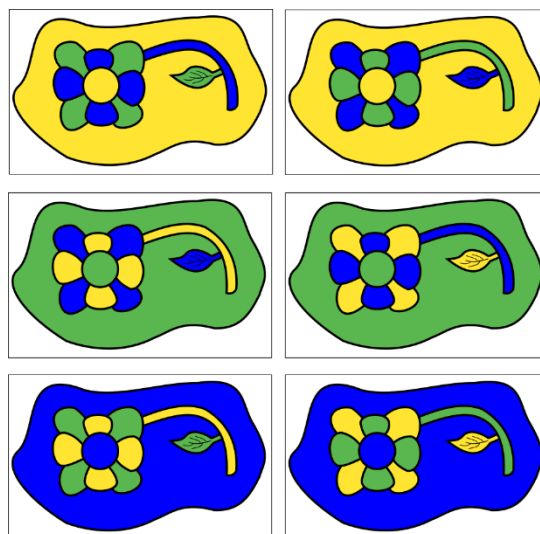
A helyes megoldások:

Hogy találhatsz rá valamelyik megoldásra?

Válassz ki egy színt a legkülső területnek. Például a sárgát (1). A legtöbb terület érintkezik ezzel, tehát a másik két színnel (kék - 2, zöld - 3) kezd el kifesteni felváltva a kisebb területeket. A virág közepe megint festhető az elsőnek választott színre (sárga, 1).



A számokkal jelölt ábrában láthatod a színektől független megoldást.



MIÉRT INFORMATIKA?

Ebben a hódfeladatban színeket kell területekhez hozzárendelni bizonyos megszorítások alapján. Az informatikában és a matematikában az ilyen feladatokat színezési problémáknak (vagy „gráfszínezési” problémáknak) nevezik.

Sokféle felhasználási területe van, és gyakran fontos, hogy minél kevesebb színt használjunk. Ilyen például a csapatok felosztása egy sporttornán, az emberek csoportokra osztása vagy a rádióadók frekvenciáinak kiosztása. A színezési problémák alkalmazásakor általában több terület van, mint ebben a feladatban. Gyakran már nem lehet kézzel és néhány színnel megtalálni a megoldást. Az informatikusoknak számítógépre van szükségük az ilyen feladatok megoldásához.

A színezési probléma térképekhez is használható. Itt az a cél, hogy olyan színezést találjunk, hogy a szomszédos országok ne legyenek egyformák. Bármely térkép találhat ilyen színezést csak négy színnel. Ezt nem könnyű belátni. Kenneth Appel és Wolfgang Haken matematikusok ezt csak 1976-ban tudták bizonyítani. Ehhez számítógépek segítségével ellenőrizték a különféle kivételeket és ellenpéldákat. Ezért vannak matematikusok, akik nem fogadják el ezt a bizonyítást.

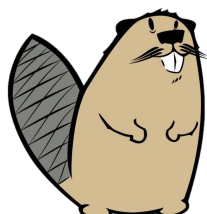
Az informatikusok a csomópontok ilyen részalmazát csomópontlefedésnek (minimális vagy optimális csúcslefedés) nevezik.

Ilyen csomópontlefedési problémákkal a mindennapi életben is találkozhatunk. Például amikor a legjobb helyeket keressük az utcai lámpák vagy a megfigyelőkamerák elhelyezéséhez.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Gr%C3%A1fok_sz%C3%ADnez%C3%A9se

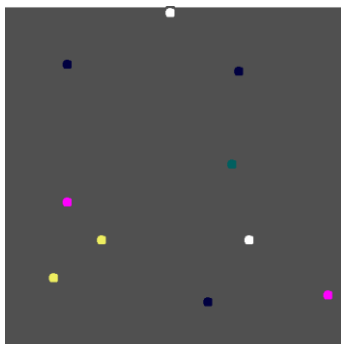
https://hu.wikipedia.org/wiki/N%C3%A9gyysz%C3%ADn-es_t%C3%A9rlefed%C3%A9s



PADLÓMINTA (2022-AT-02)

SENIOR – NEHÉZ

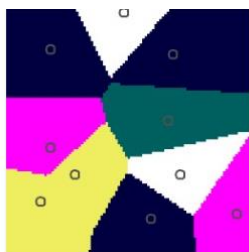
Egy négyzet alakú szoba padlójára 10 különböző színű zsetont teszünk le.



A padlót (kivéve a zsetonokat) ezután teljesen lefestjük úgy, hogy a padló minden pontjának a színét speciális színezési szabály határozza meg. Ugyanaz a szabály érvényes az egész padlóra. Ha a szabály alapján több színt is kiválaszthatunk, bármelyik választható

Párosítsd össze a következő négy szabályt a lefestett padlókkal!

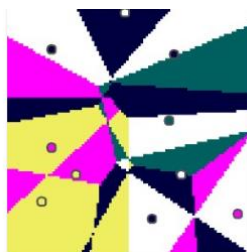
1. szabály: annak a zsetonnak a színével festünk, amely a legközelebb található.
2. szabály: annak a zsetonnak a színével festünk, amely a legtávolabb található.
3. szabály: annak a zsetonnak a színével festünk, amely a második legközelebb található.
4. szabály: annak a zsetonnak a színével festünk, amely a 6 legközelebbi közül a leggyakrabban fordul elő.



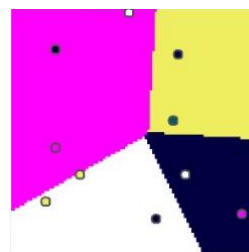
A)



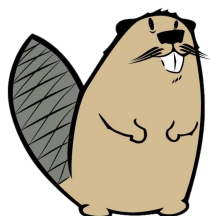
B)



C)



D)

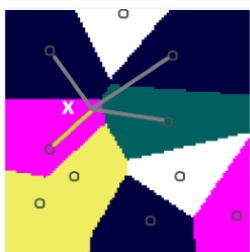


A helyes megoldás:

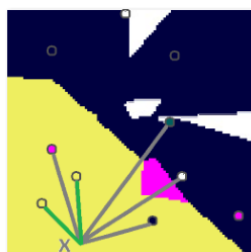
1. szabály - A
2. szabály - D
3. szabály - C
4. szabály - B

Mivel a felület minden pontjára érvényes az adott szabály, elég csupán pár megfelelő pontot ellenőrizni.

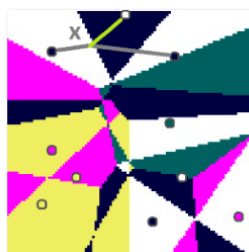
Itt találhatók a példák mind a négy esetre:



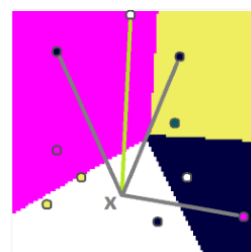
Az x pont rózsaszínre lett színezve, mivel a rózsaszín zseton fekszik az x ponthoz legközelebb.



Az x pont sárga színű lett, mert a 6 legközelebbi zseton között a sárga színű a leggyakoribb.



Az x pont fehérre lett színezve, mivel fehér a második legközelebbi zseton az x-hez.



Az x pont fehér színű lett, mert a fehér zseton esik az x ponttól legtávolabb.

MIÉRT INFORMATIKA?

Egy sík felosztása csakúgy, mint az algoritmikus szerkesztése fontos szerepet játszik az informatika különböző területein, például a szimulációkban és a számítógépes grafikában.

A Voronoi-cella (az "A" eset eredménye) különösen érdekes és gyakran megjelenik a való életben, például a hálózati lefedettségénél. Ezen kívül focimeccsek és szociálökonomiai magatartás (kapcsolat a közösség és a legközelebb elhelyezett iskolák, kórházak és szolgáltatók között) analizálásánál is lehet használni.

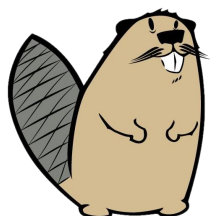
Ezekén felül nagy szerepet kap a cellák alkalmazása a különböző térinformatikai rendszerekben. Tökéletesen használhatók a csapadékeloszlás modellezésére, valamint a magasabb dimenziós változatukkal lehetőség van arra, hogy a hőmérséklet, koncentráció és sűrűség tulajdonságok kezelésével az adattárolást és az adatok kezelését geometriai módszerekkel végezzék.

Voronoi-cellák, ahogy a síkban és a térben értelmezhetők, úgy értelmezhetők például egy gömb vagy a Föld felszínén is.

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Voronoi-cella>

http://www.agt.bme.hu/tutor_h/terinfor/t27.htm



LISTA (2022-AT-04)

KADÉT - NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

A képen számokból álló sorozatot láthatsz, amit X-nek neveztünk el.

A sorozat 1-től 5-ig sorszámozott helyén a következő számok állnak: 3, 5, 2, és 4 és 1.

	1	2	3	4	5
X	3	5	2	4	1

Egy adott helyen álló számot leírhatunk a sorozat nevének és a sorszámnak zárójelezésével.

Például az X sorozat második helye $(X\ 2)$, ahol épp az 5 áll. $(X\ 2) = 5$

Egy így leírt szám maga is lehet egy hely sorszáma egy sorozatban.

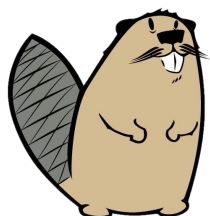
Például $(X\ (X\ 2)) = (X\ 5) = 1$

(Azaz a sorozat ötödik eleme az egy a példánk szerint)

Figyeld meg a következő 3 sorozatot, melyek neve rendre A, B és C.

A	3	2	5	1	5
B	5	4	1	3	2
C	2	5	4	3	1

Melyik számot írtuk le így: $(A\ (B\ (C\ 3)))$?



A helyes válasz az 5

Az $(A (B (C 3)))$ leírás a következőt jelenti: Az "A" sorozatban a $(B (C 3))$ helyen lévő számot írjuk le. Ezt a helyet a "B" sorozatban a $(C 3)$ helyen találjuk meg. Ez pedig a "C" sorozat 3. helyén álló szám, azaz a 4. A "B" sorozat 4. helyén álló szám a 3-as, ezért az "A" sorozat 3. helyén leírt szám a megoldás. Azaz az 5-ös.

Könnyebb a visszafejtés, ha belülről kifele haladunk és beírjuk az egyes számokat:

$$(A (B (C 3))) = (A (B 4)) = (A 3) = 5$$

MIÉRT INFORMATIKA?

Nem is olyan régen az emberek „adattfeldolgozásról” beszéltek, amikor a számítógépek használatáról volt szó. Joggal, mert a számítógépek sokféle adatot dolgoznak fel, mint például számokat, szöveget, képeket, hangokat stb. A számítógépekben tárolt érdekes adatok többsége összetett és strukturált: az időjárási állomáson naponta mért hőmérsékleteket például párok sorozataként is tárolhatjuk, ahol a párok mindegyike a mérés idejéből és a mért hőmérsékletből áll. Tehát itt van egy pár és egy sorozatstruktúra.

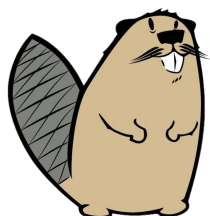
Az adatoknak nagyon sokféle kapcsolata és felépítése lehet, ezért az informatikusok sokféle úgynevezett „adatstruktúrát” fejlesztettek ki annak érdekében, hogy könnyebben tárolják az adatokat, és (ami ugyanilyen fontos) hatékonyan hozzájuk tudjanak férni, fel tudják azokat dolgozni. Egy egyszerű adatstruktúra a tömb (angolul array), amely ebben a hód feladatban is szerepel. Egy tömb meghatározott mennyiségű adatot (pl. számokat) tárolhat egymást követő helyeken (pozíciókon). A tömbben lévő adatokat a helyük alapján dolgozzuk fel, hivatkozunk azokra. Rögzített méretük miatt a tömböket „statikus” adatstruktúráknak tekintik, bár sok programozási nyelv támogatja a dinamikusabb megvalósításukat. Adatok sorozatához léteznek „dinamikus” adatstruktúrák is, például a listák, amelyek mérete szükség szerint változhat.

Akár statikus, akár dinamikus, ha egy sorozat adatstruktúra számokat tartalmaz, ezek a számok pozíciókat is meghatározhatnak ugyanabban vagy más sorozatban. Ezt az informatikában gyakran használják úgynevezett „közvetett címzésre”: A címet vagy a pozíciót a sorozatban nem közvetlenül számként adják meg, hanem közvetetten egy sorozatból származó (numerikus) értékkel, amely önmagában is közvetetten címzett. - stb..

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Adatszerkezet>

[https://hu.wikipedia.org/wiki/T%C3%B6mb_\(adatszerkezet\)](https://hu.wikipedia.org/wiki/T%C3%B6mb_(adatszerkezet))



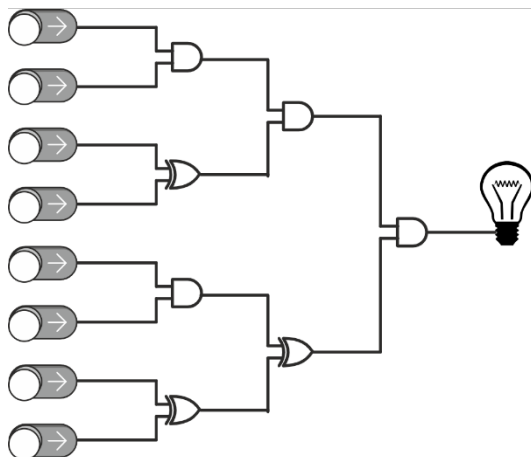
KAPCSOLD FEL AZ IZZÓT! (2022-AU-03)

BENJAMIN – NEHÉZ

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

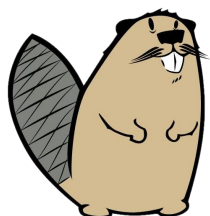
A „Kapcsold fel az izzót” játékban 8 kapcsoló kapcsolható be- illetve ki. Ezekből a kapcsolókból vezetékek vezetnek néhány alkotóelemen keresztül egy villanykörtehez.



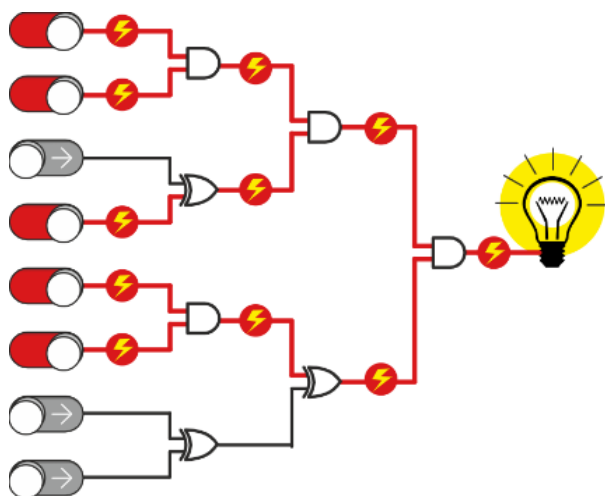
A $\Rightarrow$ alkotóelem kimeneti vezetéke csak akkor van *FELKAPCSOLVA*, ha *MINDKÉT* becsatlakozó vezeték fel van kapcsolva.

A $\mathcal{D}$ alkotóelem kimeneti vezetéke csak akkor van *FELKAPCSOLVA*, ha a becsatlakozó vezetékek közül *PONTOSAN EGY* van felkapcsolva.

Melyik kapcsolóknak kell *FELKAPCSOLVA* lenniük, hogy a végén a villanykörte világítson?



A kapcsolók 16 lehetséges kombinációja vezethet oda, hogy a végén a villanykörte fel legyen kapcsolva. Ezen megoldások egyikét láthatjátok a képen narancssárgával megjelölve.



Egy jó kiindulási pont visszafele hozzálátni ehhez a feladathoz. Ehhez számozzuk be a vezetékeket!

A villanykörte bal oldalt azzal a vezetéssel van összekapcsolva, ami a legjobboldali $\neg D$ összetevőből jön. Ahhoz, hogy ez a vezeték fel legyen kapcsolva, mindkét vezeték (2, 3) fel kell legyen kapcsolva, melyek az $\neg D$ összetevőbe vezetnek.

Ezek a következő két összetevővel vannak összekötve: $\neg D$

Tudjuk, hogy az $\neg D$ összetevőnek felkapcsolva kell lennie, tehát az ebbe vezető két vezeték (4-est, 5-öst) is fel kell kapcsolni.

Tudjuk, hogy az $\neg D$ összetevő FELKAPCSOLT állapotban kell legyen, a 6-os és a 7-es vezetékek egyikét FEL kell KAPCSOLNI, és a másikat pedig LEKAPCSOLNI.

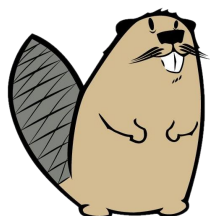
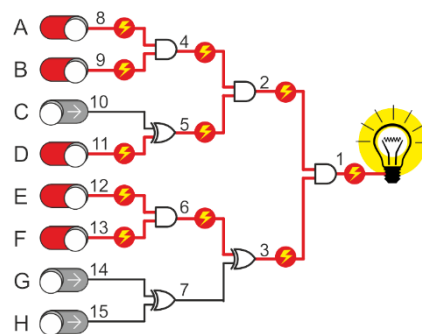
Válasszuk külön a felső és az alsó vezetékeket:

A felső résznél:

A 4-es és az 5-ös vezetékeket fel kell kapcsolni, ezért a 8-as és a 9-es vezetékeket is fel kell. Így a felső két kapcsolót (A-t és B-t) fel kell kapcsolni, és a 10-es és a 11-es vezetékek közül pontosan egyet kell felkapcsolni, tehát a C-t vagy a D-t fel kell kapcsolni.

Az alsó felénél:

Ahhoz, hogy az $\neg D$ összetevő fel legyen kapcsolva, a 6-os és a 7-es vezeték (amik a $\neg D$ összetevőbe vezetnek) közül pontosan egyet kell felkapcsolni, a másik mindenképpen lekapcsolt állapotban van. Ehhez vagy az $\neg D$ összetevőt kell felkapcsolni, ami a másik $\neg D$ összetevőbe vezet, és az $\neg D$ összetevőnek lekapcsolva kell lennie, vagy fordítva. Ahhoz, hogy ezt elérjük, több lehetőség is rendelkezésre áll, a fent mutatott megoldás egy lehetséges megvalósítás. Itt az $\neg D$ összetevő fel van kapcsolva, így az E-nek és az F-nek is felkapcsolt állapotban kell lennie, ami azt



jelenti, hogy az D összetevőt le kell kapcsolni, és a két lenti kapcsoló (G és H) közül vagy mindkettő fel van kapcsolva, vagy mindkettő le.

Az alábbi táblázatban találod az összes lehetőséget:

	A	B	C	D	E	F	G	H
1	FEL	FEL	FEL	LE	FEL	FEL	LE	LE
2	FEL	FEL	LE	FEL	FEL	FEL	LE	LE
3	FEL	FEL	FEL	LE	LE	LE	FEL	LE
4	FEL	FEL	LE	FEL	LE	LE	LE	FEL
5	FEL	FEL	FEL	LE	FEL	FEL	FEL	FEL
6	FEL	FEL	LE	FEL	FEL	FEL	FEL	FEL
7	FEL	FEL	FEL	LE	FEL	LE	FEL	LE
8	FEL	FEL	LE	FEL	LE	FEL	LE	FEL
9	FEL	FEL	FEL	LE	FEL	LE	LE	FEL
10	FEL	FEL	LE	FEL	LE	FEL	FEL	LE
11	FEL	FEL	LE	FEL	LE	LE	FEL	LE
12	FEL	FEL	FEL	LE	LE	LE	LE	FEL
13	FEL	FEL	LE	FEL	FEL	LE	FEL	LE
14	FEL	FEL	LE	FEL	FEL	LE	LE	FEL
15	FEL	FEL	FEL	LE	LE	FEL	LE	FEL
16	FEL	FEL	FEL	LE	LE	FEL	FEL	LE

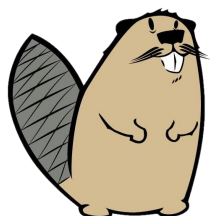
MIÉRT INFORMATIKA?

A feladatban a vezetékeken vagy keresztül folyik áram, vagy nem, tehát a kapcsolók vagy fel vannak kapcsolva, vagy le. Az informatikában az ilyen állapotok a Boole-algebra változójának értékét reprezentálják. Ezek gyakran igaz vagy hamisként, illetve 1 vagy 0-ként ismertek.

A mai számítógépek is ennek a két állapotnak a szabályai alapján működnek. Ez amiatt van, hogy a processzorba több milliárd tranzisztor van beépítve, amelyeknek az állapota szintén csak FELKAPCSOLVA vagy LEKAPCSOLVA lehet.

Utána több tranzisztorból lehet logikai kapcsolást kiépíteni. Két ilyen kapcsolás jelenik meg ebben a feladatban: az „és” kapu, ami csak akkor lesz felkapcsolva, hogyha mind a két bemenet fel van kapcsolva. A másik összetevő egy XOR (kizáró vagy) kapu, ami akkor van felkapcsolva, ha a két becsatlakozó vezeték közül pontosan egy van felkapcsolva.

Utána több tranzisztorból lehet logikai kapcsolást kiépíteni. Két ilyen kapcsolás jelenik meg ebben a feladatban: az „és” kapu D , ami csak akkor lesz felkapcsolva, hogyha mind a két bemenet fel van kapcsolva. A másik összetevő D egy XOR (kizáró vagy) kapu, ami akkor van felkapcsolva, ha a két becsatlakozó vezeték közül pontosan egy van felkapcsolva.



A	B	Az „és” kapu kimenetelei	A XOR kapu kimenetei
KI	BE	Off 	Off
Off	On	Off 	On
On	Off	Off 	On
On	On	On 	Off

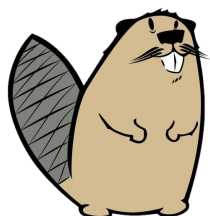
A logikai kapuk megfelelő kombinációja által tudnak a számítógépek nagyon gyorsan bonyolult számításokat végezni. Magasabb szinteken a logikai kapukat a programozásban is használják: ha egy program részének kidolgozása több feltételen alapul, akkor ezek a feltételek ugyanolyan működésű logikai műveletek segítségével kombinálhatóak.

Egy másik terület, ahol Boole-algebrát fedezhetünk fel, a számítógépes programok. Néha egy programnak döntést kell hoznia arról, hogy mi a következő lépés, az előtte történt egy (vagy néha több) dolog alapján. Ezzel találkozhatunk a tesztelés ismétléses utasítások (iteráció, ciklus) és a feltételes elágazás összetett feltételeiben

WEBOLDALAK

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

<https://hu.wikipedia.org/wiki/Boolean>

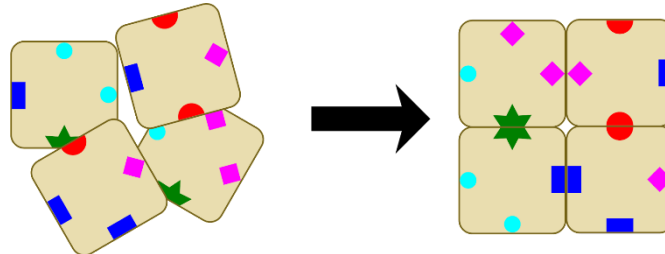


NÉGY KÁRTYA (2022-BE-02)

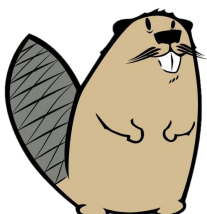
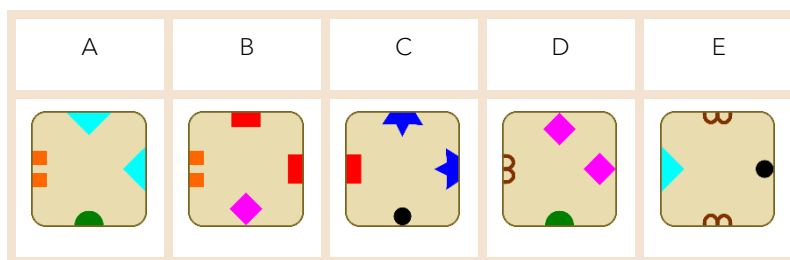
SENIOR – NEHÉZ

Alakíts ki egy négy kártyából álló négyzetet úgy, hogy az érintkező éleken szereplő szimbólumok azonosak legyenek!

A következő négy kártya egy ilyen négyzetet mutat.



A következő öt kártya közül négyből ki tudsz alakítani egy ilyen négyzetet. Melyik kártyát nem tudod felhasználni?

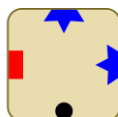


A helyes válasz a C)

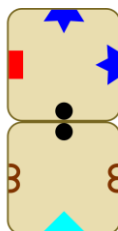
$$\binom{5}{4} \cdot 4! \cdot 4^4 = 30720$$

különböző lehetőség van arra, hogy kiválasszuk az öt kártyából a négy szükségeset. Ha belegondolunk abba, hogy a forgásszimmetria miatt legalább négy megoldás van minden kártyához, rájövünk, hogy ez túl sok, hogy kipróbáljuk mindet.

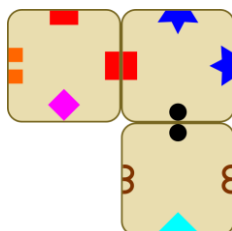
Először a kártyákon lévő szimbólumok felosztását kell megnéznünk. Így feltűnik, hogy csak a C) kártyán van kék félcsillag . Mivel ez kétszer is szerepel rajta – mégpedig egy saroknál, ezért a két szomszédos kártyát csak a bal oldalára és alá tehetjük, ha nem forgatjuk el a kártyát (a forgatás az első kártyánál felesleges lenne).



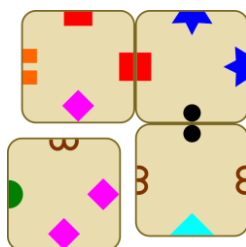
Ezután vesszük az egyetlen kártyát, amin fekete kör van , tehát ezt kell az előző kártya alá tenni, ez az E) jelű kártya.



A piros téglalap  is csak egy kártyán szerepel, ez a B) jelű kártya. Ezen két piros téglalap van, de mivel nincs több kártya piros téglalappal, ezért úgy kell forgatnunk, hogy a másik piros téglalap fent legyen, ne pedig lent.

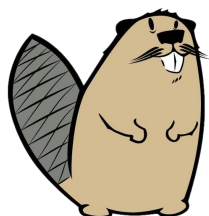


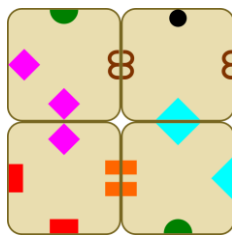
Végül már csak egy olyan kártya kell, amin egy lila négyzet  és a két barna hullám  is van. A D) jelű kártyán megvannak ezek a szimbólumok, viszont rossz sorrendben: a barna hullámnak óramutató járása szerint a lila négyzet után kellene következnie, de ez fel van cserélve.



Mivel egyik kártyát sem tudjuk mással helyettesíteni, ezért jól látszik, hogy a két kék félcsillaggal ellátott C) jelű kártya nem használható a feltételeknek megfelelő négyzet alkotására.

A többi négyzetből viszont össze lehet rakni egy megfelelő négyzetet:





Ebből látszik, hogy a C) jelű kártya az egyetlen, ami nem illeszthető egy megfelelő négyzetbe, tehát a C) a helyes válasz.

Ha követjük a leírt szabályokat, akkor az A) választ kapjuk.

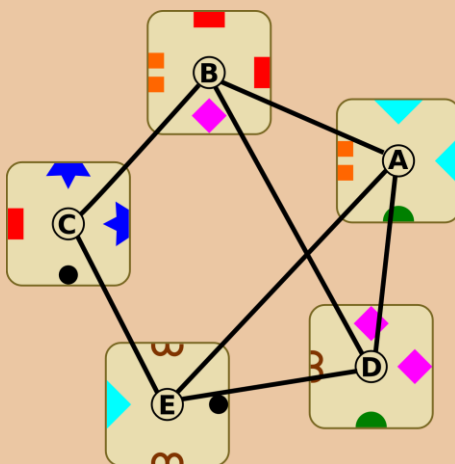
A B válaszban egy kapcsolat helytelenül lett megrajzolva (az 1. végpont és a 2. végpont között mindkét irányban), és egy nem szerepel (2. végpont és 5. végpont mindkét irányban).

Az itt használt módszernél valójában nem lehet „x” az átlóban. Ez a végpont és saját maga között lenne egy kapcsolat (szál), ami bizonyos hálókbán megengedett, de a pókhálókbán nem fordul elő. A C válaszban azonban 2 ilyen kapcsolat is van (az 1. végpontban és a 4. végpontban).

A hálók minden ábrázolásának szimmetrikusnak kell lennie a bal felső saroktól a jobb alsó sarokig (úgy, hogy a szimmetriatengely az átló). A D válasz tartalmazza a kapcsolatot a 2. végponttól az 5. végpontig, de hiányzik az ennek megfelelő másik kapcsolat az 5. végponttól a 2. végpontig.

MIÉRT INFORMATIKA?

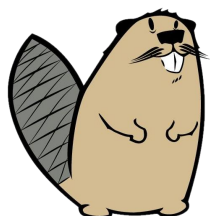
Ha két kártyán feltűnik egy azonos szimbólum, akkor azokat egymás mellé lehet tenni. Ezt egy gráf segítségével ábrázolni tudjuk: a kártyák lesznek a csúcsok, a közös szimbólumokat jelölik az élek. Az öt kártya gráfja így néz ki:



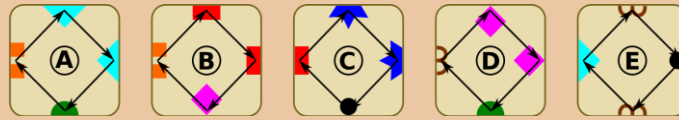
Ha négy kártyából alakítunk ki egy négyzetet úgy, hogy a szomszédos kártyáknak legyen azonos szimbóluma, az azt jelenti, hogy a gráfon egy kört kapunk a négy kártya bejárásával, például A-ból E-be C-be B-be és vissza az A-ba. Megjelöltük ezt az utat, mint A-E-C-B-A, és ezzel minden olyan körre gondoltunk, ami ugyanezt a sorrendet tartja, tehát E-C-B-A-E, C-B-A-E-C és B-A-E-C-B.

A gráfban pontosan három darab négy kártyából álló kör van: a már említett A-E-C-B-A, A-E-D-B-

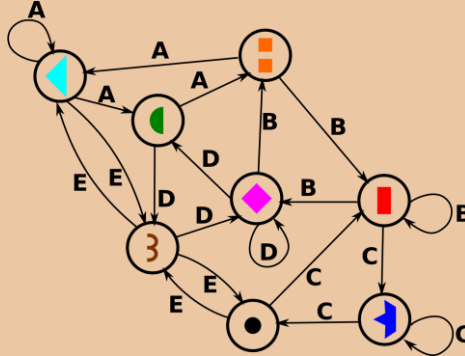
A, és B-D-E-C-B. Ez csökkenti a megoldások számát $\binom{5}{4} \cdot 4! \cdot 4^4 = 30720$ -ról (illetve 7680-ról, a forgásszimmetriától eltekintve) 12-re (illetve 3-ra, a forgásszimmetriától eltekintve). Ezek már gyorsan ellenőrizhetők, és megtalálható a jó megoldás (A-E-D-B-A).



Olyan gráfot is lehet használni, ahol szimbólumok jelölik a csomópontokat. Ehhez egységesen, például az óramutató járásával megegyezően kell meghatározni a szimbólumok sorrendjét egy kártyán belül:



A gráfban tehát van egy *irányított él* két szimbólum között, ha azok az óramutató járásával megegyező irányban szomszédosak:

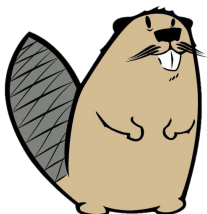


Csak akkor lehet megoldani a feladatot, ha találunk egy olyan utat, ami pontosan négy kártyán megy át. Ez az út a - - - - , ami az A, E, D és B kártyákon megy át, ahogy a megoldásunk is mutatja.

WEBOLDALAK

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

[https://hu.wikipedia.org/wiki/Absztrakci%C3%B3s_elv_\(sz%C3%A1m%C3%ADt%C3%B3g%C3%A9p-programoz%C3%A1s\)](https://hu.wikipedia.org/wiki/Absztrakci%C3%B3s_elv_(sz%C3%A1m%C3%ADt%C3%B3g%C3%A9p-programoz%C3%A1s))



VARÁZSFÖLDE (2022-CA-02)

SENIOR - NEHÉZ

Varázsföldén négyféle mágikus tárgy található: varázskalap , kristálygömb , varázskönyv  és bájital .

A varázskalapok és a kristálygömbök kétféleképpen alakíthatók át.

A táblázat azt mutatja, hogy mi jön létre az egyes tárgyakból. Minden átalakításkor a táblázatban meghatározott sorrendben, az eredeti tárgy helyére kerülnek az új mágikus tárgyak:

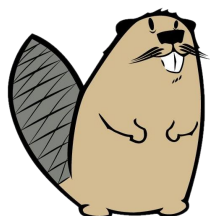
Az átalakítások tetszőleges számú alkalommal és bármilyen sorrendben megtörténhetnek.

Ily módon egyetlen mágikus tárgyból mágikus tárgyak hosszú sora keletkezhet.

	Előtte	Utána
1		
2		 
3		 
4		  

Egy varázskalapból  kiindulva melyik sor NEM jöhet létre?

A)	    
B)	       
C)	     
D)	   

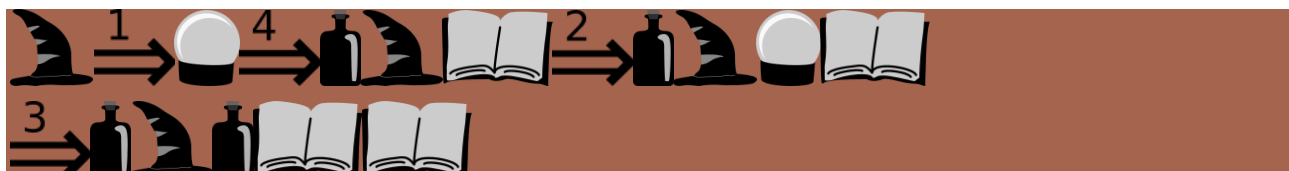


A helyes válasz a B)

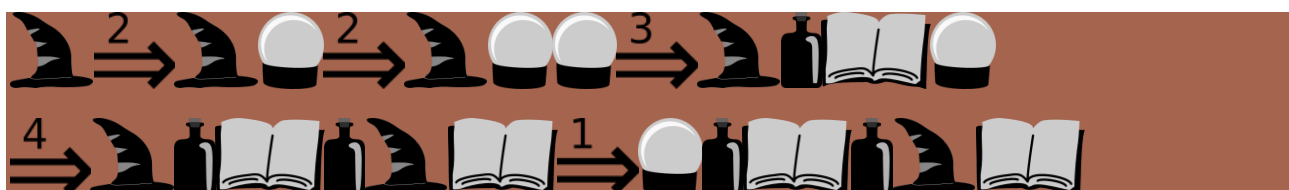
Számozzuk be a lehetséges átalakításokat a táblázatban fentről lefele haladva.

Egy varázskalapból kiindulva az A, C és D válaszokban található sorok létrehozhatók az alábbi átalakításokat követve:

A:



C:



D:



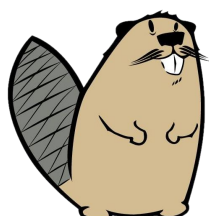
A B válasz különböző számú varázskönyvet és varázssitalt tartalmaz. De mindig, amikor létrejön egy varázskönyv, akkor egy varázssital is létrejön. Azaz azonos számmal jelennek meg az átalakításokkor. Ezeket a varázstárgyakat azonban nem lehet továbbalakítani - így nem „tűnhetnek” el a sorból. Tehát mindig azonos számban kell megjelenniük a sorban. Emiatt varázsgömbből kiindulva sem állítható elő a B válasz.

MIÉRT INFORMATIKA?

Ebben a hódfeladatban a kristálygömbök és varázskalapok átalakítása során mindig új kristálygömbök és varázskalapok születhetnek. Emiatt végtelen számú elrendezés készíthető. Tehát nem lehet minden olyan elrendezést leírni, ami az átalakítások segítségével felmerülhet. De erre nincs is szükség, mert az elrendezések halmazát maguk az átalakítások pontosan meghatározzák.

Már a számítógépek létezése előtt felmerült az elrendezések végtelen halmazainak leírása egy viszonylag kicsi, de mindenképpen véges transzformációhalmaz segítségével. Az informatikában a transzformációkat (átalakításokat) „csereszabályoknak”, a szabályhalmazokat „nyelvtanoknak”, az azokat meghatározó halmazokat pedig „nyelvnek” nevezik. Ez az eldöntési probléma központi szerepet játszik ezekben a „formális nyelvekben”: egy elrendezés előállítható-e a szabályok alkalmazásával?

Ennek a hódfeladatnak a megválaszolásához négy elrendezésre kellett ezt a feladatot megoldani. Szerencsére az itt megfogalmazott nyelvten a „kontextusmentes nyelvtenok” osztályába tartozik: a kristálygömbök és a varázskalapok anélkül is átalakulhatnak, hogy figyelembe vennék a körülöttük lévőket, azaz vizsgálunk kellene a környezetüket. A döntési probléma általában könnyen

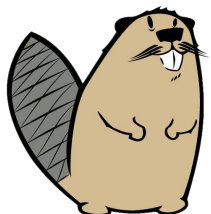


megoldható a környezetfüggetlen nyelvtanoknál, ezért nagyon népszerűek a számítástechnikában, például programozási nyelvek leírására.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Form%C3%A1lis_nyelv

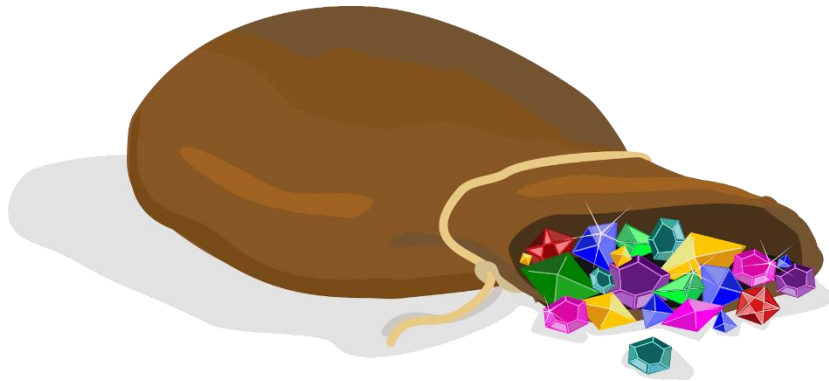
https://hmn.wiki/hu/Context-free_grammar



TÖMÖR MEGJELENÍTÉS (2022-CA-04)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES



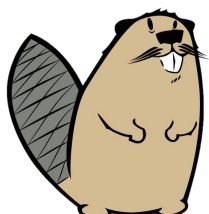
Péternek különböző értékű drágakövei vannak.

Sára tudja, hogy Péternek vannak drágakövei, de nem tudja, hogy melyik mennyit ér. Szeretné tudni, hogy melyik drágakő a legértékesebb.

Ehhez háromszor megismétli a következőket:

- Kiválaszt négyet Péter drágakövei közül és megkérdezi, hogy melyik ezek közül a legértékesebb.

Legfeljebb hány drágakőve lehet Péternek, ha Sára a három kérdés végeztével megtalálja a legértékesebbet?



A helyes válasz a 10

Ha Péternek 10 drágaköve van, Sára összesen nyolc különböző drágakövet választhat az első két kérdéshez. Az egyes kérdések két „győztese” (azaz a négy kiválasztott drágakő közül a legértékesebbek) lehet az „összesített győztes”, tehát a legértékesebb drágakő. A többi hat drágakő kiesett. Az utolsó kérdésnél az eddigi győztesek és a még nem választott két drágakő következik. Ennek a kérdésnek a nyertese lesz az összesített győztes.

Tehát 10 drágakő esetén akár így is meglehet Sára, hogy megtalálja a legértékesebbet. Ha Péternek 11 drágaköve lenne, sajnos már nem sikerülne:

Ha Sára, mint fent, az első két kérdésnél összesen nyolc különböző drágakövet választ ki, marad a két nyertes és az eddig nem választott három drágakő, ami túl sok ahhoz, hogy a harmadik kérdésben megállapítsa az összesített győztest. Ha Sára az első kérdés győztesét a második kérdésben 3 „új” drágakőhöz hasonlítja, akkor tudja, hogy a kiválasztott hét drágakő közül melyik a legértékesebb. Ezt a drágakövet még össze kell hasonlítani a maradék négy drágakővel. Ez is túl sok drágakő lenne a harmadik kérdéshez.

Ha 10 drágakő esetén Sára az első két kérdésben csak hat vagy még kevesebb különböző drágakövet választana, vagy Péternek több, mint 10 drágaköve lenne, három kérdés után Sára nem tudná biztosan, hogy melyik a legértékesebb drágakő.

MIÉRT INFORMATIKA?

Ahhoz, hogy megválaszoljuk a kérdést, meg kell találnunk a legjobb stratégiát, ami mindig működni fog Sára számára. A stratégia egy algoritmus, ami olyan lépéssorozat, ami megoldja a problémát. Ebben a feladatban kiértékeljük az algoritmust, hogy megtaláljuk a maximális számú drágakövet, amire működik. A drágakövek maximális számának meghatározására szolgáló algoritmusnak van néhány megkövetelése:

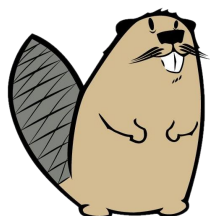
1. meg kell találni a négy közül a legdrágább drágakövet
2. csak három négyes csoportot tudunk megvizsgálni

Sok olyan algoritmus van, ami bizonyos helyzetekben jól működik, másokban viszont nem, például egy rendezési algoritmus képes lehet pozitív egész számokat rendezni, de negatív egészeket nem. Egyes útvonalkereső algoritmusok meg tudják találni a legrövidebb útvonalat két város között, de túlságosan lelassulnak olyan térképre alkalmazva, melyben több ezer összekapcsolt város van. Ilyenkor gyakran inkább gyorsabb algoritmusokat választanak, amik megtalálják az egyik legrövidebb útvonalat, de nem feltétlenül a legrövidebbet. Egy jó megoldás gyorsan megtalálva jobb, mint a semmilyen megoldás!

A számítógépeken való futtatásra programozott algoritmusok kiértékelésének számos módja van. Például figyelembe vehetjük egy algoritmus futási idejét (várhatóan mennyi ideig tart az elejétől a végéig), vagy az algoritmus által felhasznált memóriaterület mennyiségét, ami azt jelzi, hogy mennyi számítógépmemóriát használhat fel végrehajtása során.

másik leírás a Masiar and Wilfred Workinggroup 2-től:

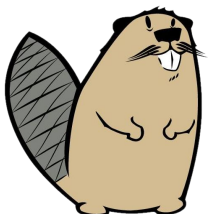
Ez a feladat egy olyan algoritmusról szól, amit feltételek korlátoznak. Esetünkben Sára csak három kérdést tehet fel, és mindegyik kérdés csak 4 elemet tartalmazhat. A korlátozások ellenére jól működik ez az algoritmus 11-nél kisebb méretű gyűjteményekre, de egyébként nem működik. Az algoritmus korlátozásának számos oka lehet. Például előfordulhat, hogy egy művelet meghatározott időn belül végre kell hajtani, ami a valós idejű operációsrendszerekben szükséges. Egy további ok lehet, hogy a műveletek külső költségeket okozhatnak vagy károsíthatnak egy



alkatrészt. Nem probléma, ha az algoritmus egy bizonyos küszöb felett meghibásodik, mindaddig, amíg biztosítva van, hogy sosem éri el ezt a küszöböt. Például ennek a feladatnak a korlátozott stratégiája, hogy a gyűjtemény mérete sose legyen 10-nél több.

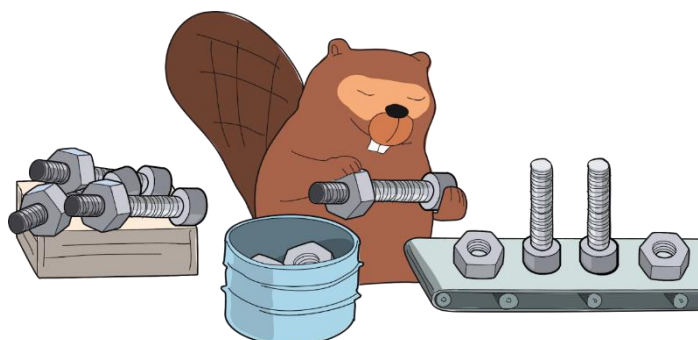
WEBOLDAL

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



JUNIOR – KÖNNYŰ

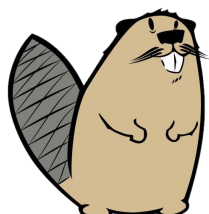
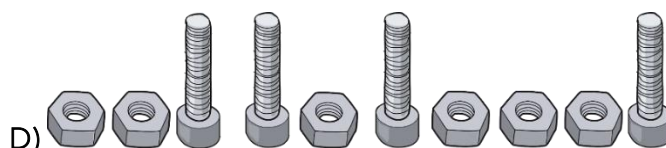
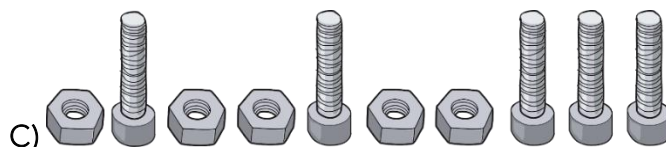
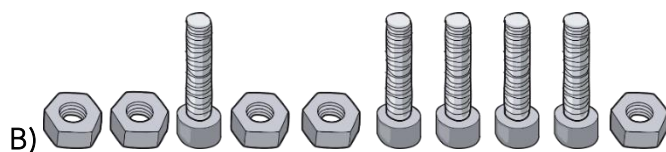
csavarokat



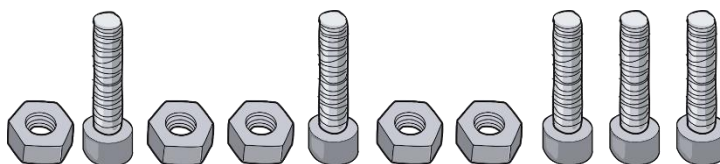
- Leveszi a következő alkatrészt a futószalagról.
 - Ha az egy anya, akkor beleteszi a vödörbe.
 - Ha az egy csavar, kivesz egy anyát a vödörből, rácsavarja a csavarra és a kész alkatrészt a dobozba teszi.

- Ben elvesz egy csavart a futószalagról, de nincs anya a vödörben, amit rá lehetne csavarni.
- Ben az összes alkatrészt feldolgozta az futószalagról, de még mindig vannak anyák a vödörben.

A) 



A helyes válasz a C)

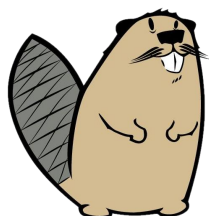


A táblázat a kész alkatrészek dobozának, az anyák vödrének és a futószalagnak az állapotát mutatja.

A megoldás lépésről lépésre:

A táblázat a kész alkatrészek dobozának, az anyák vödrének és a futószalagnak az állapotát mutatja.

Doboz	Vödör	Futószalag
üres	üres	
üres		
	üres	





Miért nem jó a többi válasz?

A) hibához vezet a megjelölt pontban.
Itt Ben levett egy csavart, de nem maradt anyá a vödörben.

B) hibához vezet a megjelölt pontban.
Ben eddig 4 anyát csavart fel 4 csavarra. Tehát a vödör üres. De most levett egy ötödik csavart, amihez már nincs anyá.

D) hibát eredményez a teljes sorozat feldolgozása után.
Ben 4 csavarra 4 anyát már felcsavart, de marad még 2 anyá a vödörben.

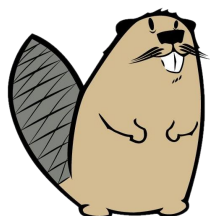
MIÉRT INFORMATIKA?

Ben olyan alkatrészeket dolgoz fel, amelyek egymás után jönnek a futószalagon. Egy nagy vödört használ az anyák ideiglenes tárolására. Hasonló elrendezést használnak az elméleti informatikában olyan algoritmusok modelljeként, amik problémák egy bizonyos osztályát képesek megoldani: a veremautomatákat. A veremautomata adatokat (számokat vagy karaktereket) dolgoz fel, amelyeket egymás után kap bemenetként. Egyetlen végtelenül nagy tárolóból, egy veremből áll. Ellentétben a feladatban szereplő vödörrel, a veremben lévő elemek meghatározott sorrendűek, és csak az utolsóként behelyezett elemet távolíthatjuk el az veremből („last in first out”, LIFO).

Egy veremautomata használható a környezetfüggetlen nyelvek felismerésére. Az informatikában a nyelv bizonyos szabályok szerint kialakított karakterláncok összessége. A nyelvek egy egyszerű típusa a környezetfüggetlen nyelv. A környezetfüggetlen nyelv jó példája a jól formált zárójel-kifejezések sorozata. Egy jól formált zárójel-kifejezésben minden nyitott zárójel bezárul egyszer. A jól formált példák ((())) és ((())). A (((() és ((() nem jól formált kifejezések, mert marad lezáratlan zárójelpár. A feladatban szereplő anyákat és csavarokat tekinthetjük nyitó és záró zárójelnek. Tehát Ben csak akkor dolgozza fel futószalagon érkező alkatrészsorot hiba nélkül, ha az egy jól formált zárójel-kifejezést jelent. A zárójel-kifejezések ellenőrzése a fordító egy fontos feladata, ami a programszövegeket futtatható programokká fordítja le. Ezért a legtöbb programozási nyelv programszövegei beágyazott függvényhívásokat és zárójeles aritmetikai kifejezéseket tartalmaznak.

WEBOLDAL

<https://gobertpartners.com/what-is-pushdown-automata-with-examples>



GREEDY TROLLOK (2022-CH-04)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Gabi a szülőfalujából a kastélyba akar menni, de a falvak közötti erdőben mohó trollok várakoznak az ösvényeken. Mindegyikük egy adott érmét (1, 2, 3) kér mindenkitől, aki arra áthalad.

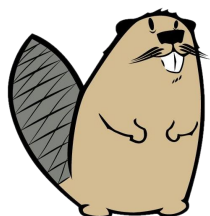
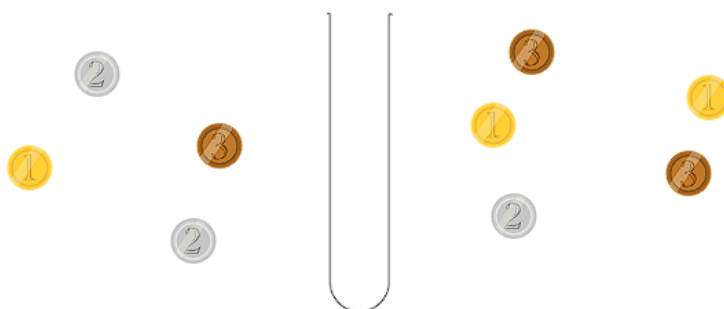
Néhányan egy speciális gyógynövényt (, , ,) is szeretnének.

Szerencsére ezek a gyógynövények megtalálhatók az ösvényeken. A képen láthatod, hogy az egyes trollok melyik érmét akarják és kérnek-e hozzá gyógynövényt is, illetve melyik ösvényen, melyik gyógynövény gyűjthető. Természetesen nem lehet gyógynövényt gyűjteni anélkül, hogy ne fizetnének ki a trollt ezen az ösvényen.



Gabi indulás előtt egy tartóba helyezi a szükséges érméket, amelyből mindig a legfelsőt tudja kivenni. A gyógynövényeket pedig mindig egymásra helyezi a zsákjában és mindig csak a legfelsőhöz fér hozzá. A kastélyba már nem akar gyógynövényeket bevinni.

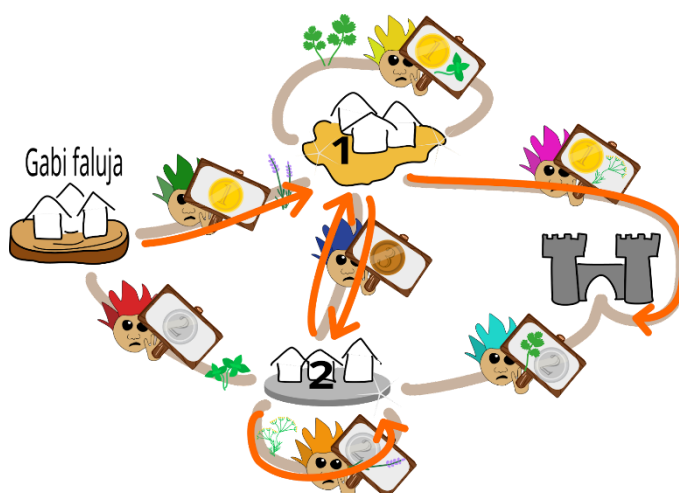
Tedd a tartóba az öt érmét a megfelelő sorrendben, hogy Gabi ki tudja fizetni a kapzsi trollokat a kastély felé vezető úton. Ne feledkezz meg a gyógynövényekről sem!



Két megoldás lehetséges:

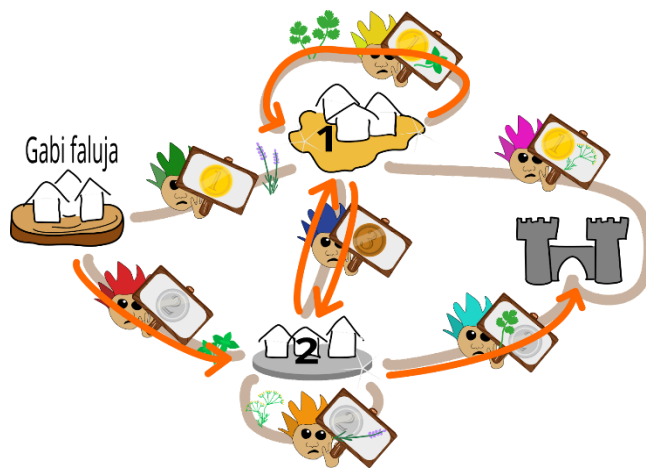
1. megoldás

 Gabi először azon az úton megy, ahol
 fizet egy aranyat  és szedhet
 levendulát . Aztán elmegy egy réz
 fizetésével a 2-sel jelölt faluba,
 ahol az erdei ösvényen kirándulva
 egy ezüstöt  és egy levendulát
 kell fizetnie, de kaprot 
 szedhet. Ezután visszatér az 1-sel jelölt faluba
 (újabb rézpénzt  fizetve). Onnan már
 mehet a kastélyba, átadva a trollnak egy
 aranyat  és a kaprot .



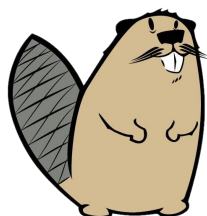
2. megoldás

 A másik megoldás, hogy Gabi a 2-sel jelölt
 faluba megy először. Utközben fizet egy
 ezüstöt  és bazsalikomot  szed.
 Aztán elmegy az 1-sel jelölt faluba és
 útközben fizet egy rézet . Aztán
 kirándul az erdőben, ahol egy aranyat 
 és a bazsalikomot kell kifizetnie , de
 ahol koriandert  gyűjthet. Ezután visszatér a 2-
 sel jelölt faluba (újabb rézpénzt 
 fizetve). Onnan mehet a kastélyba, mert van egy ezüstje  és koriandere .



Csak ezek a megoldások lehetségesek, mert vagy kaporra , vagy korianderre  van szüksége
 a kastélyba kerüléshez. Ahhoz, hogy megkapja a kaprot, először levendulát  kell gyűjtenie. A
 koriander megszerzéséhez pedig először bazsalikomot  kell gyűjtenie.

Ehhez pedig csak 5 érmét használhat fel. Minden más út több, mint öt érmébe kerülne neki



MIÉRT INFORMATIKA?

Ebben a hód feladatban két adatszerkezetet is használtunk, melyek sokszor előfordulnak az informatikában.

A falvak, a kastély és a közöttük lévő ösvények egy gráfként is megadhatóak. Minden útnak van egy bizonyos költsége (az érme), és begyűjthető valami, ami egy veremben tárolódik.

A verem az „utolsó az elsőként” elvet követő adatszerkezet, amely a következő műveleteket támogatja: behelyezünk egy objektumot a verem tetejére (push), vagy kivesszük a verem tetején lévő objektumot (pop).

Ezt a gráfot veremautomatának (pushdown) tekinthetjük. A veremautomata szerkezeti felépítése abban különbözik a véges automatáktól, hogy kiegészül egy memóriával, amit veremnek neveznek. Azaz különböző állapotokból (a falvak és a vár), egy bemeneti ábécéből (érmék), egy verem memóriából (gyógynövények), állapot átmenetektől (ösvények), egy kezdőállapotból (itt Gabi faluja) és végállapotok halmazából (itt a kastély) áll. A feladatban szereplő automata különlegessége, hogy minden út mindkét irányba „járható”, ami nem minden veremautomatára jellemző.

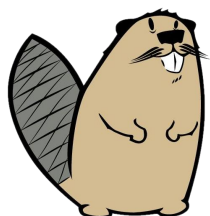
Ez a környezetfüggetlen nyelvtanokhoz kapcsolódó megoldás, mely kellően hatékony a legtöbb programozási nyelv szintaxisának leírásához; valójában a legtöbb programozási nyelv szintaxisának meghatározására környezetfüggetlen nyelvtanokat használnak.

WEBOLDALAK

Környezetfüggetlen nyelvtan - Wikipédia (wikipedia.org)

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

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



ÉTELREJTEKHELY (2022-CH-08)

BENJAMIN - KÖZEPES

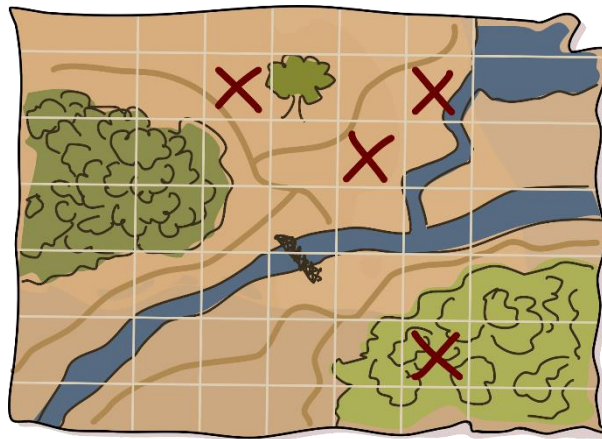
KADÉT - KÖNNYŰ

Castorusnak két jó rejtekhelye van, ahol élelmet rejthet el. Hogy emlékezzen rájuk, egy „X”-szel jelöli ezeket a térképen. Ha riválisa, Biberina megtalálja a térképet, tudni fogja, hol vannak elrejtve az élelmek.

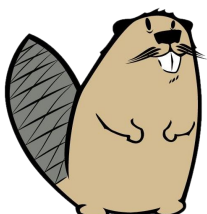
Hogy Biberinát megzavarja, Castorus több „X”-et rak a térkép más mezőire is. Azonban gondoskodik róla, hogy az „X”-ek száma minden sorban és oszlopban 0 vagy páros legyen.

Ezután törli a rejtekhelyeit mutató két „X”-et.

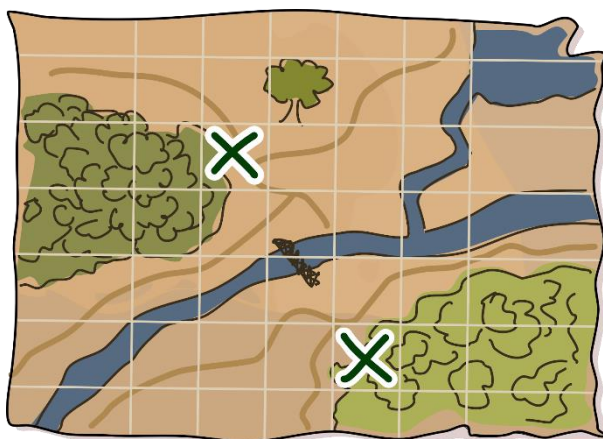
Itt láthatod az eredményt:



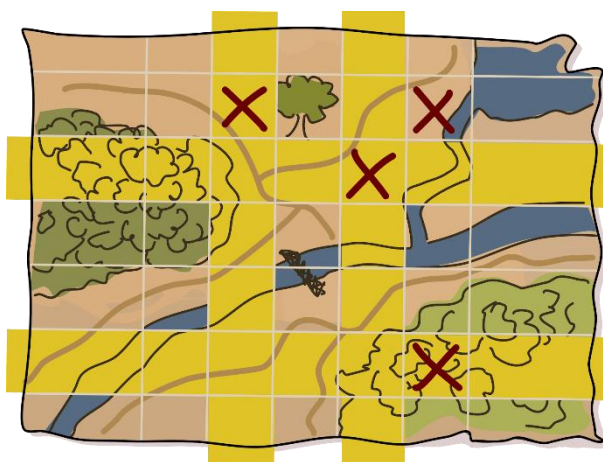
Hol található Castorus két rejtekhelye?



Itt vannak Castorus rejtekhelyei:)



A rejtekhelyek megtalálásához nézzük meg az eredeti térképet, és megállapíthatjuk, hogy két olyan sor és oszlop van, ahol az „X”-ek száma nem páros: 3. és 6. sor és 3. és 5. oszlop.



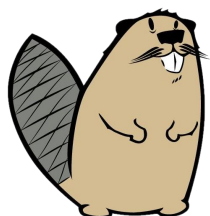
A rejtekhelyeket mutató „X”-ek innen lettek törölve. Tudjuk, hogy minden sorban és oszlopban páros számú „X”-nek kell lennie a törölt „X”-ek újrarajzolása után.

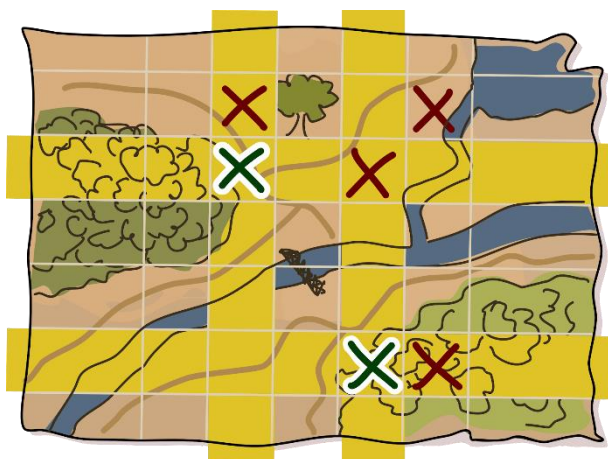
Az érintett sorok és oszlopok négy pontban metszik egymást. Ezek a metszéspontok különösen érdekesek számunkra. Ha nem metszéspontba tennénk egy „X”-et, az oszlopban páros „X”-számot érhetnénk el, viszont ezzel egyidőben abban a sorban páratlant és fordítva. Ezért a két rejtekhelyet jelző „X”-nek a metszéspontokban kell lenniük.

Összesen négy metszéspont van. A jobb felső metszéspontban található egy „X”: ez már nem lehet rejtekhely, mert tudjuk, hogy Castorus a rejtekhelyeket jelző „X”-eket törölte.

A páros számú „X” visszaállításához a 2. sor baloldali metszéspontjába tennünk kell egy „X”-et. Itt lesz egy rejtekhely. A másik rejtekhely nem lehet a bal alsó metszéspontban, mert így ebben az oszlopban három „X” lenne. A másik rejtekhely tehát a jobb alsó metszéspontban lesz.

Itt látható a térkép, mielőtt Castorus törölte volna az „X”-eket, páros számú „X”-el minden sorban és oszlopban:





MIÉRT INFORMATIKA?

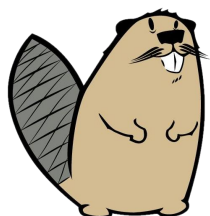
A Castorus itt egy olyan trükköt alkalmaz, amit általában az informatikában használnak: a paritásbitek. Ezek a hibaészlelési és hibajavító kódok néven ismert technikák sorozatának része. Az ötlet az, hogy amikor az adatokat bitek sorozataként (lehet 0 vagy 1) tároljuk vagy továbbítjuk, hozzáadunk extra biteket. Ezek segítenek felismerni, ha átviteli- vagy memóriahiba történt - jellemzően akkor, amikor egy bit átfordult, pl. ha egy bitet 1-ként küldtek, és hibásan 0-ként fogadták vagy fordítva.

Ha például egy egyszerű hibaérzékelő kódot használnánk, egy paritásbit lenne hozzátéve, hogy az egyesek száma mindig páros legyen. 0110101-hez hozzáadnánk egy 0-t, hogy 01101010 legyen (az 1-es bitek száma páros marad). Ha a második bit megfordult és az üzenet most 00101010-ként lett elküldve, akkor ez a fogadott üzenet nem felel meg a páros paritásnak (három darab 1-es bit). Vigyázzunk, hogy ez a módszer nem képes felismerni a problémát, ha egy bitnél több hiba. Valamint a hiba helyét sem határozhatjuk meg vele.

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Parit%C3%A1sbit>

<https://hu.wikipedia.org/wiki/Hibajav%C3%ADt%C3%A1s>

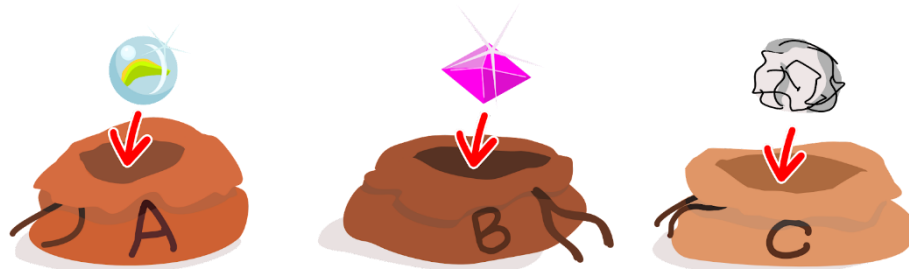


KINCSEK A ZSÁKBAN (2022-CH-14)

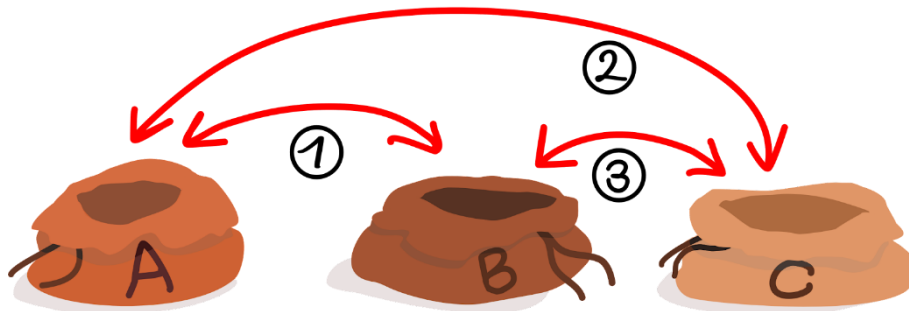
KISHÓD – NEHÉZ

BENJAMIN – KÖZEPES

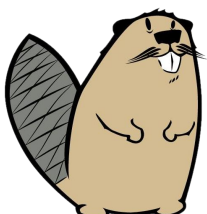
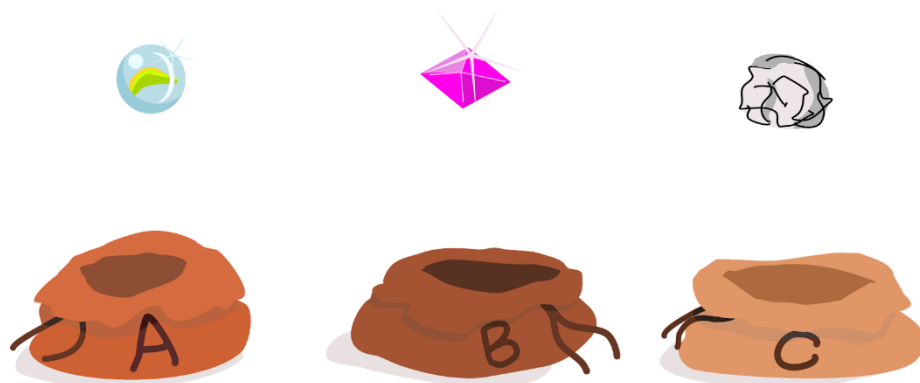
Lilla egy üveggolyót tesz az „A” zsákba, egy drágakövet a „B” zsákba és egy darab papírt a „C” zsákba.



Ezután kicseréli az „A” zsák és a „B” zsák tartalmát, majd az „A” és a „C” zsákét, végül pedig a „B” és a „C” tartalmát.



Ezek után hol található a 3 tárgy?



A megoldás menete:

Az elején a következőképp helyezkedik el a 3 tárgy:



Lilla háromszor cseréli ki a tárgyakat. Az első csere (A-B) után így néznek ki a zsákok:



A második csere (A-C) után:



A harmadik, egyben utolsó csere (B-C) után:



Ezért a végén a papír az „A”, a drágakő a „B”, az üveggolyó a „C” zsákban lesz. Ezt az eredményt egyszerűbben is el lehet érni, mégpedig egy, az „A” és „C” zsák tartalmát érintő cserével.

MIÉRT INFORMATIKA?

Ez a feladat a sorrendről szól. A tárgyak sorrendjét rendezésnek is nevezhetjük. Egy másik sorrend más rendezést jelent. Két tárgy megcserélése esetén megváltozik a tárgyak sorrendje, és ez egy másik rendezéshez vezet. Ebben a feladatban az üveggolyó-drágakő-papír sorrendből a 3 csere után papír-drágakő-üveggolyó sorrend alakult ki.

Elgondolkodtató kérdés lehet, hogy milyen különböző rendezéseket lehet a 3 tárgy segítségével kialakítani.

Érdekes az a tény, hogy a cserékkel bármelyik kívánt rendezést el tudjuk érni. Ehhez n tárgy esetén maximum $n-1$ csere szükséges.

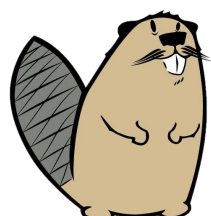
Az informatikában sokféle rendezést ismerünk, ezeknek a hatékonysága függ attól, hogy mi a kezdő sorozat, milyen sorrendbe akarjuk rendezni az elemeket, esetleg ezeknek van-e valamilyen speciális tulajdonsága

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Permut%C3%A1ci%C3%B3>

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

<https://www.tantaki.hu/matek/kombinatorika>



MARI SZOMSZÉDAI (2022-CY-01)

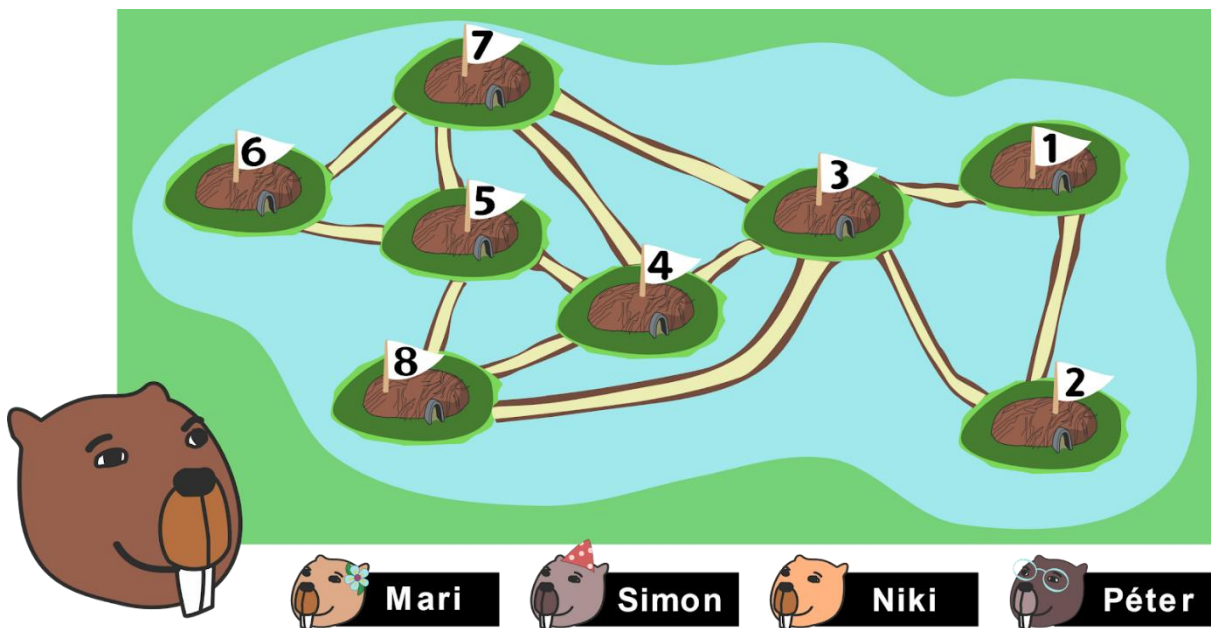
BENJAMIN - NEHÉZ

KADÉT – KÖZEPES

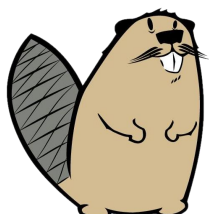
JUNIOR – KÖNNYŰ

A kártyákon nyolc hód várát láthatod. Két hód szomszédja egymásnak, ha egy csatorna összeköti a várukat.

- Marinak, Simonnak és Péternek pontosan 4 szomszédja van
- Nikinek csak Simon és Péter a szomszédja.

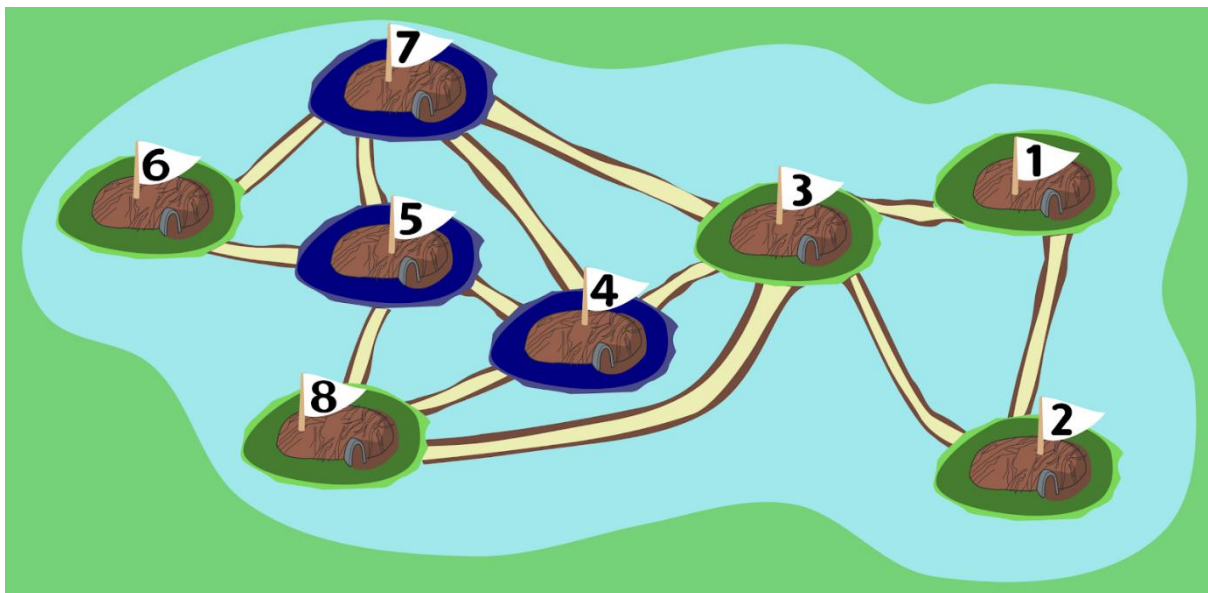


Melyik várban lakik Mari?



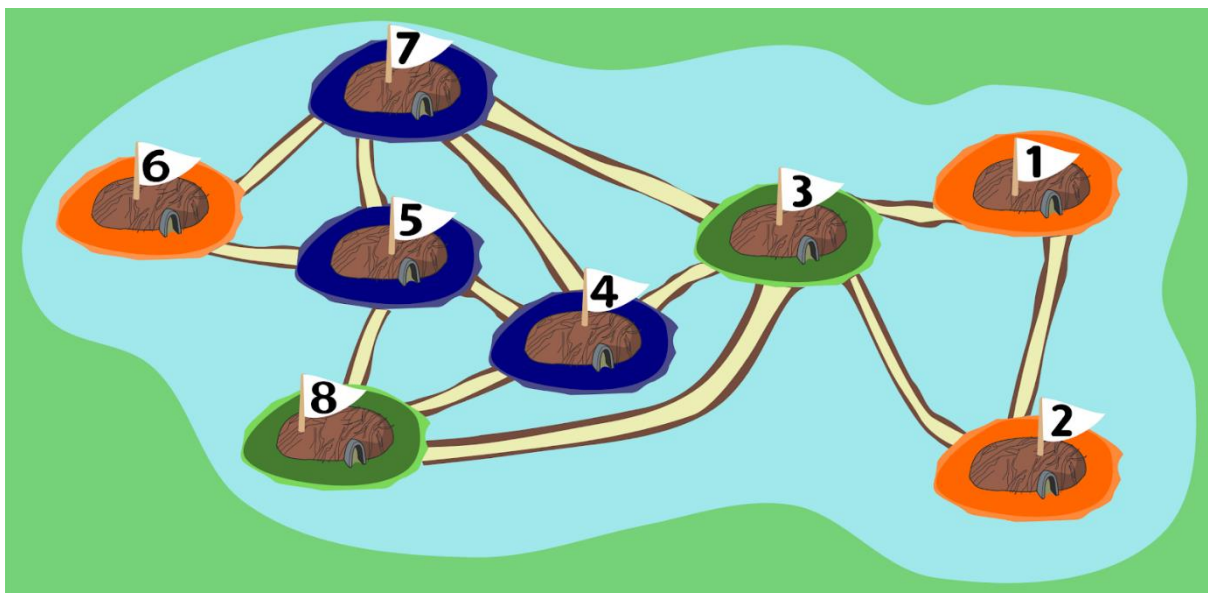
A helyes válasz a 4-es hódvár

A probléma megoldásához a várak közötti csatornák számából kell kiindulnunk. keressük meg azokat a várakat, melyekből négy kimenő csatorna indul. Ezekből összesen három van: 4-es, 5-ös és a 7-es.



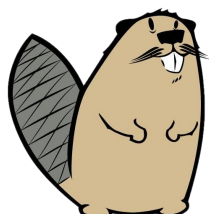
Következésképp a három vár Marihoz, Péterhez és Simonhoz tartozik, de nem tudjuk ki melyikben lakik, így ki kell találnunk melyikben él Mari.

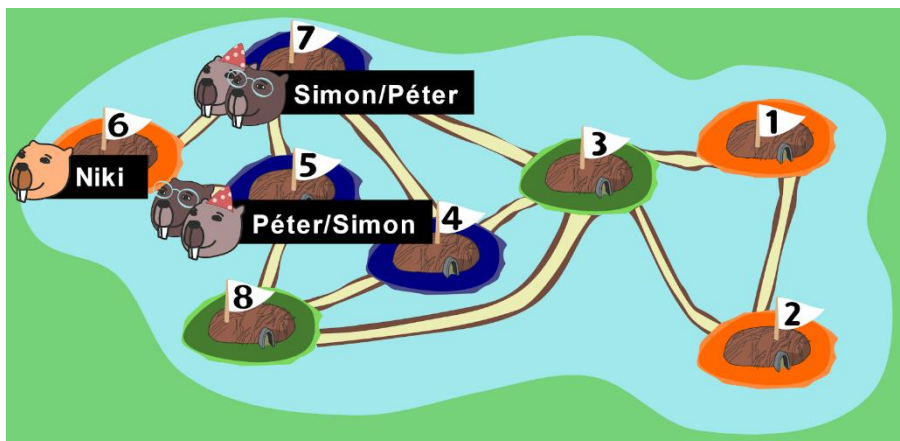
A második információ Niki várára vonatkozik. Ebből következtetni tudunk arra, hogy az ő várából pontosan két csatorna indul. Tehát Niki a következő három vár egyikében él: 1-es, 2-es, vagy 6-os.



Mivel tudjuk, hogy Niki két szomszédja Simon és Péter, így arra következtethetünk, hogy

- Niki a 6-os számú várban él
- Simon és Péter az 5-ös és a 7-es várban élnek (vagy fordítva).



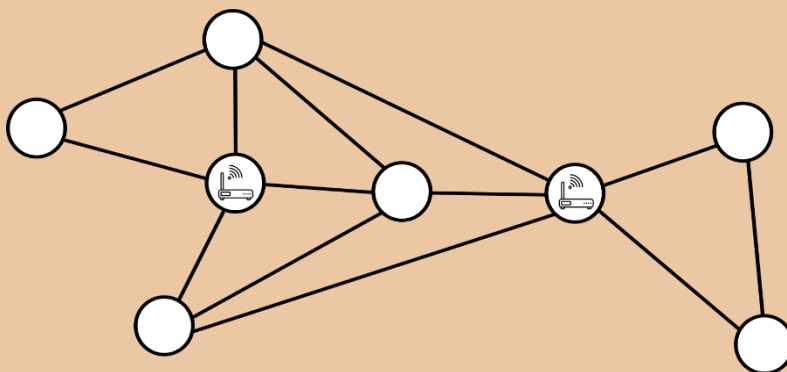


Végül csak egy vár van, amiből négy csatorna indul, a feltételek szerint tehát ez Mari vára. Ez a 4-es számú vár!

MIÉRT INFORMATIKA?

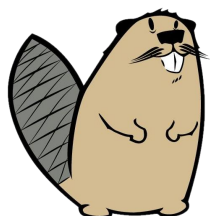
Ebben a feladatban két várat minden esetben egyetlen csatorna köt össze. A várak és csatornák összessége egy hálózatot alkot, amely az összes vár kapcsolatát ábrázolja. A kapcsolatok ilyen hálózatát az informatikában és a matematikában gráfnak nevezzük. Egy gráfot olyan csúcsok összességének lehet tekinteni, amelyek éllel vannak összekötve. Ebben a feladatban a várak ábrázolják a csúcsokat és a csatornák az éleket.

A gráfok tudományát gráfelméletnek hívjuk. Ez a tudomány használható a tárgyak közötti páronkénti kapcsolat modellezéséhez. A gráfok matematikai modellek a természet és a technológia hálószerű struktúráira. Például: szociális struktúrák, úthálózatok, számítógépes hálózatok, elektromos kapcsolások, ellátóhálózatok vagy kémiai molekulák. A gráfok segítséget nyújthatnak a hálózati problémák leírásában és megoldásában például, ha egy jó helyet szeretnénk találni a routernek egy épületben, vagy biztosítani, hogy a ház minden szobájában erős Wi-Fi legyen.



WEBOLDAL

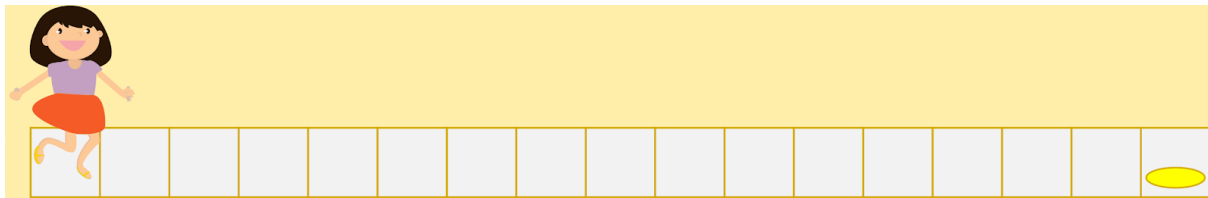
<https://hu.wikipedia.org/wiki/Gr%C3%A1felm%C3%A9let>



UGRÓISKOLA (2022-CZ-04B)

BENJAMIN - KÖNNYŰ

Vera szeret ugróiskolázni. Készített magának egy 17 négyzetből álló pályát. Az utolsó négyzetbe egy érmét tett, ő pedig beállt az első négyzetbe.



A következő szabályok szerint szeretne elugrálni az érméhez minden négyzetbe legalább egyszer beleugorva:

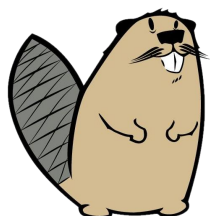
- Ha három előreutató nyíl  áll, akkor 3 négyzettel ugrik előre
- Ha egy visszafele mutató nyíl  áll, akkor egy négyzetet ugrik hátra.

Az első négyzetbe berajzolt 3 előreutató nyilat és előreugrott hármat.

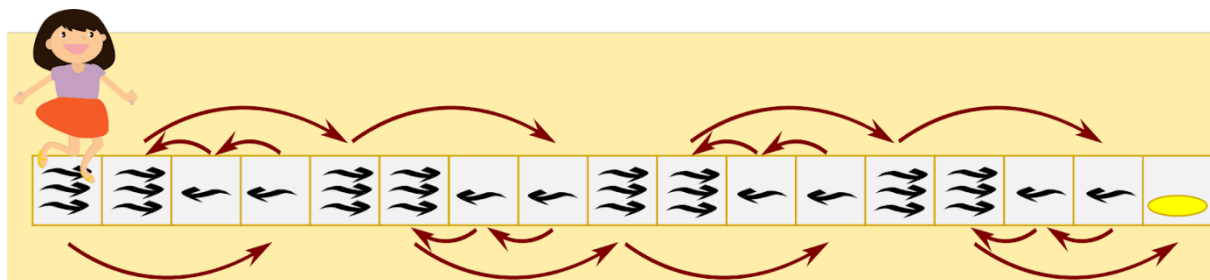


Hogy rajzolja be a további nyilakat, hogy eljusson az érméig és minden négyzetbe legalább egyszer beleugorjon?

- A) 
- B) 
- C) 
- D) 



A helyes válasz a D)



A színezett nyilak mutatják, hogy a D) megoldásnál miképp ugrál Vera.

Az A) válasznál Vera kihagyja balról a második négyzetet. A C) válasznál balról a második és harmadik négyzetet hagyja ki. A B) válasznál hármas ugrásokkal jut el az érméig, az összes visszaugrást tartalmazó négyzetet kihagyva.

MIÉRT INFORMATIKA?

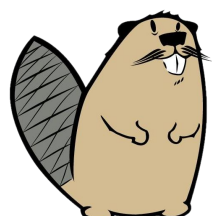
Az egymást követő cselekvések módjának megtalálását egy cél elérésére algoritmizálásnak nevezzük. Egy robot vagy egy számítógépes alkalmazás is algoritmusokon alapuló utasításokkal programozható. Az egyik legalapvetőbb algoritmikus szerkezet az itt is használt egymás utáni végrehajtás, azaz szekvencia.

Ebben a hódfeladatban nem csak a végrehajtás volt a feladat (amit akkor használtál, ha a megoldásokból ellenőrizted le, hogy megfelelő-e az adott utasítás sorozat), hanem egy megoldás megtalálása is. Ez sokszor hosszadalmas, időt és erőforrást igénylő feladat. Itt például a visszalépéses (backtrack) algoritmust is alkalmazhatjuk: amikor elindulunk és ha nem jó a megoldásunk, visszalépünk valameddig.

WEBOLDALAK

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

https://hu.wikipedia.org/wiki/Visszal%C3%A9p%C3%A9s_keres%C3%A9s



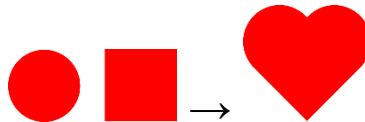
SZÍV GRAFIKA (2022-DE-02)

KISHÓD – NEHÉZ

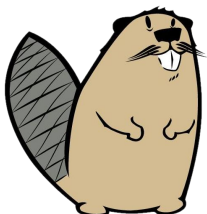
BENJAMIN – KÖZEPES

Tina egy körből és egy négyzetből indul ki, amiket szívvé alakít. Ehhez csak a következő három átalakítást használhatja fel:

- Egy forma tetszés szerinti elforgatása
- Egy forma tetszés szerinti eltolása
- Egy forma megduplázása úgy, hogy mind a kettő ugyanazon a helyen marad.

**Mit és milyen sorrendben csinált?**

- A) Kör megduplázása, négyzet elforgatása, kör eltolása, kör eltolása
- B) Négyzet megduplázása, négyzet elforgatása, négyzet eltolása, kör eltolása
- C) Kör megduplázása, kör elforgatása, kör eltolása, négyzet eltolása
- D) Kör eltolása, kör eltolása, kör megduplázása, négyzet eltolása



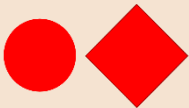
A helyes válasz az A)

Hogyha pontosan megnézzük a szívet, megállapítható, hogy két körből és egy 1/8-fordulattal elforgatott négyzetből áll. Tehát az átalakítások között szükség van egy kör megduplázására, hogy megkapjuk a két kört, és egy négyzet elforgatására, hogy a négyzetet 1/8-fordulattal el lehessen forgatni. Ezek alapján a B), C) és D) válaszok kiesnek, mivel:

- A B) válaszban egy megduplázott kör helyett egy megduplázott négyzet szerepel
- A C) válaszban egy elforgatott négyzet helyett egy elforgatott kör szerepel
- A D) válaszban egy elforgatott forma sem szerepel, tehát a négyzet sem lehet elforgatva

Viszont az A) válasz helyes egyáltalán? A formákat még el is kell tolni.

A következő átalakítások szükségesek:

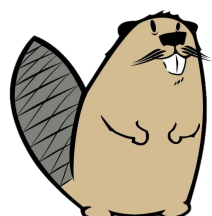
alaphelyzet	
a kör megduplázása után	
a négyzet elforgatása után	
a kör eltolása után	
eredmény	

Ezért az A) válasz a helyes; kör megduplázás, négyzet elforgatás, kör eltolás, kör eltolás

MIÉRT INFORMATIKA?

A képfeldolgozó programokban sok különböző átalakítással lehet egy képet alkotni. Ebben a feladatban olyan átalakítások szerepelnek, mint forgatás, eltolás vagy megduplázás. De ez önmagában még nem elegendő: a számítógépnek meg kell adni, hogy mennyire kell egy formát elforgatni vagy hova kell eltolni.

Természetesen le lehetne írni egy hosszabb utat is, ahogy egy körből és egy négyzetből szívet alkotunk. Az informatikában viszont az a jobb, ha a lehető legkevesebb alapátalakítást használunk, amiket meg lehet ismételni vagy másképp végrehajtani. Ezt általánosításnak hívjuk, amikor



általános érvényű megoldásokat alkotunk egyedi példákból. Például így hangozhatnak a parancsok:

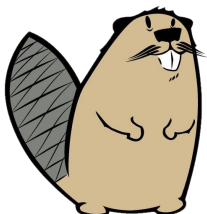
- Egy forma elforgatása: forgasd el a formát, mennyire
- Egy forma eltolása: told el a formát, hova
- Egy forma megduplázása: duplázd meg a formát

Tina képfeldolgozó programja különösnek tűnhet: a program ahelyett, hogy a képet, mint a fotóknál, pixelek formájában mentené el, a formák leírását (például: kör, 2 cm-es sugár, kitöltés színe) tárolja. Így lehetséges, hogy két forma egymáson fekszik, mint a két kör, és utána az egyik a kettő közül elmozdítható anélkül, hogy az alatta lévő felülírná. Az ilyen grafikákat vektorgrafikáknak hívjuk. Ezeket gyakran használják, ha absztrakt formákat kell magas minőségben rajzolni. A többi grafikát pixelgrafikának hívjuk, ezek gyakran fotók vagy fotórealisztikus rajzok. A vektorgrafika a kicsinyítéseknél és nagyításoknál is sokkal hatékonyabb és szebb képet ad.

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Vektorgrafika>

<https://hu.wikipedia.org/wiki/Rasztergrafika>



OKOSSZEMÜVEG (2022-DE-03)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Tina vak és a tájékozódáshoz egy speciális szemüveget használ. A szemüvegen kamerák vannak és egy intelligens tárgyfelismerő rendszerrel rendelkezik, melynek segítségével felismeri az utat ■, mezőt ■, házat ■ és fát ■.

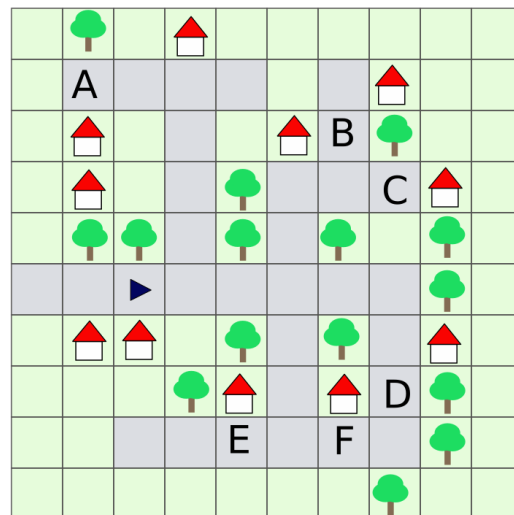
Amikor Tina az úton haladva egy új mezőre lép mindig háttal lesz annak a mezőnek, ahonnan ellépett, és a szemüveg elmondja neki, mi van tőle balra, előtte, tőle jobbra (mindig ebben a sorrendben).

Tina a nyíllal (▶) jelölt mezőn áll és a nyíl irányába, jobbra néz.

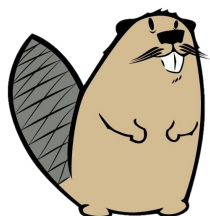
Ekkor ezt hallja: fa, út, ház.

Ezután elindul és a következőket hallja:

fa, út, ház
út, út, mező
fa, út, fa
út, út, út
fa, út, fa
fa, ház, út
út, út, fa
ház, út, fa



Végül Tina a betűkkel jelzett mezők egyikén áll. Melyiken?

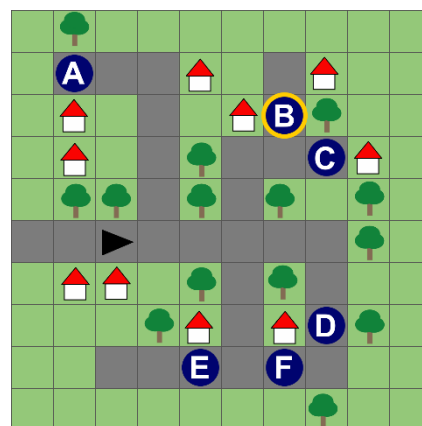


A helyes válasz a B)

Itt az a hatékony módszer, ha a hat célmezőre koncentrálnunk, és megnézzük, hogy az utolsónak hallott „ház, út, fa” kombinációhoz passzolnak-e. Ezzel kizárhatjuk a C-t, az E-t és az F-et és az A-t. Az utolsó előtti lépés az „út, út, fa”, ezzel ki lehet zárni a D-t. Így csak a B marad.

Másik megoldás, hogy megpróbálunk végig menni az úton, úgy, hogy az megfeleljen a Tina szemüvegétől hallottaknak. Az egyetlen út, ami megfelelő, az a B.

Ha Tina útját a szemüveg információi alapján próbáljuk lekövetni, nem mindig lehet rögtön eldönteni, hogy hova lépett Tina. A negyedik lépésnél balra és jobbra is fát látna, mindegy melyik irányba indulna el a három közül. Ebben az esetben muszáj a következő lépés információit is figyelembe venni, hogy a negyedik lépést egyértelműen meg lehessen határozni.



MIÉRT INFORMATIKA?

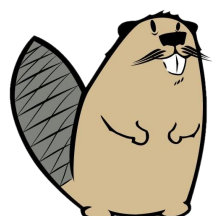
Ez a hódfeladat bemutatja nem csak Tina, de az autonóm tájékozódás lehetőségeit: a rögzített információk (illetve a képek elemzésének) értelmezését követően geometriai adatokra (ház, fa, út) tehetünk szert. Az autonóm tájékozódás elengedhetetlen az önvezető autók esetében, hiszen a látott tárgyakat, jeleket felismerve tudnak közlekedni ezek a járművek.

A megoldásunkhoz pedig a második módon a visszalépéses keresés algoritmusát alkalmazva jutottunk el. Ez egy igen népszerű algoritmus, ami több szituációban is alkalmaznak az informatikusok. A lényege, hogy az ember fokozatosan építi fel a lépéseket, és elhagyja azokat, amiről megállapítható, hogy nem vezet helyes megoldáshoz. A teljes kipróbálással szemben a visszalépéses keresés előnye, hogy nem kell a lépéseket az elejétől fogva újra végrehajtani, mivel csak arra a lépésre tér vissza, ahol a rossz lépés előtt az utolsó döntést hozta, és onnan folytatja.

WEBOLDALAK

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

<https://en.wikipedia.org/wiki/Backtracking>



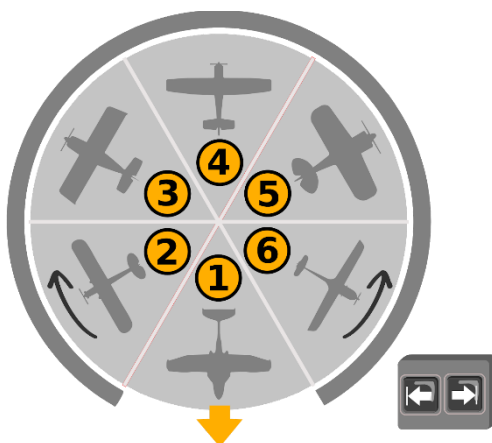
KEREK HANGÁR (2022-DE-05)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

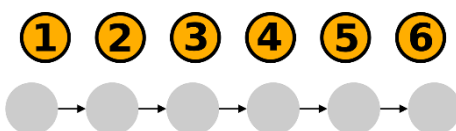
SENIOR – KÖNNYŰ

Hódváros repterén hat repülő parkol egy hangárban, ahol egy forgó korongon hat parkolóhely került kialakításra. A bejáratnál található két, nyilat jelző gomb  . Egy gombnyomással egy parkolóhellyel tudjuk elfordítani a forgó korongot – balra vagy jobbra a nyílnak megfelelően.

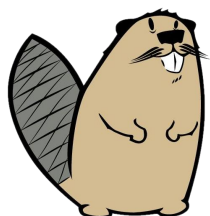


Reggelente, amikor a pilóták kiállnak a repülőjükkel, mindig az 1-es parkolóhely van a hangárajtóban, és az itt parkoló repülő kigurulhat. Legjobb esetben még ötször kell megnyomni a gombokat, hogy az összes többi repülő is kigurulhasson. Például, ha a pilóták 1., 6., 5., 4., 3., 2. sorrendben szeretnék elhagyni a parkolóhelyeket, csak meg kell nyomni ötször a  gombot.

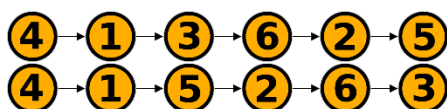
De mi a legrosszabb eset? Milyen sorrendnél kell a legtöbbször megnyomni a gombokat? Mutass egy példát ilyen kedvezőtlen sorrendre!



Húzd a parkolóhelyek számát a legkedvezőtlenebb sorrendbe!



Kettő helyes válasz van:



Hogy megtaláljuk a választ, mindig azt a repülőt kell választanunk, amelyik parkolóhelye a legnagyobb távolságra van a hangárajtótól.

„ 4” jelenti azt, hogy három gombnyomás után kigurult a 4-es parkolóhelyen parkoló repülő.

Minden esetben a 4-es parkolóhely van a legmesszebb, tehát azzal kell kezdeni. Ezután pedig visszaforgatjuk az 1-est, hiszen ott még parkol repülő és az van a legmesszebb. A következő választásnál jobbra is és balra is indulhatunk – emiatt lesz két megoldásunk:

4 1 3 6 2 5:



4 1 5 2 6 3:



Mindkét esetben összesen 16 gombnyomás szükséges.

Nem is lehet több 16 gombnyomásnál, mert csak az elején következhet két hármass fordítás egymás után. Ezt követően legfeljebb két és három fordítás váltakozhat és összesen 6 repülők van.

MIÉRT INFORMATIKA?

A kerek hangárnak az az előnye, hogy a repülőket kevés helyet foglalva le lehet parkolni. A ki- és beparkolás azonban általában tovább tart, mint egy hagyományos hangár esetén.

Az eljárások hatékonysága központi téma az informatikában, mert ez az algoritmusok fontos értékelési szempontja. A hatékonyság leggyakrabban a teljesítéshez szükséges időről szól (futásidő-hatékonyság), de általában az összes szükséges erőforrásra vonatkozik, például a felhasznált memória méretére (memóriahatékonyság), a futási időre, az algoritmus összetettségére.

A mi konkrét hangárról szóló példánkban is az egyik erőforrás megtakarítása egy másik erőforrás szükségletét növeli. Az adott helyzettől és az erőforrások elérhetőségétől függ, hogy melyik erőforrás kap nagyobb szerepet.

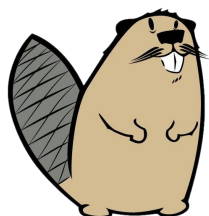
Például a buborékrendeztés és a Tim-rendeztés is sorozatok rendezésére való algoritmusok. A buborékrendeztés időben a sorozatot az elemek számának négyzetével arányosan rendezi ($O(n^2)$), de kevés további memóriát igényel, ami a sorozat hosszához képest állandó.

A Tim-rendeztés sokkal gyorsabban rendez, mint a buborékrendeztés ($O(n \log n)$), de helyigénye lineárisan növekszik a sorozat méretével. Ha egy adott alkalmazáshoz hosszú sorozatok gyors rendezése szükséges, a Tim-rendeztés jobb választás. Ha a rendezés memóriaterületének minimalizálása fontosabb, akkor viszont a buborékrendeztés jobb választás.

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Hat%C3%A9konys%C3%A1g>

<https://hu.wikipedia.org/wiki/Bubor%C3%A9krendezt%C3%A9s>



FILMNÉZÉS (2022-DE-07)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Néhány barát szeretne közösen megnézni egy filmet. Hét film közül tudnak választani. A döntéshez mindenki értékeli az egyes filmeket, hogy szerinte

jó , közepes  vagy rossz .

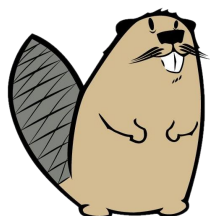
Az eredményt a lenti táblázatban láthatod. Sajnos nem sikerült kedvenc filmet választaniuk.

Egy film akkor „kedvenc”, ha minden barát ezt jelöli meg az egyik legjobbnak. Például az 1. film nem lett kedvenc, mert Niki a 4. filmnek adta a legjobb értékelést.

Anna a lehető legkevesebb barátját szeretné meggyőzni arról, hogy változtassák meg a véleményüket, hogy egy film kedvenc lehessen.

	1	2	3	4	5	6	7
Anna							
Fanni							
Niki							
Gréti							
Eszter							
Rózsa							

Segíts Annának, hogy a lehető legkevesebb értékelés megváltoztatásával sikerüljön kedvenc filmet találni! Melyik film véleményeit kell megváltoztatni?



A helyes válasz a 6. film véleményeinek megváltoztatása

Kezdetben nincs kedvenc. Minden filmnél vannak olyanok, akik egy másikat jobbra értékelték.

Film	Azok, akik más filmet jobbra értékelték
1	4: Fanni, Niki, Gréti és Rózsa
2	3: Niki, Eszti és Rózsa
3	3: Niki, Eszti és Rózsa
4	3: Fanni, Eszti és Rózsa
5	3: Fanni, Gréti és Eszti
6	2: Niki és Rózsa
7	3: Niki, Gréti és Rózsa

A 6. filmnél csak 2 barát van, akik mást jobbra értékelték. A többi filmnél ez még kettőnél is több. Annának tehát meg kell győznie Nikit és Rózsát, hogy javítsanak a 6. film értékelésén. Így két változtatással sikerül kedvenc filmet választani. Tehát az értékelések javítása az egyik lehetséges stratégia.

Niki és Rózsa dönthetnek úgy is, hogy leminősítenek más filmeket: ha Niki csökkenti a 4. film értékelését, és Rózsa csökkenti az 5. film értékelését, akkor is a 6. film lesz a kedvenc. Ebben az esetben is két változtatásra van szükség.

Lehetséges, hogy Rózsa csökkenti az 5. film értékelését, Niki pedig növeli a 6. film értékelését. Vagy Niki csökkenti a 4. film értékelését, Rózsa pedig növeli a 6. filmét. Mindkét esetben a 6. film lesz a kedvenc.

Összességében tehát négyféleképpen lehet módosítani kettejük értékeléseit úgy, hogy végül a 6. film legyen a kedvenc.

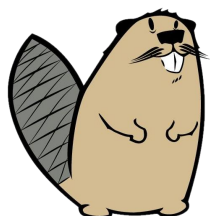
MIÉRT INFORMATIKA?

Hogyan oldjuk meg ezt a feladatot? Minden filmnél és minden barátnál ellenőrizhetjük, hogy az adott barát jobbra értékelt-e más filmeket vagy sem. És így létrejön a fenti táblázat. Ez a táblázat segít megtudni, hogy kik azok, akiknek értékeléseket módosítani kell, hogy a lehető legkevesebb változtatást alkalmazva legyen egy kedvenc.

Anna valóban használhatja ezt az algoritmust a problémája megoldására. Azonban ez egy hatékony algoritmus? Lehetne Anna még gyorsabb?

A következőkben M -el jelöljük a filmek számát és F -vel a barátok számát. Annának mind az $M \cdot F$ értékelés végig kell néznie, és mindegyiket össze kell hasonlítani annak a barátjának a többi filmre adott $M-1$ értékelésével. Összességében Annának meg kell néznie $M \cdot (M-1) \cdot F$ értékelést.

Annak megállapításához, hogy valamelyik értékelés problémás-e, Annának csak össze kell hasonlítani azt az adott személy által adott legjobb értékeléssel. Ha az illető jobban szeret egy másik filmet, akkor az Anna által nézett film nem lehet kedvenc.



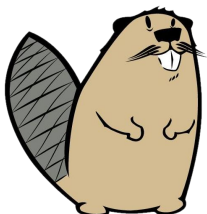
Más szóval, ha Annának először meg kell állapítania emberenként a legjobb értékelést ($M \cdot F$ összes értékelés), aztán az $M \cdot F$ értékelésnél megnézni, hogy az adott értékelés kisebb-e, mint az aktuális barát legjobb értékelése.

Összességében ez az alternatív algoritmus a legjobb értékelések célzott előre kiszámításával ahhoz vezet, hogy Anna csak $2 \cdot M \cdot F$ értékelést néz végig. Ha $M = 7$ és $F = 6$, ez 84 összegzést, míg az első algoritmus 252 összegzést igényel. A második algoritmus is helyesen oldja meg Anna feladatát, de hatékonyabb, mint az első algoritmus.

Az informatika egyik legfontosabb feladata, hogy ne csak megoldja a problémákat, hanem a lehető leghatékonyabban tegye ezt. A gyorsabb számítógépekkel a megoldások gyorsabban kerülnek kiszámításra. Ha azonban nem ismertek hatékony algoritmusok egy probléma megoldására, akkor a gyorsabb számítógépeknek is elérhetjük a határait.

WEBOLDAL

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

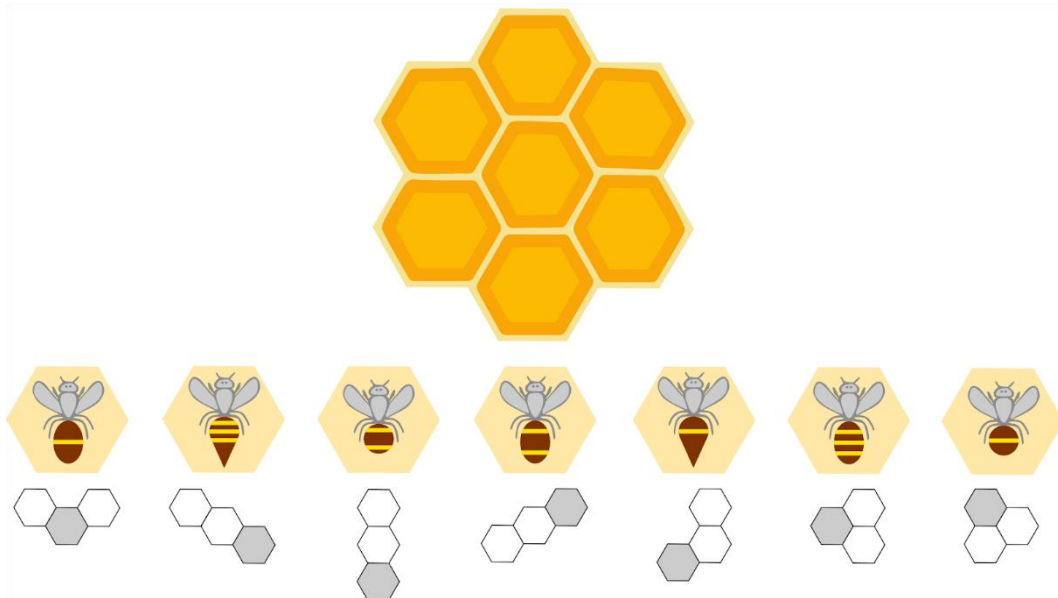


MÉHKAPTÁR (2022-FR-02)

KISHÓD – KÖZEPES

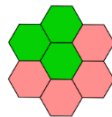
BENJAMIN – KÖNNYŰ

Egy hódnak segítségre van szüksége, hogy be tudja fogni az összes méhet a kaptárába.

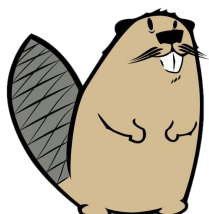


Minden méh alatt található egy szabály, amely szerint mindhárom mezőnek a kaptárban kell lennie és a méh a szürke mezőn van.

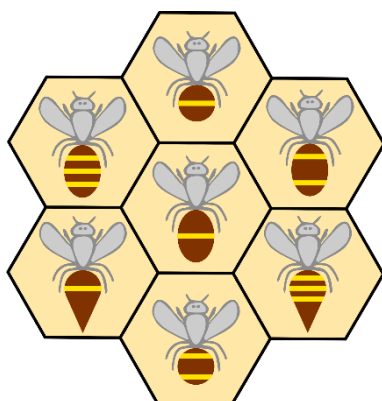
Például a jobb szélső méh elhelyezhető a 3 sötétzölddel jelölt mező egyikében.



Húzd a méheket a kaptárba az alattuk található szabályok szerint!

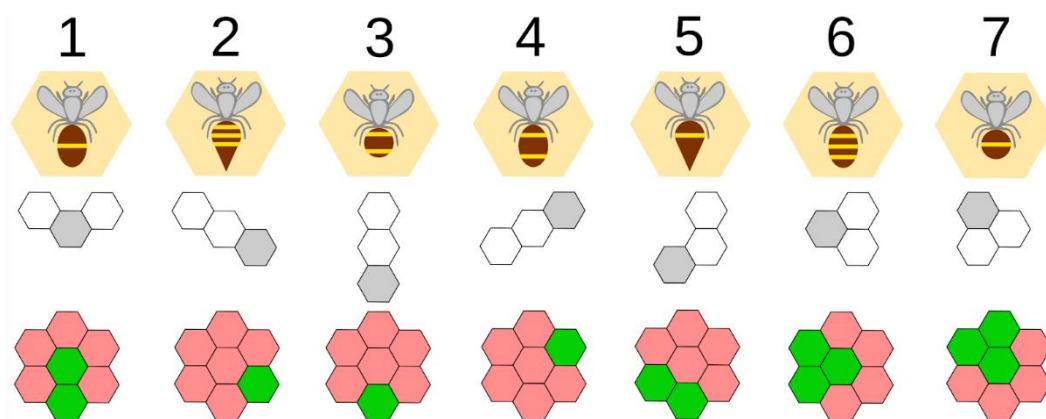


A helyes válasz a következő:



A feladatot megoldhatjuk egyszerűen kipróbálással. Ez azonban nagyon sok időt vehet igénybe. Hogy gyorsabb megoldást találjunk, nézzük meg közelebbről a méhek szabályait. A következő képen az egyes méhek és a hozzájuk tartozó szabályok láthatóak. A sötétzöld mezőkbe helyezhetőek el a méhek.

Látható, hogy a szabályok szerint néhány méhnek több, másoknak kevesebb lehetősége van az elhelyezésre. A 2-es méh esetében például csak egyetlen lehetőségünk van.



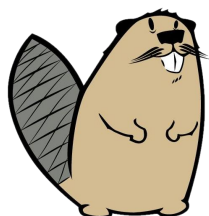
Először azokat a méheket helyezzük el, akiknek csak egyetlen lehetőségük van, azaz a 2., 3. és 4. méheket.

Ezután csak egyetlen lehetőségünk marad az 1. és 5. méhek elhelyezésére. A 6. és végül a 7. méhet is ugyanígy el tudjuk helyezni.

MIÉRT INFORMATIKA?

Ebben a feladatban hét méhet kell hét különböző mezőn elhelyezni. Számos módja van a méhek elhelyezésének. Ha figyelembe vesszük a szabályokat, a lehetőségek száma jelentősen csökken, viszont még mindig túl magas, minden lehetőség kipróbálása túl nagy erőfeszítést jelentene. A feladat gyors megoldásának kulcsa a helyes sorrend. Ebben az esetben a legkorlátozottabb méhekkel kezdünk, amiknek csak egyetlen lehetséges helyük van a kaptárban, és így korlátozzuk a vizsgálendő esetek számát.

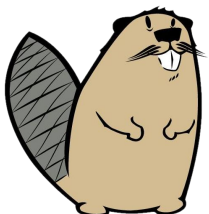
Az ilyen megoldást az informatikában heurisztikának nevezik. Ebben az esetben az ügyes megoldási sorrendnek köszönhetően néhány lépésben meg lehetett találni a helyes megoldást. Néhány probléma esetén azonban, mint például a navigációs rendszer különböző helyei közötti



útvonal tervezése, a heurisztikus megközelítés a pontosság rovására megy. Nagyon sok lehetséges megoldás létezik. Annak érdekében, hogy biztosan megtaláljuk a legjobb útvonalat, a teljes útvonalhálózaton keresztül minden lehetséges útvonalat ki kell számítani, és összehasonlítani, ami óriási számítási ráfordítással járna. Ha először kipróbáljuk azokat a lehetőségeket, amelyek nagy valószínűséggel vezetnek jó megoldáshoz, a számítási ráfordítás jelentősen csökkenthető. Így néhány másodperc alatt megtalálhatjuk a jó útvonalat ahelyett, hogy évekig számolnánk a legjobb után.

WEBOLDAL

<https://hu.wikipedia.org/wiki/Heurisztika>



ANTIVÍRUS (2022-HU-02)

BENJAMIN – KÖZEPES

KADÉT – KÖNNYŰ

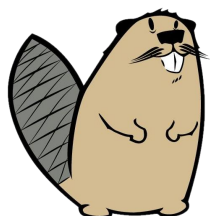
Ati fejlesztett egy új biztonsági programot, ami észlelni tudja a veszélyes szoftvereket. Megvizsgálja, hogy az alábbi listából hány utasítást talál meg bennük:

- mentés a rendszermappába
- beolvasás a rendszermappából
- hozzáadás az automatikus indításhoz
- küldés e-mailben

Ha a program az utasítások között legalább kettő olyat talál, amely a listában megadott összes szót tartalmazza, akkor a szoftver veszélyes és elkülöníti egy mappába.

Az alábbiak közül melyik szoftver lesz elkülönítve?

A)	C)
fájl beolvasás a dokumentumok mappából; fájl megnyitás; fájl mentés a dokumentumok mappába; fájl küldés e-mailben;	fájl beolvasás a rendszermappából; fájl megnyitás; fájl mentés a dokumentumok mappába; fájl küldés e-mailben;
B)	D)
fájl beolvasás a dokumentumok mappából; fájl megnyitás; fájl hozzáadás az automatikus indításhoz;	fájl beolvasás az automatikus indítás mappából; fájl megnyitás; fájl mentés a dokumentumok mappába; fájl küldés e-mailben;



A helyes válasz a C)

A következő táblázatban a feladat szövegében megadott négy utasítás alá van húzva.

Az A, B és D válaszlehetőségekben csak egyet lehet találni a négy utasítás közül.

A)	C)
fájl beolvasás a dokumentumok mappából; fájl megnyitás; fájl mentés a dokumentumok mappába; fájl küldés e-mailben;	fájl beolvasás a <u>rendszer</u> mappából; fájl megnyitás; fájl mentés a dokumentumok mappába; fájl küldés e-mailben;

B)	D)
fájl beolvasás a dokumentumok mappából; fájl megnyitás; fájl <u>hozzáadás az automatikus indításhoz</u> ;	fájl beolvasás az automatikus indítás mappából; fájl megnyitás; fájl mentés a dokumentumok mappába; fájl küldés e-mailben;

MIÉRT INFORMATIKA?

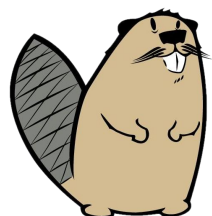
A biztonsági szoftvereket arra fejlesztették ki, hogy észleljék és eltávolítsák a veszélyes programokat, szoftvereket, amik árthatnak a számítógépnek. Néha lehetséges azonosítani egy veszélyes programot (például egy vírust) a fájl nevéből. A legtöbb biztonsági program mindig átvizsgálja a kódot és megpróbálja megtalálni a kritikus részeket a kódban az adatbázisában lévő információkat használva.

A biztonsági szoftverek néha „homokozót” – egy biztonságos környezetet – használnak, ahol le lehet futtatni a kódot és ellenőrizni, hogy az eredmény eltér-e a várttól.

A legújabb szoftver mesterséges intelligenciát (AI) is használ és „tanul”, bővíti az adatbázisát, hogy könnyebben és gyorsabban észlelje a vírusokat.

WEBOLDAL

https://hu.frwiki.wiki/wiki/Logiciel_antivirus



VIGYÁZZ, GALÓCA! (2022-HU-04)

KADÉT – KÖNNYŰ

JUNIOR – KÖNNYŰ

A „Vigyázz, galóca!” játék elején pontosan egy légyölő galóca látható a játéktáblán. Az összes többi mező pedig le van fordítva. Ha felfordítasz egy mezőt, vagy egy másik galóca vagy a szomszédos mezőkön található galócák száma jelenik meg. Ha minden olyan mezőt felfordítasz, amelyen nincs rejtett galóca, akkor nyertél.

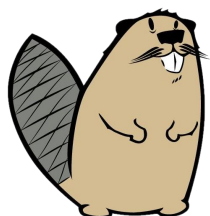
Íme egy példa egy játéktáblára, melyen minden mező látható:

0	1	1	1
1	3		2
1			2
1	2	2	1

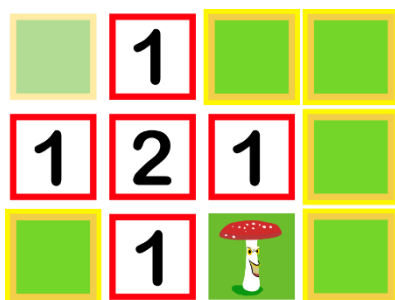
Kezdjünk egy új játékot!

Melyik mezők azok, amelyeken biztosan nincs galóca?

	1		
1	2	1	
	1		



A helyes válasz:



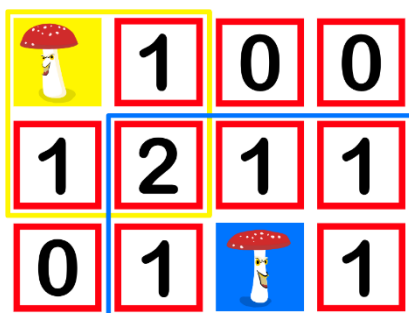
A helyes válasz magyarázatához betűket helyeztünk a mezőkre.



Ha egy mezőn olyan szám szerepel, ahány szomszédján már látunk galócát, akkor biztosan tudhatjuk, hogy a többi szomszédján már nem lesz galóca, csak szám.

- A „D” mezőn emiatt nem lehet galóca, hiszen a jobbra mellette lévő '1'-es szomszédjában már látunk egy galócát.
- A látható galóca feletti '1'-es miatt már a „B”, „C”, „E” és „F” mezőkön sincs galóca.
- Az „A” mezőn viszont biztosan galóca áll, hiszen a szomszédos számok: 1, 2 és 1 különben nem lennének helyesek.

Ha minden mezőt felfedünk tehát, akkor így néz ki a játéktáblánk:

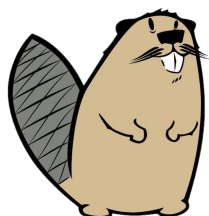


MIÉRT INFORMATIKA?

Hogyan oldottuk meg ezt a problémát?

Ehhez a hódfeladathoz hasonló rejtvények megoldásához csak logikusan kell „gondolkodni”.

De néha egy feltételezéssel kell kezdeni, és abból kiindulva logikusan gondolkodni tovább a lépéseken. Ha hibát találunk (például itt az „1” egy elemre utal, és emiatt nem lehet másik), „visszalépünk” és módosítjuk a legutóbbi feltételezésünket.



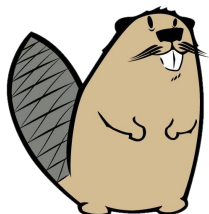
Ez a nagyon népszerű, Visszalépéses keresés (Backtrack) nevű algoritmus. A visszalépés hátránya, hogy néha sokat kell „visszalépni”, ami időt és erőforrást pazarol. És természetesen meg kell határozni egy egyértelmű befejezést, esetleg azt is észrevenni, ha nem létezik megoldás.

A játék eredetileg „aknakereső” néven ismert, és a végtelen változata Turing teljes (Turing-complete), azaz funkcionálisan futtatható egy Turing gépen.

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Aknakeres%C5%91_\(vide%C3%B3j%C3%A1t%C3%A9k\)](https://hu.wikipedia.org/wiki/Aknakeres%C5%91_(vide%C3%B3j%C3%A1t%C3%A9k))

https://hu.wikipedia.org/wiki/Visszal%C3%A9p%C3%A9ses_keres%C3%A9s



ERDEI PARTY (2022-IE-02)

KISHÓD – NEHÉZ

A három barát, Hód Helga, Róka Réka és Medve Marci bulit tartanak. Az alábbi táblázat mutatja, hogy melyikük mit ehet.

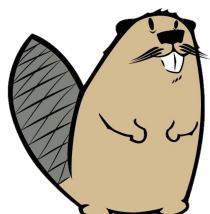
				
 Hód Helga	Igen	Nem	Nem	Igen
 Róka Réka	Nem	Igen	Nem	Igen
 Medve Marci	Nem	Igen	Igen	Igen

Kilenc adag ételük van, és minden barátnak három adagot szeretnének kiosztani.



Oszd el az ételt úgy, hogy minden barát három adagot kapjon! Húzd az egyes ételeket a megfelelő baráthoz!



A helyes válasz a következő:

A feladat megoldását kezdjük Hód Helgával, mert ő az egyetlen, aki eszik leveleket. Ez azt jelenti, hogy minden levél hozzákerül. Ezenkívül az egyetlen étel, amit még eszik a bogyó, tehát a harmadik ételadagja bogyókból áll. A következő Medve Marci, mert ő az egyetlen, aki eszik gombákat, tehát mind a három adag gomba hozzákerül. Róka Réka kapja a megmaradt ételeket: két adag bogyót és a halat.

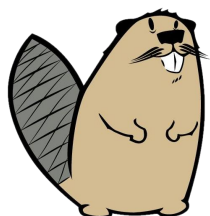
Van más mód is a megoldásra. Például Róka Réka csak halat és bogyókat eszik, tehát kapnia kell két adag bogyót, és a harmadik ételle lehet hal vagy bogyó. Hód Helga az egyetlen, aki eszik leveleket, tehát minden levél hozzákerül. A többi étel közül csak bogyókat eszik, tehát a harmadik ételle ez lesz. Mivel kiosztottuk az összes adag bogyót, ezért Róka Réka harmadik ételle a hal lesz. Medve Marci kapja a megmaradt ételeket: három adag gombát.

MIÉRT INFORMATIKA?

Ennek a feladatnak a megoldásához, amikor több feltételt kell teljesítenünk, legtöbbször a logikát használhatjuk.

Maga a feladat egy logikai tulajdonságtáblázatot mutat be, és a kihívás az, hogyan kell ezt olvasni. Egyértelmű megszorítás van (igen vagy nem) minden állatra és élelmiszerre, ami megmondja, hogy eheti-e az adott típusú ételt vagy sem. Ezenkívül minden ételadag csak egy állathoz kerülhet, és ennek az elosztásnak meg kell felelnie a logikai táblázatban foglaltaknak. Ha először megkeressük az egyértelmű részeket, és azokat elhelyezzük a megoldásunkban, más feltételek is szűkülnek.

Az informatikában használnak olyan programozási nyelveket (mint pl. a Prolog), melyek logikai kifejezéseket fogalmaznak meg és következtetéseket hajtanak végre. Alapgondolatuk, hogy a programozó csak azt mondja meg, mit kell csinálni, a megoldás módját a program maga keresi meg. A logikai tételbizonyítások alapján működő nyelvet a természetes nyelv leírására, kockázatelemzésre és mesterséges intelligenciát megvalósító alkalmazásoknál is használják.

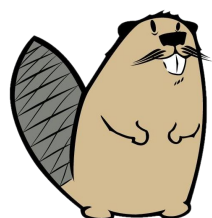


WEBOLDALAK

<https://hu.wikipedia.org/wiki/Logika>

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

<https://web.archive.org/web/20041119191330/http://www.sulinet.hu/termesztvilaga/archiv/2000/0014/07.html>



MŰVÉSZET HAGYOMÁNYOSAN (2022-IN-01)

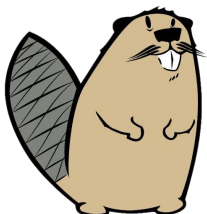
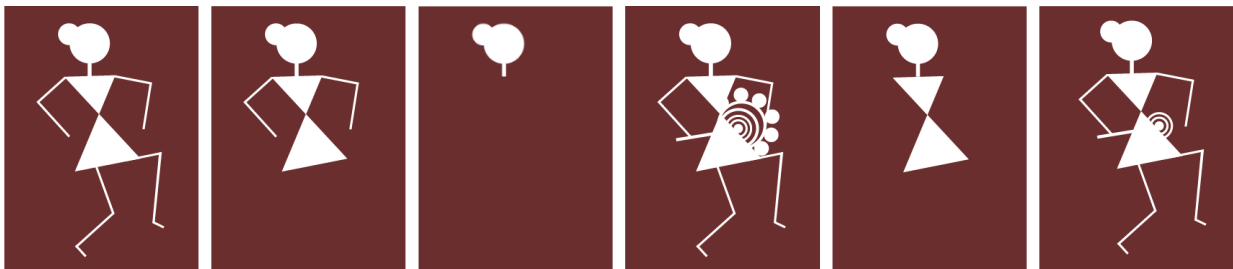
KISHÓD – KÖNNYŰ

Deepa a Warli festést tanulja. A következő képet szeretné lefesteni:

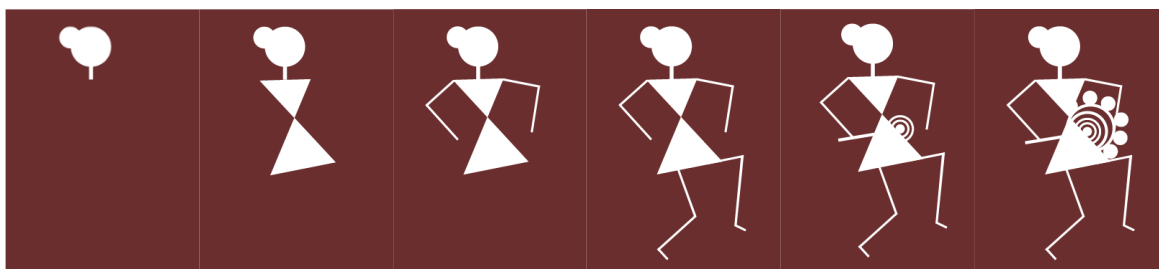


Útmutató kártyák segítségével lépésről lépésre haladna. De a kártyák leestek és összekeveredtek.

Helyezd ismét a helyes sorrendbe a kártyákat:



A kártyák helyes sorrendje:



Minden kártyán az előzőhöz képest valami új mozzanat, részlet kerül megjelenítésre.

MIÉRT IINFORMATIKA?

Egy képet különböző rétegekből is fel lehet építeni. Fontos, hogy a rétegek egymáson helyezkedjenek el és sorrendjük van. A fentebbi feladat megoldásánál a képet egy meghatározott sorrendben tudjuk csak előállítani a kártyákból. Minden egyes kártya egy újabb elemet tartalmaz amit, ha egymásra pakolunk akkor a kész képet kapjuk meg. Úgy érdemes a rétegekre gondolni, mint átlátszó fóliákra, melyek a háttér előtt helyezkednek el. Ezeken a fóliákon bármi lehet. Ebből a felfogásból máris következik, hogy amelyik fólia felül van, az azon levő képdarab, vagy alakzat kitakarhat az alatta levő fólián található képdarabból, vagy bármiből, ami éppen az előbb említett fólia alatt van

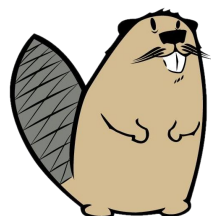
Manapság az összes profibb képszerkesztő program ilyen rétegekkel dolgozik, de a rétegek megfigyelhetők az operációs rendszer kezelőfelületén is, amikor több ablakot egymásra pakolva használunk.

A Warli festészet a törzsi művészet egyik formája, amely az indiai Maharastra törzstől ered. Alapvető geometriai alakzatokat használnak, és az egyetlen használt szín a rizsliszt és víz keverékéből készült fehér pigment.

WEBOLDALAK

<https://gobertpartners.com/where-warli-painting-is-used>

http://tehetseg.inf.elte.hu/tananyagok/kepszerk/lecke11_lap1.html



KAGYLÓK ÉS KÖVEK (2022-IT-02)

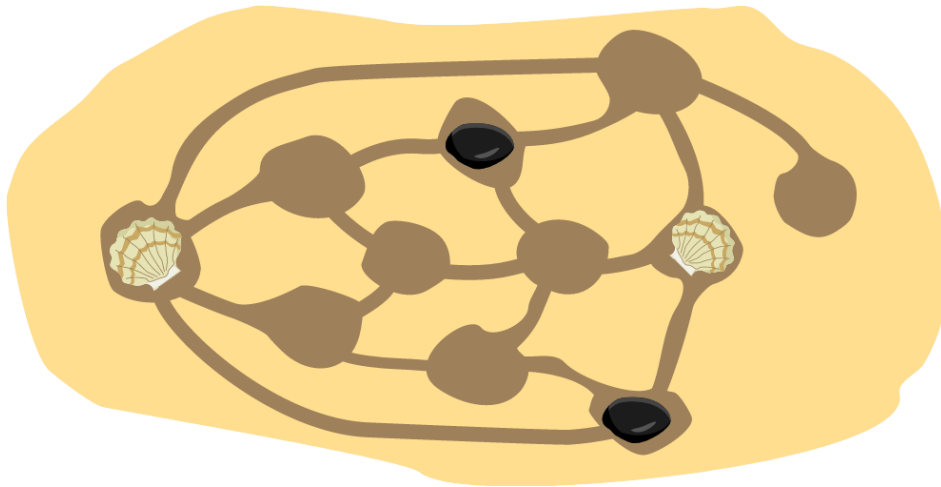
JUNIOR – NEHÉZ

SENIOR – KÖZEPES

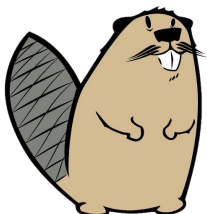
Anna és Bence a tengerparton játszanak. A homokba ásnak pár mélyedést és néhányat árkokkal kötnék össze. Anna kagylókat, Bence pedig kavicsokat használ bábunak.

Felváltva helyezik a bábujukat egy-egy szabad mélyedésbe. Az a játékos, aki két saját bábuját két, árkokkal közvetlenül összekötött mélyedésbe helyezi, veszít. A képen néhány lépés után látható a játék állása.

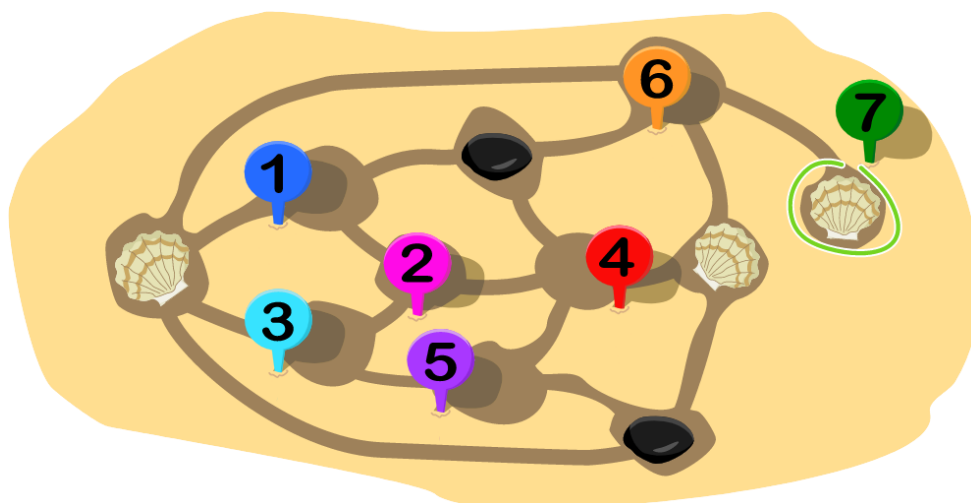
Ezután néhány formát felcserélt egymással.



Anna következik. Melyik szabad mélyedésbe helyezze a következő kagylóját, hogy biztosan nyerjen?

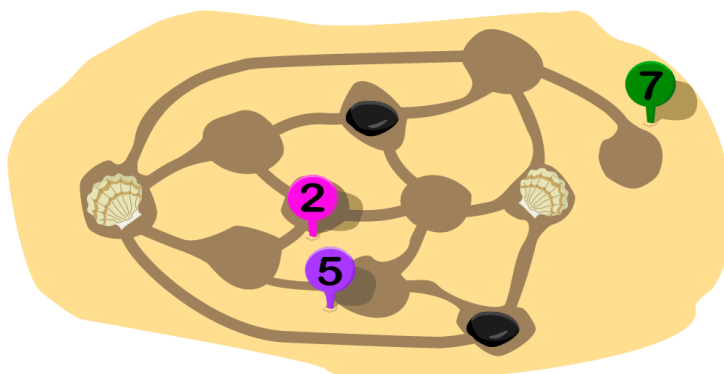


A helyes válasz a képen 7-sel jelölt hely:

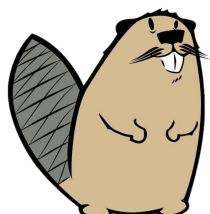
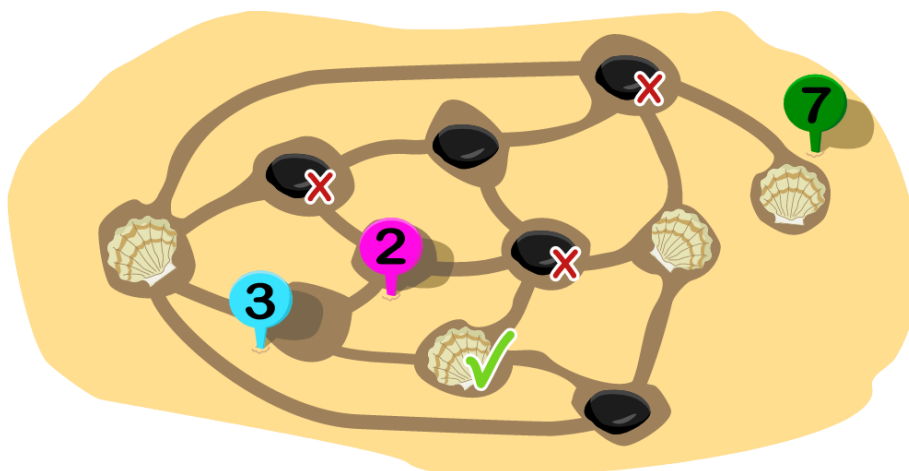


Anna következik. Hogy ne legyen két árokkal közvetlenül összekötött mélyedésben kagyló, az 1, 3, 4 és 6 jelzésű mélyedések szóba sem jöhetnek.

Tehát marad a 2-es, 5-ös vagy 7-es mélyedés.



Ha Anna a 7-es mélyedést választja, Bencének vagy 2-es, vagy a 3-as mélyedésbe kell a következő kavicsát elhelyeznie. Ezután Anna mindkét esetben beteheti a következő kagylót az 5-ös mélyedésbe. De a következő lépésben Bence már nem tud a szabályoknak megfelelően kavicsot elhelyezni, így veszít.



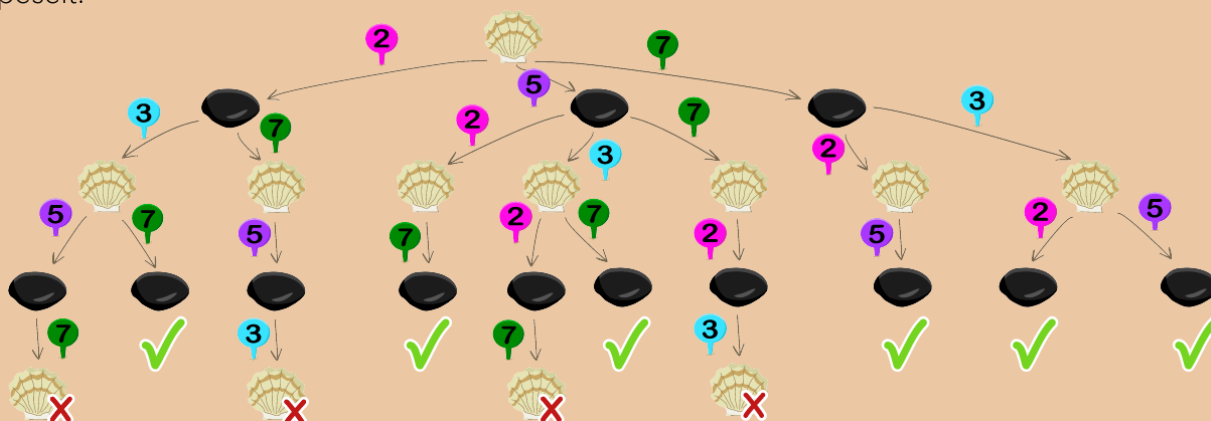
Ha Anna a 2-es mélyedést választja, Bence a 7-est (hiszen győzni szeretne), majd Annának már csak az 5-ös marad. Bence ezután még a 3-as mélyedésbe helyezheti a kavicsát. Annának ezek után azonban már nem marad olyan mélyedés, amelyet nem köt össze árok egy már kagylót tartalmazó mélyedéssel. Tehát veszít.

Ugyanez játszódik le, ha Anna az 5-ös mélyedést választja. Bence a 7-eset, ami után Annának már csak a 2-es lehetséges. Bence a 3-asba helyezve a kavicsot pedig megnyeri a játékot, hiszen Annának megint nem maradt lehetősége elhelyezni a kagylóját.

De sajnos, ha Anna átgondolja, Bence akkor sem nyerhetne, ha ő következne...

MIÉRT INFORMATIKA?

Egy úgynevezett játékfá segítségével szisztematikusan bemutathatjuk Anna és Bence lehetséges lépéseit:



Ezen a játékfán láthatod, hogy Anna mely lépésekkel tudja megszerezni a győzelmet: a jobb oldali ágba, amely akkor kezdődik, amikor Anna a 7-es mélyedést választja, csak azok a helyzetek lehetségesek, amelyekben ő nyer.

A játékelmélet a matematika egy speciális területe. Itt olyan játékok kimenetelére vonatkozó kijelentéseket veszik figyelembe, amelyekben két vagy több játékos felváltva lép és a véletlen nem befolyásolja a lépésüket. Az informatika az ilyen játékfák értékelésére szolgáló algoritmusokkal is foglalkozik. A megfelelő számítási teljesítménnyel rendelkező számítógépek már játszhatnak és nyerhetnek emberek ellen olyan játékokban, mint a sakk. A játékelmélet azonban modelleket is kínál olyan komplex pszichológiai, közgazdasági és egyéb rendszerek működéséhez, amelyekben a „játékosok” interakcióba lépnek. Gondoljunk csak például a vásárlók vásárlási szokásainak módosulására árváltozások esetén vagy a közúti forgalomban az útvonalválasztásra.

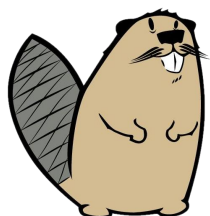
Anna és Bence játéka a „COL” egy példánya. Ez egy kétjátékos játék, amelyet Colin Vout mutatott be, és John Horton Conway (*1937-†2020) matematikus, On Numbers and Games (1976-ban megjelent) könyvében is fontos szerepet játszott.

WEBOLDALAK

<https://hu.wikipedia.org/wiki/J%C3%A1t%C3%A9kelm%C3%A9let>

[https://en.wikipedia.org/wiki/Col_\(game\)](https://en.wikipedia.org/wiki/Col_(game))

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

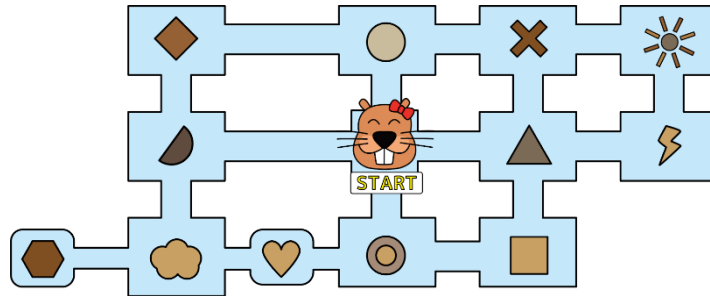


HOL VAN A HÓD (2022-JP-02)

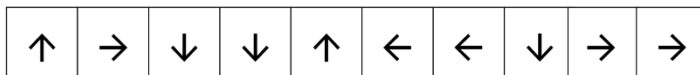
BENJAMIN – KÖNNYŰ

Bibako az alábbi térkép segítségével el szeretne jutni a START-tal jelölt szobából Be-taro szobájába.

A térképen minden szobát egy szimbólummal jelölt meg.



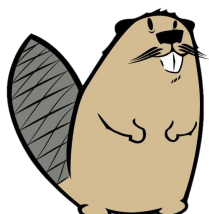
Ezután a következő nyilak szerint mozgott:



Minden nyíl azt jelzi, hogy melyik irányba kell mennie Bibakonak egyik szobából a következőbe.

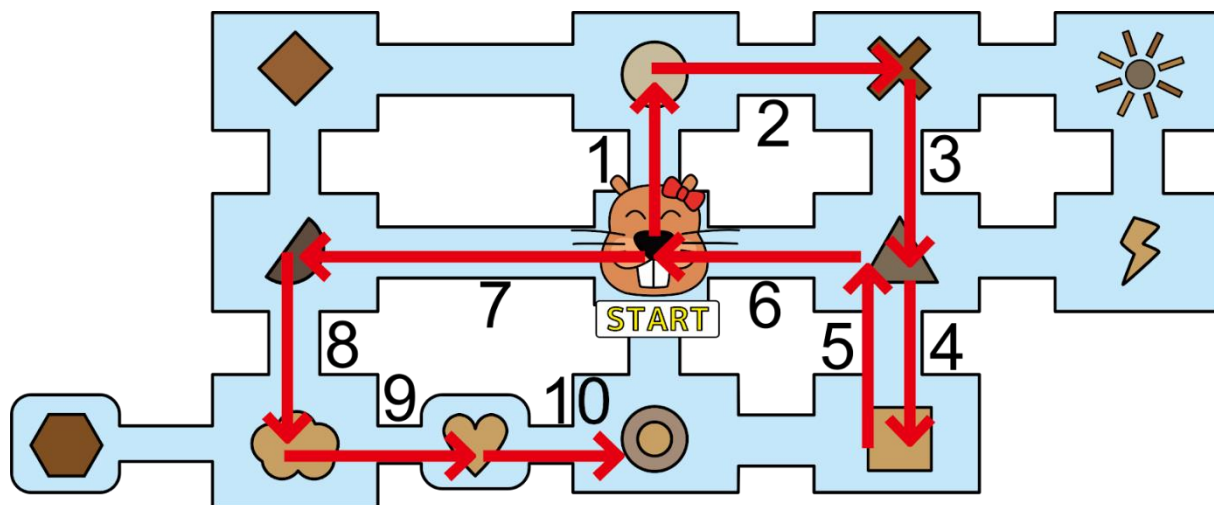
Melyik szimbólummal jelölte Be-taro szobáját?

- A)
- B)
- C)
- D)



A helyes válasz a B: 

Bibako a piros nyilak szerint mozog a térképen.



Tehát az utolsó nyíl után a -vel jelölt szobában lesz.

MIÉRT INFORMATIKA?

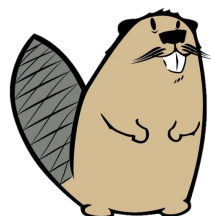
Ebben a feladatban minden nyíl azt mutatja Bibakonak, hogy mit kell tennie a következő lépésben. A nyilak teljes sorozata határozza meg az utat. Ez a feladat a szekvenciát, azaz az utasítások sorban történő végrehajtását mutatja meg. A programozás során ez a legalapvetőbb algoritmikus szerkezet, amit a programozók használnak.

A szekvencia önálló használata azonban nagyon lekorlátozza azt, hogy milyen problémákat oldhatunk meg az adott algoritmussal. Manapság már nem csak szekvenciákat használunk, hanem különféle utasításformákat például feltételhez kötünk, vagy többszörösen alkalmazzuk azokat egymás után, ezzel hatékonyabbá és egyszerűbbé téve az algoritmusokat. Ezt strukturált programozásnak hívjuk, amelynek alaptételét 1966-ban két olasz tudós, Corrado Böhm és Giuseppe Jacopini fektette le.

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Utas%C3%ADt%C3%A1s_\(informatika\)](https://hu.wikipedia.org/wiki/Utas%C3%ADt%C3%A1s_(informatika))

[https://hu.wikipedia.org/wiki/Program_\(informatika\)](https://hu.wikipedia.org/wiki/Program_(informatika))



HÓDBURGER RECEPT (2022-KR-03)

KISHÓD – NEHÉZ

BENJAMIN – KÖZEPES

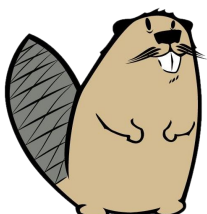
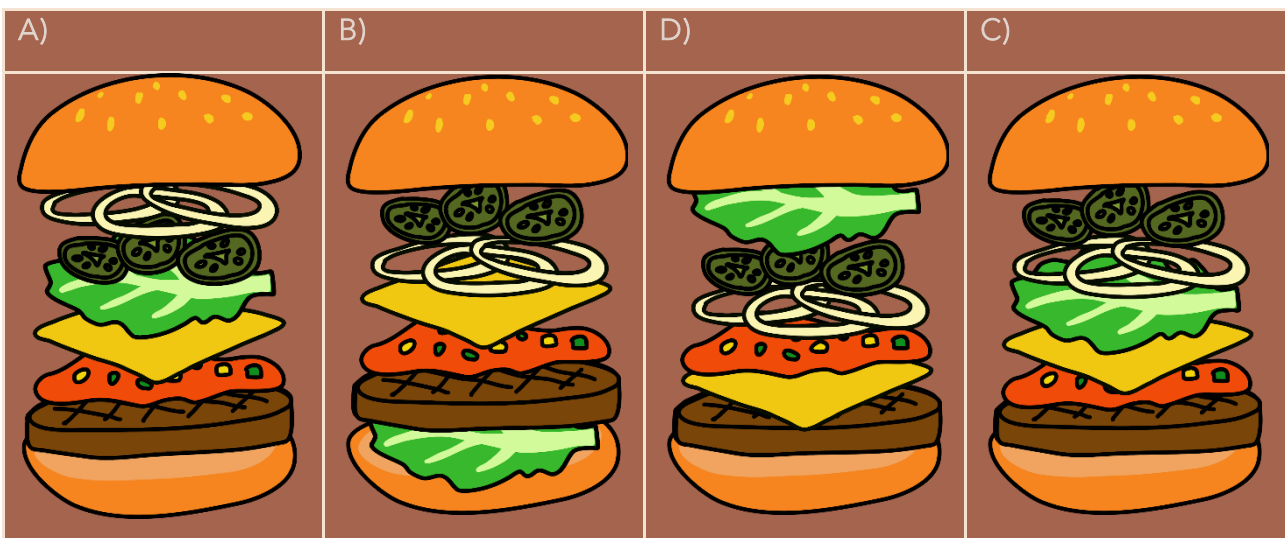
KADÉT – KÖNNYŰ

A hódburger hozzávalói a következők:

zsemle	hús	szósz	uborka	saláta	hagyma	sajt
						

Egy valódi hódburger az alábbi szabályoknak tesz eleget:

1. A szósz közvetlenül a húson van.
2. A hús és a sajt az uborka, a saláta és a hagyma alatt van.
3. A hagymák nem érnek hozzá a zsemlékhez.

Melyik hódburger készülhetett a három szabály alapján?

A helyes válasz a D)

A helyes válasz a D. Ahhoz, hogy eljussunk a megoldáshoz, meg kell nézni az összes hódburgernél, hogy úgy van-e összerakva, hogy teljesül a megadott három szabály.

(A) Ez a hódburger megfelel az 1. és a 2. szabálynak. Viszont a hagyma hozzáér a felső zsemléhez, ezért nem teljesül a 3. szabály.

(B) Ez a hódburger az 1. szabály alapján megfelelő, de a saláta a hús és a sajt alatt van, tehát a második szabály nem teljesül.

(C) Ez a hódburger megfelel a második szabálynak, mivel a hús és a sajt az uborka, a saláta és a hagyma alá kerültek. A harmadik szabály is teljesül ennél a hódburgernél, mivel a hagyma nem ér hozzá egyik zsemlefeléhez sem. Viszont a szósz nem közvetlenül van a húson, ezért az első szabály nem teljesül.

(D) Ez a hódburger az összes szabály alapján megfelelő, ezért a D egy igazi hódburger.

MIÉRT INFORMATIKA?

Ebben a feladatban a hóburgerek három szabály szerint készülnek el. Minden egyes hóburgernél követni kell mind a három szabályt. Mind a három szabály egy-egy feltétel, amelyeket mindegyik burgernél teljesíteni kell ahhoz, hogy hódburger legyen.

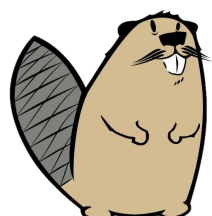
Az informatikában gyakran kell ellenőrizni a feltételeket ahhoz, hogy megtaláljunk egy megoldást. Ebben az ellenőrzésben a szabályok (feltételek) „és” kapcsolattal vannak összekötve. Ez azt jelenti, hogy minden szabálynak (feltételnek) egyszerre kell teljesülnie.

Az adott megoldás ellenőrzése, hogy valamennyi feltételt teljesít-e, egy alapjaiban véve más feladat, mint egy lehetséges megoldás találása. Ezt nevezik „Constraint Satisfaction Problem-nek” (kényszer kielégítési problémának). Legtöbbször sokkal nehezebb egy olyan megoldást találni, ami minden szabálynak megfelel, mint ellenőrizni, hogy egy megoldás teljesíti-e a szabályokat. Ez a számítógépekre is igaz.

WEBOLDALAK

https://wikihuhu.top/wiki/Constraint_satisfaction

<https://hu.wikipedia.org/wiki/Konjunkci%C3%B3>



HÓDBAJNOKSÁG (2022-KR-06)

SENIOR – NEHÉZ

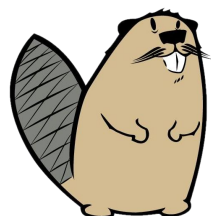
A 3 fordulós hódbajnokságon 8 hód vesz részt. Az egyes fordulókban a hódok pontokat gyűjtenek.

- Az 1. fordulóban két, négy hódból álló csapat alakul véletlenszerűen. Az egyes hódok pontjait összegzik. A legtöbb pontot szerző csapat nyer, és továbbjut a második fordulóba. A vesztesek folytatják a játékot az 5-8. helyekért.
- A 2. fordulóban ugyanazok a szabályok. A csapatok most két-két hódból állnak. A győztesek a döntőbe jutnak. A vesztesek tovább játszanak a 3. és a 4. helyért.
- A 3. fordulóban nem csapatok, hanem az idáig eljutott két hód mérkőzik meg egymással.

Ada  hód a hódbajnokság győztese. Az alábbiakban az egyes hódok fordulónkénti pontjai láthatóak.

Név	Ada 	Brown 	Candy 	Daisy 	Eden 	Funny 	George 	Hugh 
1. forduló	15	16	19	18	17	20	19	19
2. forduló	20	27	30	24	28	24	30	30
3. forduló	10	14	11	15	16	13	9	12

Melyik három hód volt Ada csapattársa az 1. fordulóban?



Az 1. fordulóban Ada három csapattársa: Daisy 🐿️, Funny 🐿️ és George 🐿️.

A döntőt egyéniben rendezik. George az egyetlen hód, akinek kevesebb pontja van, mint Adának, tehát a 2. fordulóban biztosan ugyanabban a csapatban voltak.

A 2. fordulóban együtt 50 pontot értek el. Ennek az értéknek magasabbnak kell lennie, mint a másik csapat összpontszáma. A két hód, Daisy és Funny az egyetlen pár, amelynek összesített pontja kevesebb, mint 50. Ezért biztosan ugyanabban a csapatban voltak, mint Ada és George az 1. fordulóban.

Most, hogy megtaláltuk az 1. fordulóban Ada csapatából mind a négy hódot, az 1. forduló második csapatának összetételét is ismerjük.

Az 1. fordulóban Ada csapata (Ada, Daisy, Funny, George) 72 pontot ért el. A másik csapatnak (azaz Brown, Candy, Eden, Hough) csak 71 pontja volt. Így Ada csapata nyert.

A 2. fordulóban (Ada, George) 50, míg (Daisy, Funny) csak 48 pontot ért el. A 3. fordulóban, azaz a döntőben Ada 10 ponttal nyert George 9 pontja ellen. Így nyerte meg Ada a hódbajnokságot.

MIÉRT INFORMATIKA?

A feladat megoldásához ki tudjuk alakítani az első kör összes lehetséges csapatát. Ha ismerjük a két csapat egyikét, ismerjük a másik csapatot is. Összességében $73 = 35$ kombináció van. Ezen kombinációk mindegyikénél meg kell néznünk az 1., 2. és a döntő eredményeit, mielőtt eldönthetjük, hogy valójában kik voltak Ada csapattársai az 1. fordulóban. Ez sok időt vesz igénybe.

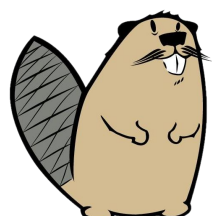
Egy ilyen feladat megoldása érdekében az informatikusok hatékonyabb megközelítésekre törekednek. Ahelyett, hogy előre haladnánk, azaz az elsőtől a harmadikig, visszafelé is levezethetjük a helyes megoldást. Ez akár nagyon gyorsan is megtörténhet, ahogy azt a fenti magyarázatban láttuk.

Ezt a módszert fordított keresésnek nevezik. Azokban a helyzetekben használják, amikor olyan megoldást keresnek, amelynek meg kell felelnie bizonyos feltételeknek. Egyes esetekben az előre és visszafelé keresést kombinálják, hogy megoldást találjanak.

Az előre és a visszafelé történő keresés két gyakori problémamegoldó stratégia. Minden tudományterületen, nem csak az informatikában, használják problémák megoldására.

WEBOLDAL

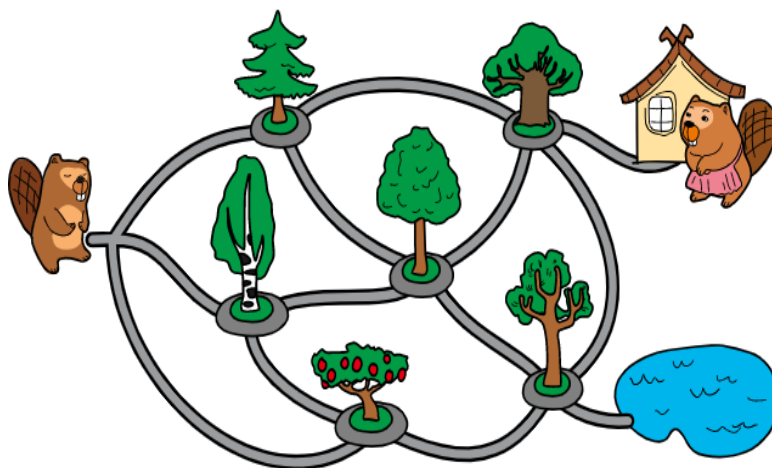
https://hu.wikipedia.org/wiki/K%C3%A9rt%C3%A1ny%C3%BA_keres%C3%A9s



ÖSVÉNY A NAGYMAMÁHOZ (2022-LT-08)

KISHÓD – KÖNNYŰ

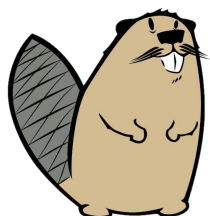
Egy hód szeretné meglátogatni a nagymamáját. Számos különböző, fákkal szegélyezett ösvény vezet a nagymama háza felé.

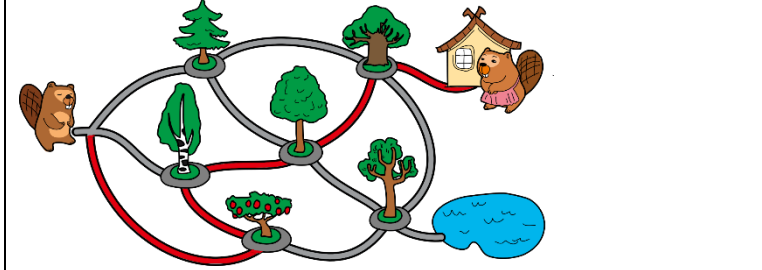
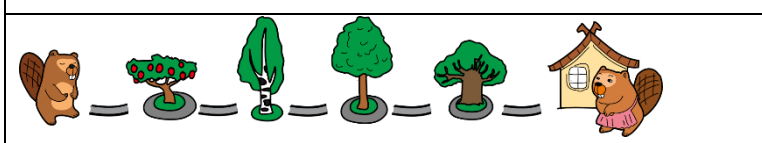
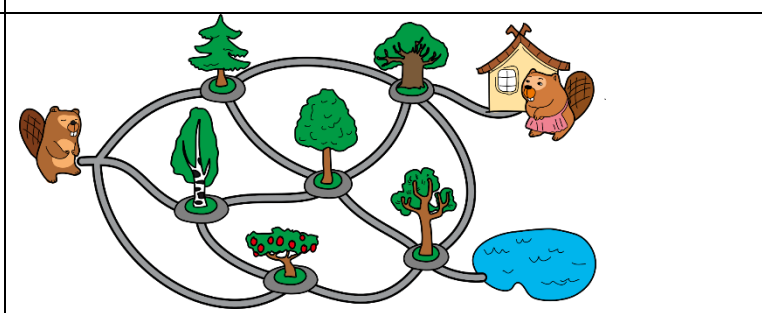
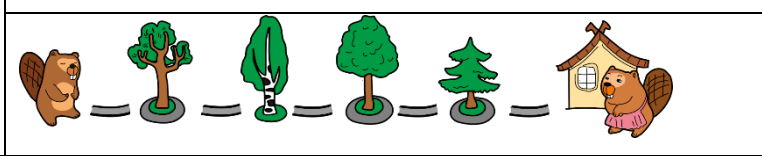
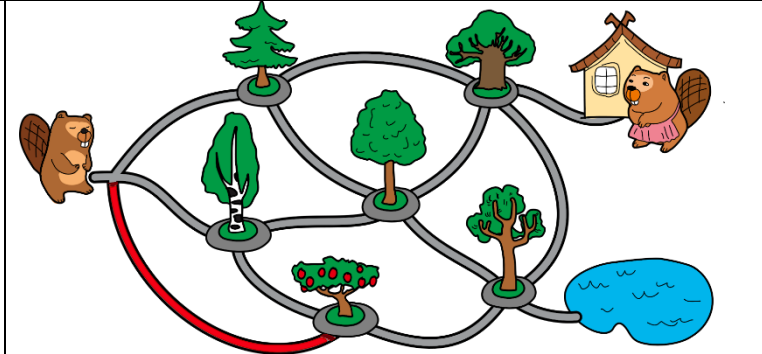
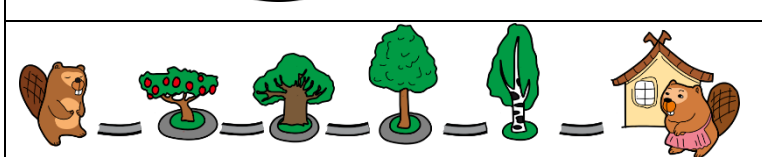


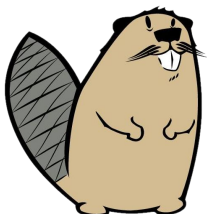
A következő ösvények közül melyik vezet a nagymama házához?

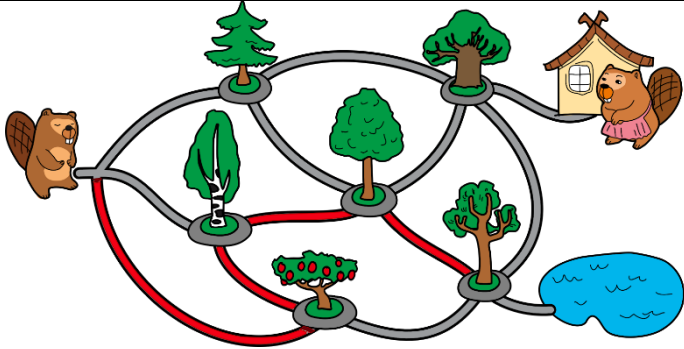
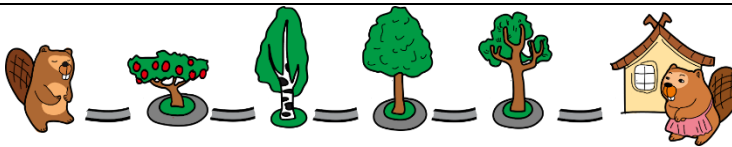
- A)
- B)
- C)
- D)

A helyes válasz az A)



A)	 	Ez a helyes válasz.
B)	 	Ez nem lehet az első fa, ami mellett a hód elhalad. 
C)	 	Ez lehet az első fa a nagymamához vezető úton, de innen a hód nem tud eljutni a következő fához.  



D)		Ez az ösvény egy tóhoz vezet, nem a nagymama házához.
		

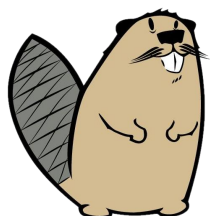
MIÉRT INFORMATIKA?

Az útvonaltervezés (más néven navigációs probléma) egy számítási probléma, azaz valamilyen algoritmussal megoldható. Ebben a feladatban keresünk egy olyan sorozatot, ami az objektumot (hódot) a forrásból a célba (nagymama háza) mozgatja úgy, hogy közben adott objektumokat (fákat) érintünk. Ezen problémák alapjai a nagyon sokszor használt (útvonaltervezés, várostervezés, adattárolás) gráf struktúrák, amelyekkel kapcsolatban sokféle probléma megfogalmazható. Talán a legismertebb ilyen probléma a legrövidebb út megtalálása. A probléma megoldására a korábban említett algoritmusok közül rengeteg félélt választhatunk.

Egy neves holland informatikus, matematikus, Edsger Wybe Dijkstra is készített egy algoritmust, amelyet róla is neveztek el. Manapság ez a legelterjedtebb útkereső algoritmus. Esetében a probléma az volt, hogy melyik út a legrövidebb Rotterdamból Groningenbe. A megoldása annyira jól sikerült, hogy a világon szinte már mindenki ismeri a nevét.

WEBOLDAL

https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%A1si_probl%C3%A9ma



AMŐBA (2022-LV-03)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Az amőba egy kétszemélyes játék az alábbi szabályokkal:

A 3 x 3 mezős táblán a két játékos felváltva helyezi el a jelét (az egyik játékos X, a másik O) egy szabad mezőn. Az a játékos nyer, aki először helyez el három ugyanolyan jelet egy sorban, oszlopban vagy átlósan. Ha az összes mező betelt, és senki sem nyert, a játék döntetlennel végződik.

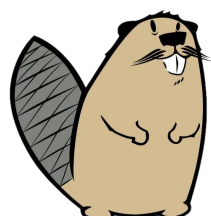
Például a következő képek egy játék első lépéseit és az utolsó lépést mutatják. (Az adott lépés szürkével van kiemelve.)

		X			O		X			O		X			O		X
																O	
																	O

A játék végén létrejött helyzetet érvényes véghelyzetnek nevezzük. Nem minden lehetséges helyzet érvényes véghelyzet, mert a játékszabályok pontosan meghatározzák, hogy mikor ér véget egy játék.

A következő helyzetek közül melyik mutat érvényes véghelyzetet?

A)	B)	C)	D)
X O X	X O X	X X O	X O X
O X O	O X	O X	O X O
O O X	O X X	O O X	O X



A helyes válasz a C)

A C) válasz helyes, mert az egyik játékos nyert (három O egy átlóban), és ezután nem történt több lépés.

Az A) válasz helytelen. X játékos nyerte a játékot, de O játékos több O jelet írt az üres mezőkbe. Mivel mindig a győztes helyezi el az utolsó jelet, sosem lehet kevesebb jele, mint a vesztesnek.

A B) válasz helytelen, mert 5 X-et tartalmaz, de csak 3 O-t. Ez nem lehetséges; mert az O és X jelek száma legfeljebb 1-gyel térhet el.

A D) válasz helytelen; mert még nincs nyertes, és a mezők sincsenek teljesen kitöltve.

MIÉRT INFORMATIKA?

feladat megoldása során ellenőriztük, hogy a válaszlehetőségek négy képe érvényes véghelyzetet mutat-e. A játékszabályokból új szabályokat vezethetünk le az érvényes véghelyzetekre vonatkozóan, például a következőket:

- 1) Az X és az O száma közötti különbség 0, -1 vagy 1.
- 2) Ha egyik játékos sem nyert, minden mezőt ki kell tölteni.
- 3) A vesztes legfeljebb annyi jelet helyezhet el, amennyit a győztes.
- 4) Egy befejezett játékban legfeljebb egy, három azonos jelből álló sor lehet.

Ezek az új szabályok nem játékszabályok, hanem csak annak ellenőrzésére szolgálnak, hogy egy helyzet érvényes véghelyzet-e. Ha egy kép ütközik e szabályok bármelyikével, az nem lehet érvényes befejezés.

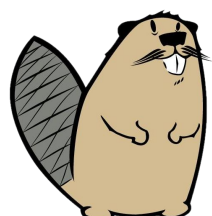
A szabályok nagyon fontosak az informatikában. Nem csak egy lehetséges megoldás megtalálásában segítenek, de azt is támogatják, hogy le tudjuk ellenőrizni, hogy egy megoldás érvényes-e, megfelel-e a megfogalmazott szabályoknak.

Amikor például egy programot írunk, és le szeretnénk fordítani a számítógép nyelvére, egy úgynevezett interpreter ellenőrzi, hogy a beírt szöveg megfelel-e a programozási nyelv szintaktikai szabályainak.

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Am%C5%91ba_\(j%C3%A1t%C3%A9k\)](https://hu.wikipedia.org/wiki/Am%C5%91ba_(j%C3%A1t%C3%A9k))

[https://hu.wikipedia.org/wiki/%C3%89rtelmez%C5%91_\(informatika\)](https://hu.wikipedia.org/wiki/%C3%89rtelmez%C5%91_(informatika))



MÁRIA KINCSET KERES (2022-MK-01)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

Mária talál egy titokzatos ládát, amiben egy kincs van elrejtve. Sajnos a láda le van zárva. Ahhoz, hogy kinyíljon, ki kell találnia, mi a 3 alakzattól álló kód.

Az alábbi útmutatások segítenek ebben:

1		Az alakzatok közül egyik szerepel a kódban és a megfelelő helyen található.
2		Az alakzatok közül egyik sem szerepel a kódban.
3		Két alakzat szerepel a kódban, de mind a kettő rossz helyen van.
4		Egy alakzat szerepel a kódban, de rossz helyen található.
5		Egy alakzat szerepel a kódban, de rossz helyen található.

Az alábbiak közül melyik lehet a kincsesláda zárjához tartozó kód?

A)



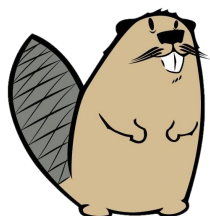
B)



C)



D)



A helyes válasz az B)

Kezdjük azzal, hogy meghatározzuk melyik alakzatok szerepelhetnek a kódban. A második útmutatás alapján el tudjuk távolítani azokat az alakzatokat, amelyek nem lehetnek részei a kódnak:

a fát , a gyémántot  és a házat .

Az utolsó útmutatás alapján az egyik alakzat a kód része, de a rossz helyen található. Mivel a fa  és a ház  nem szerepelhetnek a kódban, tudjuk, hogy a csillag  a kód része, de rossz helyen van. A harmadik útmutatás kizárja azt, hogy a csillag középen legyen. Így megállapíthatjuk, hogy hol van a végleges helye:



Mivel csak egy olyan válaszlehetőség van, ami csillaggal kezdődik, ezért meg is találtuk a kódot.

Azért, hogy meggyőződjünk a válasz helyességéről, keressük meg a hiányzó két alakzatot. Az első útmutatás szerint egy alakzat szerepel a kódban, és a megfelelő helyen található. A ház  és az első helyet (illetve az azon álló szögletes csillagot ) ki lehet zárni, ami alapján tudjuk, hogy a hold  lesz a kód része, és a megfelelő helyen található. Ezt mutatja a következő kép:



A negyedik útmutatás megadja, hogy egy alakzat bár a kód része, de rossz helyen van. A szögletes csillagot  kizárhatjuk. Ezen kívül csak a középső hely szabad még, így a szív  sem lehet a kód része. Ebből következik, hogy a kör  foglalja el a középen lévő helyet.

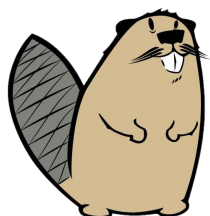


A helyes megfejtést máshogy is meg lehet kapni, azonban minden lehetőség ugyanahhoz a megoldáshoz vezet.

MIÉRT INFORMATIKA?

Ez a feladat megoldható logikusan gondolkodva, például a kizárás módszerének segítségével. A mi esetünkben a második útmutatással kezdtünk, és az ebben szereplő 3 alakzatot kizártuk, ami gyorsan a megoldáshoz vezetett. A 2. útmutatás prioritizálását stratégiának, szabálynak vagy lerövidítésnek is tekinthetjük, ez segített korlátozott tudás birtokában döntést hozni. Az informatikában ezeket a szabályokat heurisztikának hívják, amik programozhatóak és automatizálhatóak.

Minden nap szembe jön velünk sok apró döntés, melyet útmutatások alapján kell meghoznunk. Vagy egy probléma különböző (mellék-)feltételeit kell megértenünk, hogy megoldhassunk egy



feladatot. Ebben a feladatban követtük az útmutatásokat és lépésről lépésre oldottuk meg, hogy kinyíljon a kincsesláda.

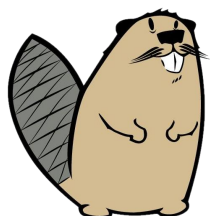
Hogy oldaná meg ezt a problémát egy számítógép? A 8 alakzatot összesen 336 féleképpen lehet a 3 helyre sorba rakni. Egy számítógép mindegyik lehetőséget kipróbálná. Az informatikában ezt teljes, vagy „nyers erő” keresésnek hívjuk. A „nyers erő” keresés (Brute-Force keresés) egy olyan módszer a probléma megoldására, amelynél a számítógép az egész keresési teret átfutja, minden lehetséges kombinációt előállít. Számunkra ez a megoldás nagyon nem tűnhet hatékonynak, mivel nekünk sok időbe telne az összes lehetőség kipróbálása. Egy számítógép viszont gyorsan és hatékonyan meg tud oldani egy ilyen problémát. Az alakzatok sora megfelelhetne egy jelszónak. Egy jelszót úgy kell kiválasztani, hogy a lehető legtöbb különböző karaktert tartalmazza, ezzel egy teljes keresés ésszerű időn belül nem adja ki a megoldást.

A második útmutatással való kezdést és ezzel a megoldási lehetőségek leszűkítését az informatikában visszalépéses keresésnek nevezzük. Elindulunk egy megoldással és feltételezve lépünk előre. Ha ellentmondásba ütközünk, visszalépünk addig, amikor egy döntést hoztunk (feltételeztünk egy továbblépést) és egy másik „irányba” lépünk onnan tovább.

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Heurisztika>

https://hu.wikipedia.org/wiki/Visszal%C3%A9p%C3%A9s_keres%C3%A9s

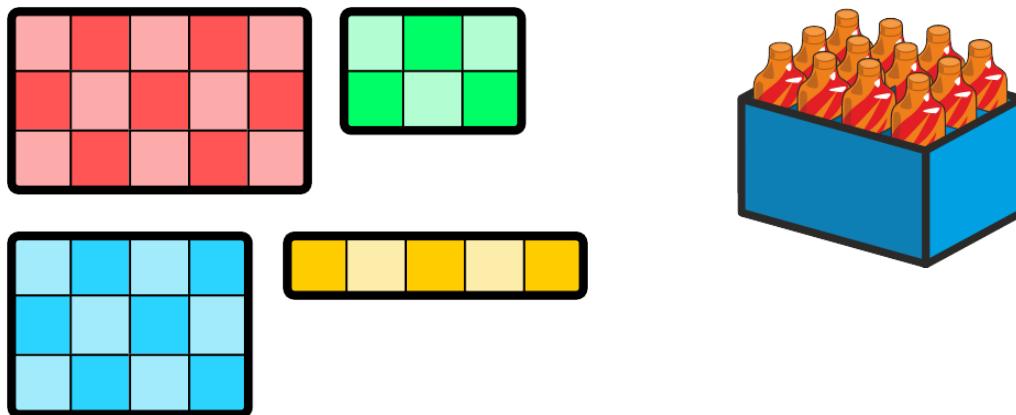


CSOMAGOLÁS (2022-NL-03)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

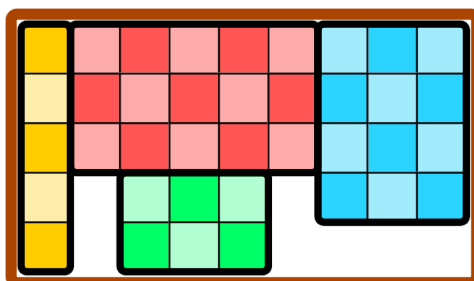
A Castorjuice gyár négyfajta üdítőt gyárt. Mindegyik üdítőfajta palackjait különböző méretű dobozokba csomagolják. Ezeket a dobozokat láthatod a képeken felülről:



A legnagyobb méretű dobozba például 15 palack üdítő fér el.

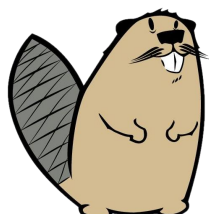
Egy reklámkampány keretén belül a gyár az összes ügyfelének küld egy válogatást, mind a négy fajta üdítőből egy-egy dobozzal. Ezt egy téglalap alapú nagyobb dobozba csomagolják. Az üdítő palackokat tartalmazó dobozok nem helyezhetők egymásra, és nem szeretnék megbontani azokat. Az üres helyeket a nagyobb dobozban majd egy-egy palackkal töltik ki.

Például, ha így helyeznék el a palackokat tartalmazó dobozokat, akkor egy 9x5 helyes nagyobb dobozra és további 7 palackra lenne szükség a csomagoláshoz.



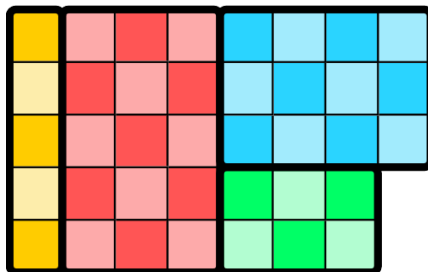
De beférne a válogatás egy kisebb dobozba is kevesebb üres hellyel.

Mennyi az a legkevesebb palack, ami szükséges az üres helyek kitöltéséhez?



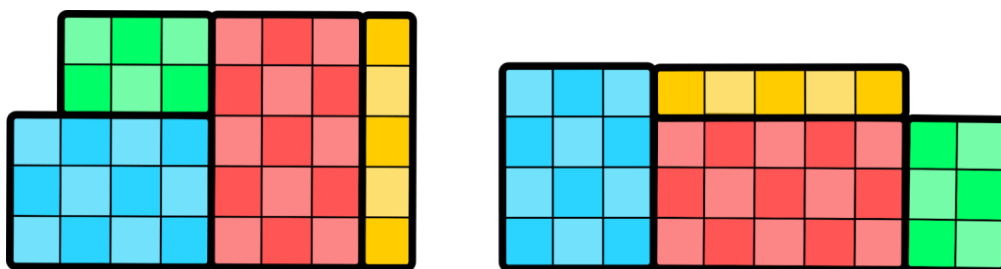
A helyes válasz a 2

Összesen $12 + 15 + 6 + 5 = 38$ palack van, amit el kell helyezni. Egy olyan doboz, amibe üres hely nélkül beleférnek vagy 1×38 , vagy 2×19 méretű lenne (a 2-n és a 19-en kívül nincs más osztója a 38-nak). Azonban a 3×4 -es és az 3×5 -ös dobozok nem férnének be egyikbe sem.



Ha 39 palack számára választunk dobozt (tehát egy üres hellyel pontosan egy plusz palackkal), akkor vagy 1×39 -es 3×13 -as méretű lenne a nagy doboz. Az alábbi méretű dobozok: 3×5 , 3×4 , 3×2 beférnének, de az 1×5 -ös már nem férne be a maradék 2×3 -as helyre.

Egy 40 palacknyi doboz a következő méreteket veheti fel: 1×40 , 2×20 , 4×10 , 5×8 . Az 1×40 -es vagy a 2×20 -as dobozba nem férne be az összes doboz. A másik két dobozba viszont mind a négy doboz belefér:

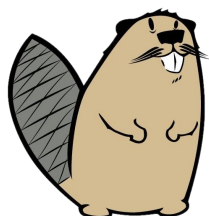


A dobozokat még további elrendezéssel is el lehet helyezni egy 40 palacknyi kartonba. Két üres helynél többet azonban nem lehet spórolni a csomagolásánál.

MIÉRT INFORMATIKA?

Ebben a hódfeladatban a téglalapokat úgy kell elhelyezni, hogy a körülöttük lévő téglalapnak minimális legyen a területe. Ez a probléma az informatikában „négyzetekre bontás” (ang.: rectangle packing) néven ismert, egy az úgynevezett csomagolási problémák közül. Kevés téglalap esetén könnyen megtalálhatjuk az optimális megoldást (ebben az esetben a lehető legkisebb doboz). Nagyobb darabszám esetén szükséges a folyamat optimalizálása; kell egy algoritmus, amit a számítógép végre tud hajtani. Sajnos a négyzetekre bontás úgy, mint a többi csomagolási probléma, NP teljes. Ez azt jelenti, hogy a problémára valószínűleg nincs hatékony algoritmus, ami megtalálja az optimális megoldást. Az informatikában ezért az NP teljes problémákra olyan hatékony algoritmusokat fejlesztenek, ami bár nem garantálja az optimális eredményt, de egy bizonyíthatóan jó megoldást talál.

Például logisztikai cégek esetében: raktározás magas polcokon, az áruk helytakarékos csomagolása vagy az áruk elhelyezése konténerekbe is ilyen feladat. Ezen kívül láthatóan más típusú problémák is leírhatóak csomagolási problémaként. Egy munkafolyamat, amit N munkavállaló M óra alatt tud elvégezni, ábrázolható egy $N \times M$ -es téglalapként. Több folyamatot lehet tehát a lehető legkevesebb emberi erőforrás és idő ráfordításával elvégezni, ha a megfelelő négyzetekre bontjuk, optimalizáljuk a problémát.

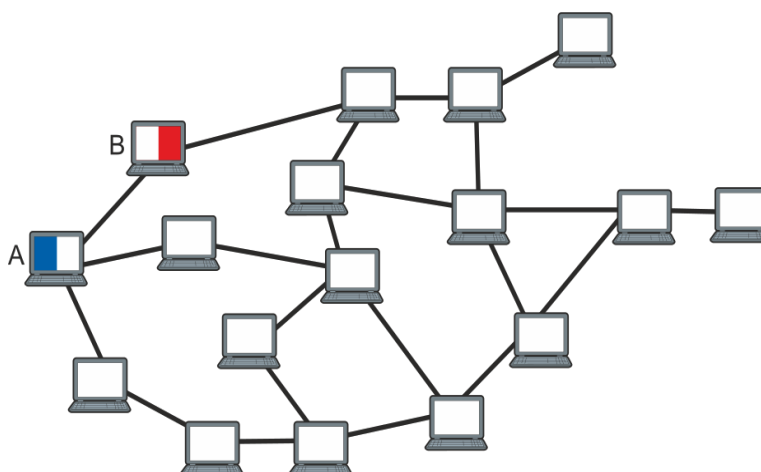


VÍRUS (2022-NZ-01)

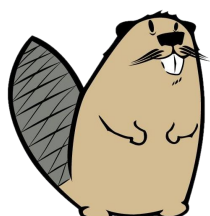
SENIOR – NEHÉZ

Egy nagyobb számítógépes hálózatban két számítógép fertőződött meg vírusokkal: az egyik a **Kék Fenyegetés**  vírussal, a másik pedig a **Piros Zavarás**  vírussal.

A vírusok egy órán belül megfertőzik az összes hozzájuk közvetlenül kapcsolódó számítógépet. A következő órán belül ezek pedig megfertőzhetik az összes hozzájuk közvetlenül kapcsolódó számítógépet stb. Ha egy számítógép mindkét vírussal fertőzött, akkor az összes kapcsolata blokkolva lesz, és a számítógép  leáll, nem is fertőz tovább.

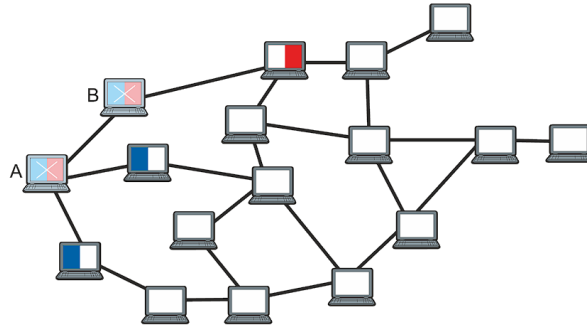


A következő számítógépes hálózaton válaszd ki minden számítógépre, hogy a **Kék Fenyegetés**  vírussal, a **Piros Zavarás**  vírussal fertőződött meg vagy leállt !

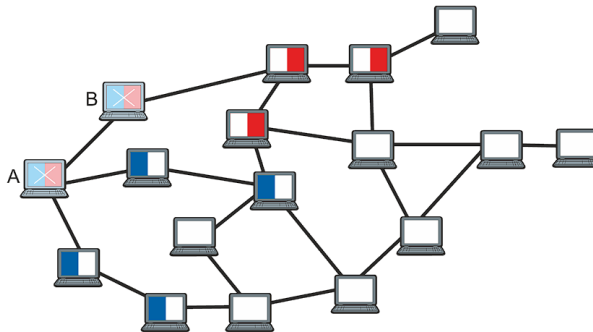


A helyes válasz magyarázattal:

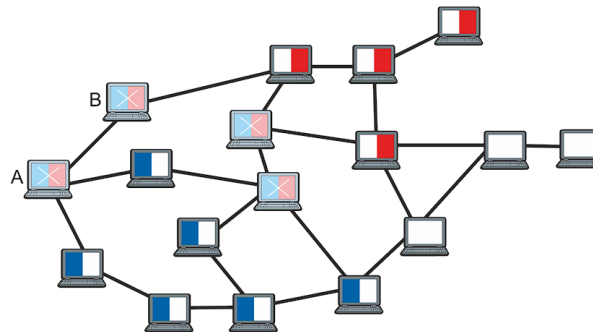
Egy óra elteltével öt fertőzött számítógép van. A két eredetileg fertőzött számítógép megfertőzte egymást, így leálltak, és már nem tudnak más számítógépet megfertőzni:



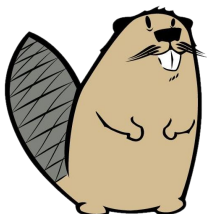
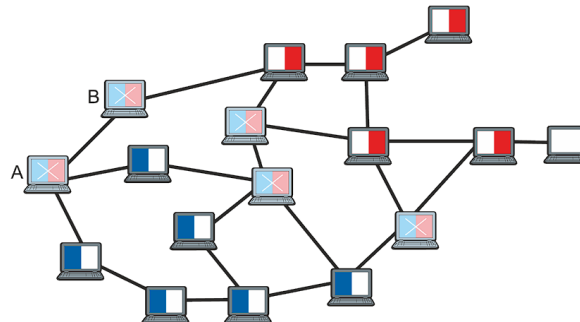
Még egy órával később már négy fertőzött számítógép van:



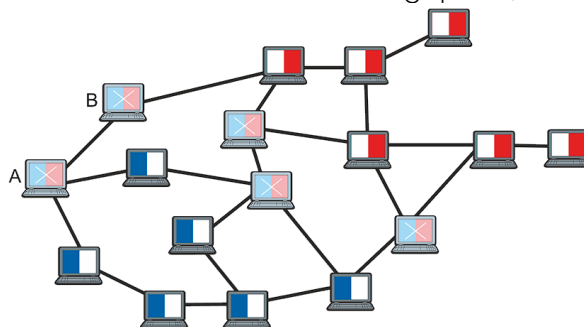
Egy újabb óra elteltével két számítógép kölcsönösen megfertőzte egymást, és így leálltak. Emellett három új számítógép lett Kék Fenyegetés vírussal fertőzött és kettőt pedig a Piros Zavarás vírus ért el:



Még egy órával később egy újabb számítógép leállt és egy megfertőződött:



Az utolsó órában a Piros Zavarás elérte a hálózat utolsó gépét is, tehát a megoldás a következő:



MIÉRT INFORMATIKA?

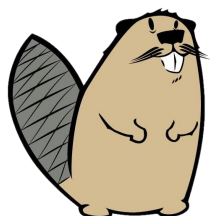
A vírusok és más rosszindulatú szoftverek komoly veszélyt jelentenek a számítógépes hálózatokra. Ezek nemcsak az érintett számítógép teljesítményét befolyásolják, hanem gyakran további károkat is okoznak. Egyes esetekben például a továbbított adatokat beolvassák, és érzékeny információkat, például jelszavakat vagy felhasználói adatokat találnak és továbbítanak kifelé. Van, amikor a vírus titkosítja a fertőzött számítógépen lévő adatokat. Ha a felhasználó ismét hozzá szeretne férni adataihoz, először egy anonim számlára pénzt kell utalnia. Néha a fertőzött számítógépeket távolról irányítják, hogy támadásokat indítsanak más számítógépek ellen.

A számítógépet teljesen letiltó vírust általában a vírus okozója nem ennek szánta, mivel ez megállítja a vírus terjedését is. Egyes vírusokat azonban kifejezetten szabotázsra és kiberhadviselésre fejlesztettek ki. Ennek eredményeként az érintett számítógépek akár véglegesen is megsérülhetnek.

Az aktuális biztonsági frissítések telepítése a vírusok elleni védekezés fontos előfeltétele. A vírusirtó programok javíthatják a védelmet, de egyes operációs rendszerekben már alapvetően megtalálhatók, így nincs szükség további programok telepítésére. Emellett elengedhetetlen az adatok rendszeres biztonsági mentése, és a rendszer szokatlan viselkedésével kapcsolatos éberség. Mindezek mellett fontos a józan óvatosság is, hiszen a vírusfertőzést sokszor kamu levelekkel vagy fertőzött oldalak látogatásával kapjuk meg.

WEBOLDALAK

https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%B3g%C3%A9pes_v%C3%ADrus
<https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%B3g%C3%A9p-h%C3%A1l%C3%B3zat>
<https://hu.wikipedia.org/wiki/V%C3%ADrusirt%C3%B3>

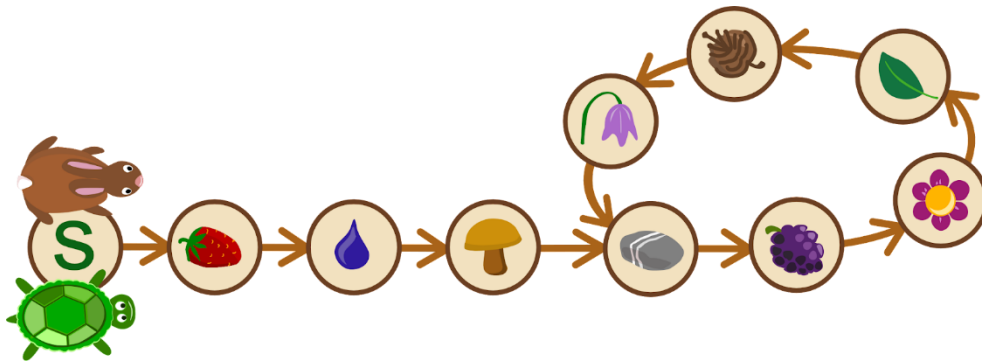


TEKNŐS ÉS NYÚL (2022-PH-03)

KISHÓD – KÖZEPES

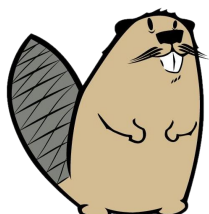
BENJAMIN – KÖNNYŰ

Egy teknős és egy nyúl versenyeznek. Ezt a pályát használják:



Egyszerre indulnak a startmezőről. A nyilakat követve haladnak mezőről mezőre. A teknős egy perc alatt egy mezőt halad. A nyúl egy perc alatt kettő mezőt halad.

Melyik mezőn fognak először találkozni a rajt után?



A helyes válasz a



A teknős és a nyúl először a  mezőn találkoznak. Ennek két ujjunk segítségével könnyen utána számolhatunk.

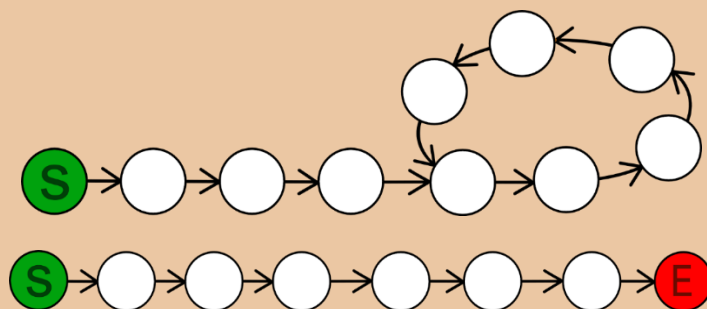
Rajt utáni percek	0	1	2	3	4	5	6	7	8	9	10	11
	S											
	S											



MIÉRT INFORMATIKA?

Ebben a feladatban a verseny egy speciális pályán zajlik. Ez a pálya egyedi mezőkből és a következő mezőre mutató nyilakból áll. Különlegessége, hogy a pálya egy olyan hurokban végződik, ahol a futók vég nélkül futhatnak. A teknős és a nyúl csak azért találkozhatnak ebben a feladatban, mert ez a hat mező egy kört alkot.

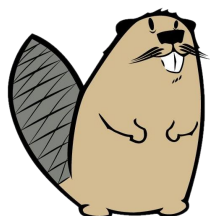
Az informatikában egy olyan pályát, amit a feladatban láttunk, listának nevezzünk. A mezők egymásra mutató körét pedig körnek. A listában minden csomópont legfeljebb egy másik csomópontra mutathat. Vannak kört tartalmazó listák és vannak kör nélküli listák.



Ha egy listában nincs kör, akkor a lista csomópontok lineáris láncából áll. Ekkor kell lennie egy végmezőnek, ahonnan nem indul ki több nyíl. A híres informatikus, Robert W. Floyd (1936-2001) olyan algoritmust tervezett, ami könnyen meg tudja különböztetni, hogy egy listában van-e kör, vagy lineáris láncból áll-e.

WEBOLDAL

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



KREATÍV HÍMZŐPROGRAM (2022-SK-02)

BENJAMIN – KÖZEPES

KADÉT – KÖNNYŰ

Lana készített egy programozható hímzőgépet. A hímzőgép kétféle öltést tud hímezni: ,



Ha ezt az összetett öltést  szeretné, akkor mindkét öltésre  és  szüksége van, és közben az anyagot egy öltéssel hátra kell tolni.

Lana a hímzőgépet a következő három gombbal tudja programozni:

	A hímzőgép  -t híméz.
	A hímzőgép  -t híméz.
	A hímzőgép az anyagot egy öltéssel hátra tolja.

A gombok lenyomásával létrejön egy program, amit a hímzőgép ismételve végrehajt. Például a

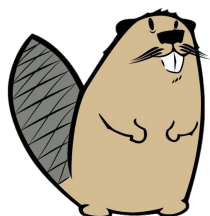
hímzőgép ezzel a programmal      a következő mintát készíti:



Az alábbi programok közül melyiket használta Lana ennek a mintának az elkészítéséhez?



- A)           
- B)       
- C)         
- D)           



A helyes válasz a C)

A helyes válasz megtalálásához fel kell ismernünk az ismétlődő mintát. Ez a minta ismétlődik: . Ez azt jelenti, hogy az első és a második parancsnak -nek kell lennie. Tehát a D) programot máris kizárhatjuk. Ezután egy csillagöltés következik. Ez az és parancsokkal jön létre. A sorrend nem számít, de az anyagot a két öltés között hátra kell tolni: . Ehhez ezek a parancsok szükségesek: vagy . A következő parancs után megint egy csillagöltés jön vagy parancsokkal. Tehát a B) és a D) programokat kizárhattuk.

A helyes program tehát a következő parancsokat tartalmazza: .

A) program a következő mintát készíti:

B) program a következő mintát készíti:

D) program pedig már az elején két csillagot hímez.

MIÉRT INFORMATIKA?

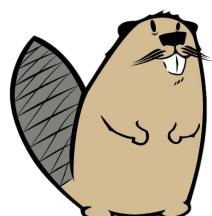
Ebben a feladatban egy ismétlődő minta jön létre egy utasítássoron keresztül. Gyakran az informatikában is a nagy, bonyolult problémákat kisebb problémákra bontják, amik könnyebben megérthetők, megoldhatók, és pl. programozhatók. Ebben a folyamatban kulcsfontosságú a visszatérő minták felismerése a megoldás újrafelhasználása érdekében. Ez megtehető például ismétlések (loop-ok) segítségével.

A hímzőgép által generált program egy programnyelven írt utasításlista. Alapvetően a programozható hímzőgép csak egy robot vagy számítógép, amely utasításokat hajt végre. Ahogy a hímzőgép hímzi a pontos öltéseket, a számítógép a program pontos utasításait hajtja végre. Az utasítások szigorú követése fontos fogalom az informatikában. Az utasítások sorrendje is fontos. Amikor megváltoztatjuk a sorrendet, általában megváltozik a program kimenete is. Esetünkben ez azt jelenti, hogy az utasítások eltérő sorrendje más öltéssorozathoz és így más mintához vezet (kivétele: csillagot hímezünk).

WEBOLDALAK

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

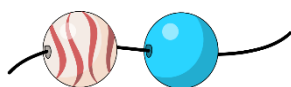
<https://hu.wikipedia.org/wiki/H%C3%ADmz%C3%A9s>



MATRÓZLÁNCOK (2022-SK-03)

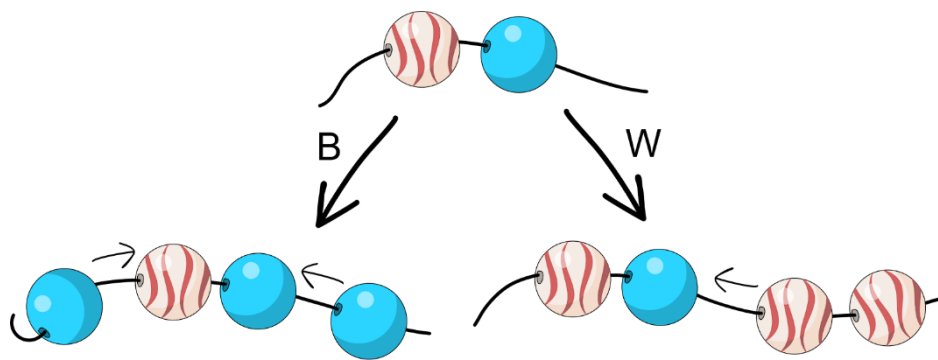
KISHÓD – NEHÉZ
 BENJAMIN – KÖZEPES
 KADÉT – KÖNNYŰ

Mónika matróznyakláncot fűz, ami fehér-piros csíkos és egyszínű kék gyöngyökből áll:
 Mindig egy csíkos és egy kék gyönggyel kezd; ebben a sorrendben:



Utána tudja meghosszabbítani a matróznyakláncot úgy, hogy:

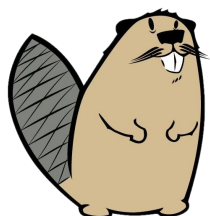
- a zsinór mindkét végéhez egy-egy kék gyöngyöt fűz hozzá (B)
- vagy két csíkos gyöngyöt fűz a zsinór jobb oldali végéhez (W)



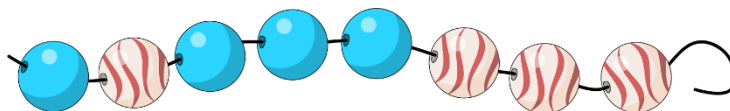
Ezeket többször meg tudja ismételni, hogy egyre hosszabb láncot fűzhessen

A következő matróznyakláncokból melyik nem lehet Mónikáé, ha csak a fenti lépéseket alkalmazza?

- A)
- B)
- C)
- D)



A helyes válasz a D)



A feladatot több különböző módszerrel is meg lehet oldani.

Például minden láncban először megkeresed a kezdő gyöngyöket, és utána B és W műveletek sorát végzed el.

Az A láncnál a kettes és a hármas gyöngyökből kiindulva végrehajtható a B-W-W művelet.

A B láncnál a hármas és a négyes gyöngyök a kiindulási pontok, majd elvégezhetőek a B-B-W műveletek.

A C láncnál a második és a harmadik gyöngyből kiindulva végrehajtható a W-B-W művelet.

Ha a D láncot megfigyeled, akkor a második és a harmadik gyönggyel kell kezdeni. Ezután egyszer el lehet végezni a B műveletet, de utána nincs olyan művelet, amivel a lánc többi részét megkapnánk.

Ez a módszer nem működik jól, ha a lánc túl hosszú és nagyon sok lehetséges gyöngy van, amiből ki lehet indulni. Ebben az esetben egy dekonstruktív módszer célravezetőbb lehet. Ennél a megoldásnál eltávolítod ismételten a gyöngyöket, úgy, hogy a B műveletet vagy a W műveletet fordítva hajtod végre, addig, ameddig két gyöngy marad.

Egy harmadik stratégiához a paritás lehet segítségünkre. A matróznyakláncok készítési szabályai alapján mindig páratlan számú egyszínű kék és páratlan számú piros fehér csíkos gyöngyöt kapunk („páratlan paritás”). Látod, hogy ez miért van így?

A D láncban páros számú gyöngy található mind a két fajtából, ezért nem lehet Mónika matróznyaklánc.

MIÉRT INFORMATIKA?

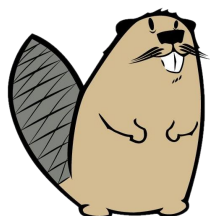
Ebben a feladatban a gyöngyöket csak a lánc végeire tudod felfűzni. Nem tudsz középre beilleszteni egy gyöngyöt. Ki sem lehet venni gyöngyöt a lánc közepéről, anélkül, hogy le nem húznánk a szélén lévő gyöngyöket.

Ezt a mentési módszert, aminél elemeket a két végén lehet hozzáadni és eltávolítani, de középen nem, az informatikában „double-ended queue-nak” vagy kettős végű sornak nevezzük.

A kettős végű sort fel lehet használni a böngészési előzmények mentésére, nyomtatás indításának tervezésére és matematikai kifejezések ellenőrzésére. Például annak ellenőrzése, hogy a zárójelek összhangban vannak-e, hasonló módon történik, mint egy láncnál az ellenőrzés, hogy Mónika matróznyaklánc lehet-e

WEBOLDAL

https://wikihuhu.top/wiki/Double-ended_queue



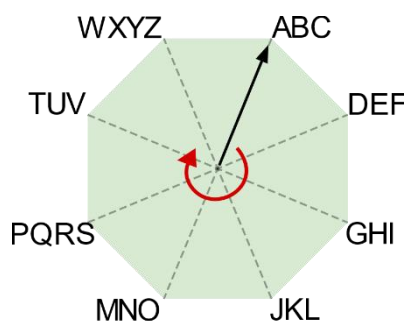
TITKOSÍRÁS 8 (2022-SK-04)

BENJAMIN – NEHÉZ

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

Az alábbi kerékkel titkos üzeneteket lehet alkotni:



A mutató az elején az „ABC” hely felé mutat.

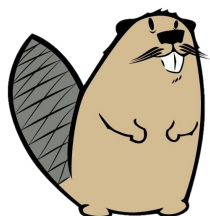
A kerék egy üzenetben minden betűt egyenként titkosít. Ehhez két számjegy szükséges:

- Az első számjegy megmutatja, hogy hány helyet fordul a mutató, az óramutató járásával megegyezően az aktuális helyről. Ekkor a mutató már arra a helyre mutat, ahol a titkosítani kívánt betű található.
- A második számjegy megadja, hogy hányadik betűt választjuk a betűsorban.

Az «ANYA» szót például így kell titkosítani: 81-42-33-11

Mit jelent az 12-44-11-43 számsor?

- A) ESIK
- B) ESTI
- C) ESZES
- D) ENEO



A helyes válasz a B)

A 12 azt jelenti, hogy a mutató az «ABC» helyről a «DEF» helyig fordul el (az első számjegy az 1), és a második betűt «E» választjuk ki (második számjegy a 2).

A 44 azt jelenti, hogy a mutató a «DEF» helyről a «PQRS» helyig fordul el (az első számjegy a 4), és a negyedik betűt «S» választjuk ki (a második számjegy a 4).

A 11 azt jelenti, hogy a mutató a «PQRS» helyről elfordul a «TUV» helyre (az első számjegy az 1), és az első betűt «T» választjuk ki (a második számjegy az 1).

A 43 azt jelenti, hogy a mutató a «TUV» helytől elfordul a «GHI» helyig (az első számjegy a 4), és harmadik betűt «I» választjuk ki (a második számjegy a 3).

Így a helyes megoldás a B) válasz: «ESTI».

MIÉRT INFORMATIKA?

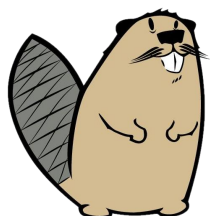
Már több ezer éve próbálnak az emberek információkat titkosítani úgy, hogy a címzett meg tudja fejteni, de más ne. Ami a rúdra tekert papírcsíkokkal (szküta) kezdődött, fejlődött tovább az eltoláson alapuló (pl. Caesar-kód) és a polialfabetikus kódoláson pl. a „Vigenére-kód”) keresztül egészen a modern nyilvános kulcsú rejtjelezésig (pl. a „GnuPG”).

Ebben a feladatban szereplő titkosítási eljárás egy polialfabetikus kódolási eljárás. Mivel ugyanazok a betűk nem feltétlenül ugyanazzal a jellel vannak kódolva: Az „A” betűt (az „ANYA” szóban) például az elején a 81 jelölte, de a végén 11-ként lett kódolva. Alapvetően ezek a titkosítási eljárások manapság számítógépek segítségével mind gyorsan és könnyen megfejthetőek.

Ebben az esetben a megfejtés könnyen átgondolható. Csupán egy kulcs van a szöveg titkosítására. Ha a mutató startpozíciója nem az ABC-től, hanem egy másik bloktól indulna, akkor is csak nyolc különböző kulcs lenne... ezért még a Caesar-kód – ami több mint 2000 éves – is „biztonságosabb”. Noha lehet érvelni amellett, hogy nem a kulcs titkos, hanem a kódolási eljárás. Ezzel szemben áll a Kreckhoffs-elv, amit Auguste Kerckhoffs (1835-1903) 1883-ban fogalmazott meg, és manapság is érvényes. Ez egyértelművé teszi, hogy a rejtjelezési rendszernek nem a kódolási eljárás titokban tartásán kell alapulnia, mivel ez könnyedén eljuthat bárkihez.

WEBOLDAL

<https://hu.wikipedia.org/wiki/Kriptogr%C3%A1fia>

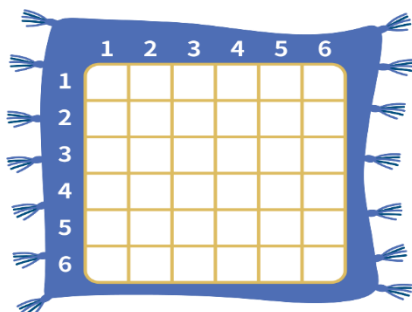


SZŐNYEGMINTA (2022-TR-02)

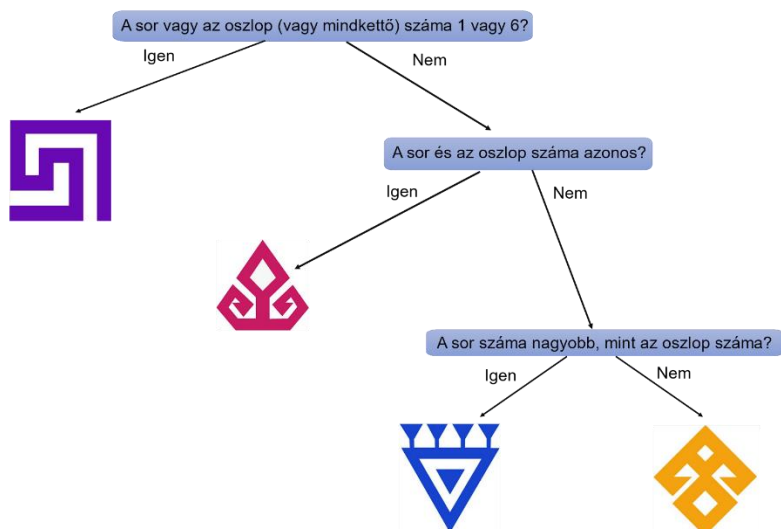
BENJAMIN – NEHÉZ

KADÉT – KÖZEPES

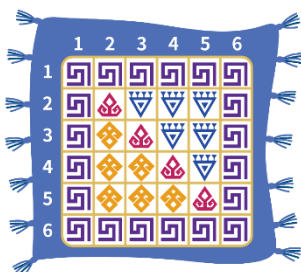
Hale, a török művész, szőnyegmintákat tervez egy hat sorból és hat oszlopból álló négyzetrácsra:



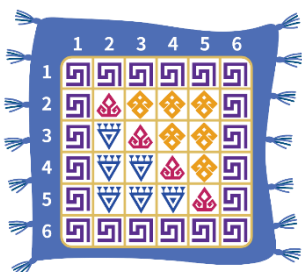
A szőnyegszövőknek készít egy leírást. A négyzetrács minden négyzetébe kell egy-egy szimbólumot helyezni az alábbi ábra alapján:



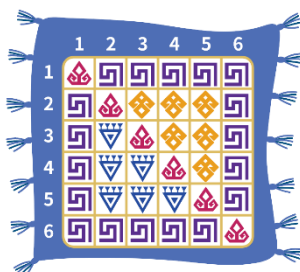
Hogy néz ki ezek alapján a szőnyeg?



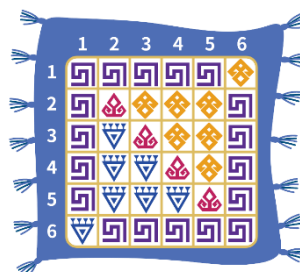
A)



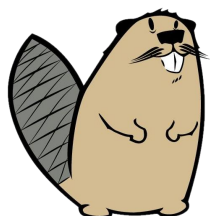
B)



C)



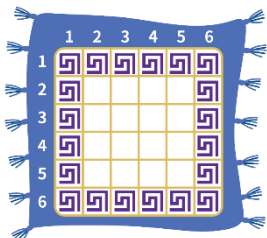
D)



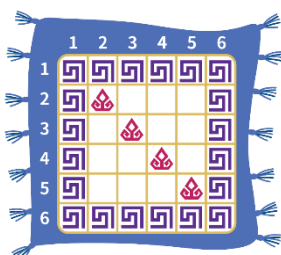
A helyes válasz a B)

A helyes válasz a B

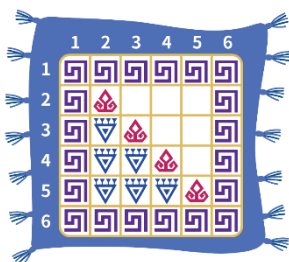
Az első kérdésre a szélén lévő négyzetek esetében mindig „Igen” a válasz. Tehát a szőnyeg szélére a lila csigavonal szimbólum kerül:



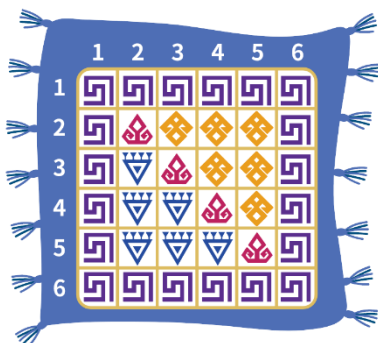
A második kérdésre az átlóban lévő négyzetek esetében lesz „igen” a válasz, így oda a piros szimbólum kerül



Az átló alatti négyzeteknél a harmadik kérdés esetén a sorok száma nagyobb, mint az oszlopok száma, így a kék szimbólum kerül oda.

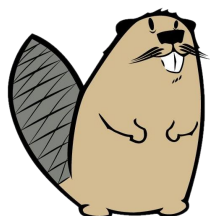


Az átló feletti négyzeteknél a harmadik kérdés esetén a sorok száma kisebb, mint az oszlopok száma, így a sárga szimbólum kerül oda.



MIÉRT INFORMATIKA?

A képet, amelyet a művész útmutatóként készített, az informatikában döntési fának is hívják.

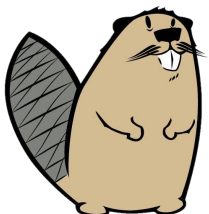


Mint egy igazi fa, ez is ágakból, elágazásokból áll – minden elágazásnál egy eldöntendő kérdés szerepel, melyre igen vagy nem a válasz. Ha fentről lefele „végigmegyünk” a fán, a kérdések megválaszolásával követjük az utat, egy összetett döntést hozunk. Bármikor, bárki megy végig a döntési fán, mindig ugyanazt az eredményt kapja. Akár egy számítógéppel is végrehajthatjuk a lépéseket, ha le tudjuk fordítani a nyelvére valamilyen módon.

Az ilyen egyértelmű elágazások algoritmikus elemek, melyek a programkészítés alapelemei is.

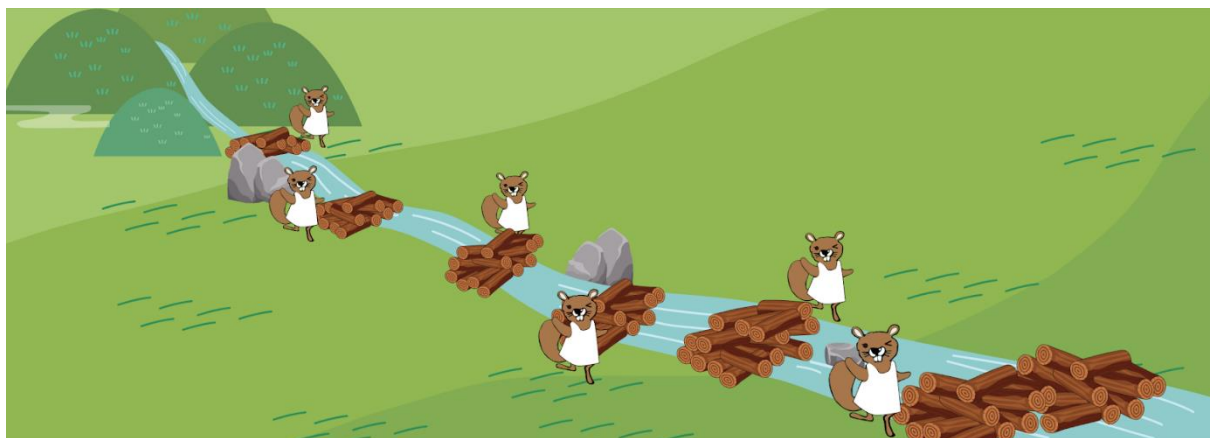
WEBOLDAL

https://hu.wikipedia.org/wiki/D%C3%B6nt%C3%A9si_fa



HÓDGÁT (2022-TW-06)

BENJAMIN – NEHÉZ



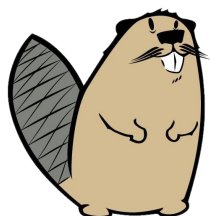
Hat hód (A, B, C, D, E, és F) egyenként épített magának egy saját gátat a Hód-patakon. Egy napon vihar támadt, a patak néhány rönköt kimosott a gátakból, és így a víz lesodorta azokat. Szerencsére korábban a hódok minden rönköt megjelöltek a nevük betűjével.

A vihar után összegyűltek a hódok, hogy átadják egymásnak a rönköket, és visszakapják a sajátjukat, ahogy a kép mutatja:



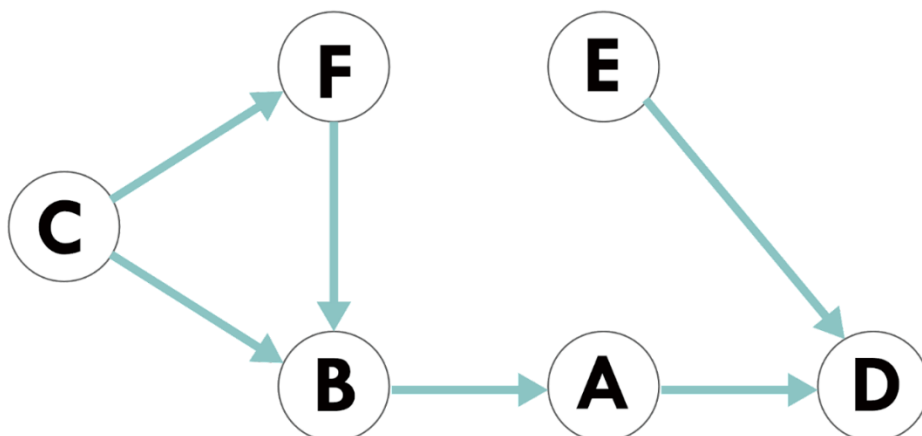
A rönkök alapján, amiket a hódok hoztak, mi lehet a gátak sorrendje, ha a folyó a kép bal felső sarka felől folyik a jobb alsó felé?

- A) $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$
- B) $C \rightarrow B \rightarrow F \rightarrow A \rightarrow D \rightarrow E$
- C) $C \rightarrow F \rightarrow B \rightarrow D \rightarrow A \rightarrow E$
- D) $E \rightarrow C \rightarrow F \rightarrow B \rightarrow A \rightarrow D$



A helyes válasz a D) $E \rightarrow C \rightarrow F \rightarrow B \rightarrow A \rightarrow D$

Mivel a rönkök fentről lefelé sodródtak a folyón, ha az A hód megtalálja B hód rönkjeit, akkor arra következtethetünk, hogy B hód feljebb építkezett az A hódnál. Segít meghatározni mindegyik hód helyét, ha a két hód közötti relatív helyet nyilakkal jelöljük, ahogy alább látható:



Ekkor a C-re és az E-re nem mutat egy nyíl sem, tehát a C és az E hód lehetne legfelül.

A B-re mutat F is, illetve a további nyilakat figyelembe véve így C-F-B-A-D biztos sorrend. Az E bárhova bekerülhet ezen a soron belül a D elé.

Emiatt a megoldásaink lehetnek: E-C-F-B-A-D; C-E-F-B-A-D; C-F-E-B-A-D; C-F-B-E-A-D vagy C-F-B-A-E-D

(Ezért az A) válasz nem helyes. A B) válasz is rossz, mert az F-nek a B felett kellene lennie a fent található gráf alapján, amiben a nyíl az F-ből a B-be mutat. A C) válasz is rossz, mivel a D-nek az E) felett kellene lennie a fenti gráf alapján, amiben a nyíl a C-ből a B-be, majd onnan az A-ba, majd a D-be mutat. Csak a D) válasz felel meg a fenti gráfnak, ezért a D) opció a helyes.)

MIÉRT INFORMATIKA?

Az irányított körmentes gráf csúcsokból és irányított élekből áll, kör nélkül. Ebben a feladatban a hódok szimbolizálják a csúcsokat. Az irány, amerre a rönkök sodródtak, az irányított él. Topológiai rendezésnek hívjuk a hódok helyes sorba rendezését a folyó folyási iránya szerint, vagyis amikor elhelyezzük a csúcsokat az él irányja alapján.

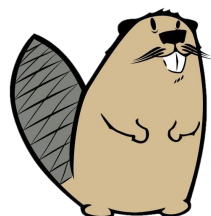
Sokféleképpen lehet úgy sorba rendezni, hogy megfeleljen a topologikus sorrendnek. Az egyik módszer Kahn algoritmusának alkalmazása. Először ki kell választani egy S csúcsot, aminek nincsen bemenő éle. Ezután ki kell törölni az S csúcsot és minden olyan élt, ami ebből indul ki. Folytasd az előbbi lépéseket addig, amíg a gráf üres nem lesz.

Az informatikában a topologikus sorrendet használják az ütemezés feldolgozására, adatsorosításra, és az alkatrész összekapcsolására a szerkesztőkben.

WEBOLDALAK

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

https://hu.wikipedia.org/wiki/Ir%C3%A1ny%C3%ADtott_k%C3%B6rmentes_gr%C3%A1f



RENDEZÉS (2022-UA-03B)

SENIOR – NEHÉZ

Júnónak 11 kártyája van, amiken 1-től 9-ig számok, illetve az A és a B betű szerepel.

Egy asztalra lerakja a kártyákat egymás mellé az $a \rightarrow b$ szabályt követve. Ez a szabály azt jelenti, hogy az a kártyának valahol a b kártya előtt (balra) kell elhelyezkednie.

A számok nagyságrendi sorrendje a szokásos és $9 < A < B$.

Mindemellett Júnó lexikografikusan a legkisebb lehetséges, asztalra letett kártyasorozatot szeretné megtalálni.

A lexikografikus elrendezésben az a kisebb a két kártyasorozatból, amelyiknél balról nézve az első nem azonos kártyán álló szám, illetve betű a kisebb.

Például a következő két kártyasorozat esetében

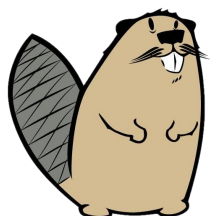
5 6 5 7 9 B

5 6 5 8 0 0

az első kártyasorozat lexikografikusan kisebb, mivel az első három kártya mind a kettőnél ugyanaz, és az elsőnél a negyedik a 7, míg a másikonál a 8 ($7 < 8$).

A 11 kártyából a szabályok alapján több kártyasorozatot is létrehozhat. Ezek közül melyik a lexikografikusan legkisebb sorozat? (Csak a legkisebb sorozatot írd le megoldásként)

1→2	1→6	2→3	6→5
3→5	3→4	5→7	7→8
7→9	1→B	B→A	A→7



A helyes válasz: 1 2 3 4 6 5 B A 7 8 9

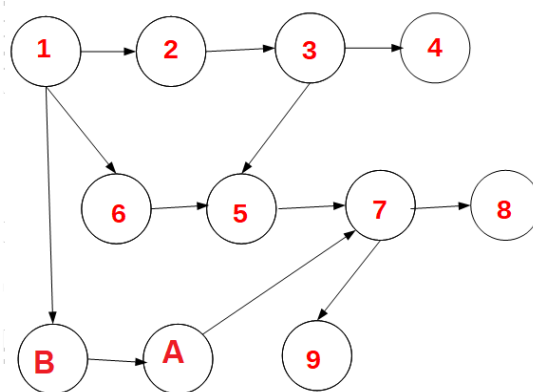
Nézzük meg a megoldásokat - és a lehetséges folytatásokat:

A 1 6 2 3 4 5 B A 7 8 9 szintén kielégíti a feltételeket, de az eredményül kapott sorrend lexikografikusan nagyobb. Hogy találhatjuk meg a megoldást?

Az egyik lehetséges megoldás, ha elkezdjük lerakosgatni a kártyákat a szabályok alapján: letesszük az 1, 2 kártyákat. A következő $1 \rightarrow *$ szabály alapján úgy helyezzük el a jobb oldalon állót (*), hogy a már lent lévőtől a sorrendje szerint jobbra vagy balra kerüljön a szabály jobboldalán álló kártya: [1, 2, 6, B]

Most azokat a szabályokat vesszük, ahol a bal oldalon szereplő már szerepel a kártyasorozatban, és erre alkalmazzuk a „beszúrásokat”: [1,2,3,6,B] $\rightarrow$ [1,2,3,6,5,B] (ezután a $3 \rightarrow 4$ szabálynál is pl. a legjobboldalibb megfelelő helyre tesszük a 4-et: azaz nála kisebbet ne előzzön meg és megfeleljen a szabálynak $\rightarrow$ [1,2,3,4,6,5,B]...

Másik megoldás, ha készítünk egy rajzot (gráfot), ahol körökkel jelöljük a kártyákat és a nyilakkal az $a \rightarrow b$ szabályokat.



Ezután készíthetünk egy listát az összes kártyáról és a hozzájuk tartozó bemenő élekről (azaz rájuk mutató nyilakról):

1: –	5: 3, 6	9: 7
2: 1	6: 1	A: B
3: 2	7: 5, A	B: 1
4: 3	8: 7	

(Az 1. kártyára nem mutat egy nyíl sem, és az 5. kártyára a 3. és a 6. kártyából is mutat nyíl.)

A lexikografikusan legkisebb kártyasorozat az alábbiak alapján található meg:

- Minden egyes lépésben a kártyák közül, amelyekre már nem mutat nyíl, kiválasztjuk a legkisebbet.
- Ez a kártyát és a kimenő nyilait eltávolítjuk.

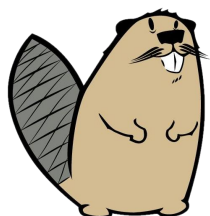
1) Először válasszuk ki az 1. kártyát. Az aktuális kártyasorozat: [1]. A listából pedig mindenhol eltávolítjuk az 1-est (pirossal jelöljük):

1: –	5: 3, 6	9: 7
2: –	6: –	A: B
3: 2	7: 5, A	B: –
4: 3	8: 7	

2) Most már alábbi kártyákra nem mutat nyíl: 2, 6, B.

Válasszuk ki a legkisebbet, azaz a 2-et.

Az aktuális kártyasorozat: [1,2]. A listából pedig mindenhol eltávolítjuk a 2-est (pirossal jelöljük):



1: =

5: 3, 6

9: 7

2: –

6: –

A: B

3: –

7: 5, A

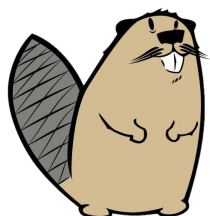
B: –

4: 3

8: 7

3) A következő lépésekben ezt ismételgetjük

Nem mutat rá nyíl	kártyasorozat	lista
3, 6, B	[1, 2, 3]	4: - 5: -, 6 6: - 7: 5, A 8: 7 9: 7 A: B B: -
4, 6, B	[1, 2, 3, 4]	5: 6 6: - 7: 5, A 8: 7 9: 7 A: B B: -
6, B	[1, 2, 3, 4, 6]	5: - 7: 5, A 8: 7 9: 7 A: B B: -
5, B	[1, 2, 3, 4, 6, 5]	7: -, A 8: 7 9: 7 A: B B: -
B	[1, 2, 3, 4, 6, 5, B]	7: A 8: 7



		9: 7 A: -
A	[1, 2, 3, 4, 6, 5, B, A]	7: - 8: 7 9: 7
7	[1, 2, 3, 4, 6, 5, B, A, 7]	8: - 9: -
8, 9	[1, 2, 3, 4, 6, 5, B, A, 7, 8]	9: -
9	[1, 2, 3, 4, 6, 5, B, A, 7, 8, 9]	

MIÉRT INFORMATIKA?

Rendezésnek hívjuk az olyan algoritmusokat, amelyek elemeket sorakoztatnak fel egy meghatározott sorrendben. Ez egy klasszikus területe az algoritmuselméletnek.

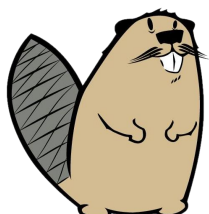
A legegyszerűbb megoldás megkeresni és ellenőrizni az összes lehetséges sorbarendezést (ezt hívják Brute-force keresésnek). Ez hatékony olyan sorozatoknál, amelyek kevés elemből állnak. Sok kártya és feltétel megadása esetén ez a módszer nagyon nehézkesé válhat és többszöri ellenőrzést követel meg ugyanazoknál a feltételeknél. Ugyanakkor óvatossá kell lenni, nehogy elkerülje a figyelmet a lexikografikus legkisebb megoldás.

Ezért ez a probléma sokkal hatékonyabban megoldható a topologikus sorrend elkészítésével – amit a második megoldásunk ír le lépésről lépésre kiegészítve egyből a számunkra legjobb megoldás megkeresésével (azaz, hogy mindig a legkisebbet válogattuk be a bal oldali oszlopból, azokból a kártyákból, amelyekre nem mutatott nyíl).

WEBOLDALAK

https://wikihuhu.top/wiki/Brute-force_search

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



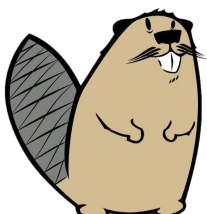
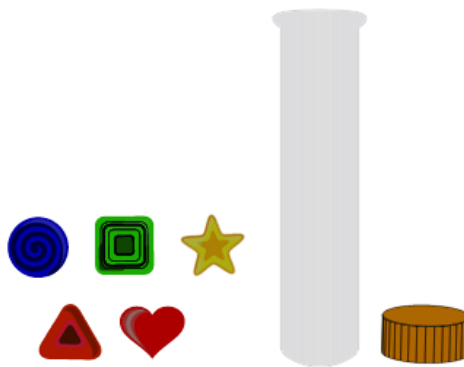
KEDVENC CUKORKA (2022-IT-01B)

KISHÓD – KÖNNYŰ

Anna minden nap 5 különböző cukorkát visz magával az iskolába. Olyan sorrendben eszi meg ezeket, ahogy a cukorkatartó tetejéről ki tudja venni.

Ma a következő sorrendben szeretné megenni a cukorkákat: , , , , 

Hogy töltsse meg a cukorkatartót, ha mindig csak a következő cukorkát szeretné kivenni és a kívánt sorrendben megenni?



A helyes válasz a következő:



Ahhoz, hogy Anna a helyes sorrendben vehesse ki a cukorkákat, észre kell vennie, hogy a legelsőnek a betett cukorkát veheti ki majd utoljára. A legutolsó betett pedig legelsőként.

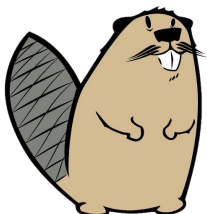
MIÉRT INFORMATIKA?

Az informatikában is gyakran fontos, hogyan gondolkodunk a sorrendről. Az ebben a hódfeladatban használt sorrendet, és az így használt tárolási módot, veremnek nevezzük. A verem olyan tároló struktúrák, amelyekhez bizonyos sorrendben adhatunk hozzá és távolítunk el objektumokat: az „utolsó elsőként” (angolul: last in, first out, röviden LIFO) elvének megfelelően működnek. Ez azt jelenti, hogy az utoljára hozzáadott objektumokat veszik ki először.

Műveletek, utasítások feldolgozásakor is gyakran használják.

WEBOLDALAK

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

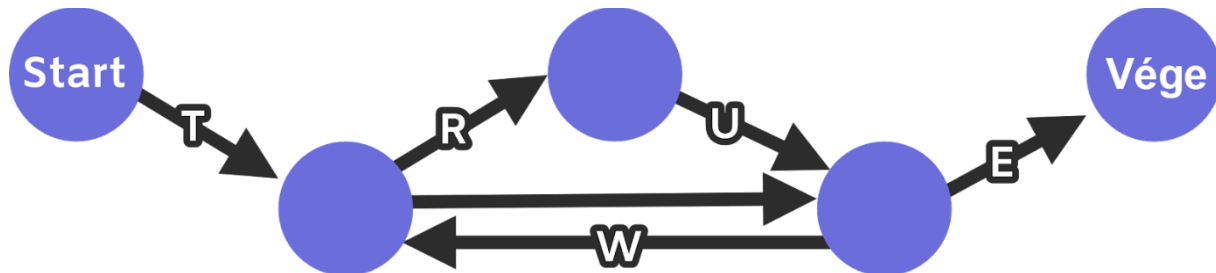


SZAVAK (2022-US-06)

JUNIOR – NEHÉZ

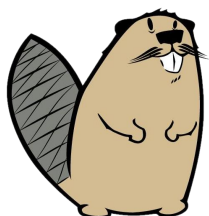
SENIOR – KÖZEPES

Brigi véletlenszerűen szavakat alkot egy speciális ábra segítségével. Mindig a „Start” körrel kezd és követi a nyilakat, amíg a „Vége” körhöz ér. Minden betűt leír, ami mellett elhalad. A betű nélkül álló nyílnál nem ír le semmit.



A fenti ábra alapján például a *twwe* vagy a *true* „szavakat” tudja megalkotni.

Hány különböző 8 betűs szót tud alkotni?



A helyes válasz a 9

Látható, hogy mindig használnia kell a „T”-t az elején és az „E”-t a végén. Szóval hat betűs szavakat keresünk az „R”, az „U” és a „W” betűkkel, melyek R-rel vagy W-vel kezdődnek (az üres nyíl miatt a vissza mutató W) és U-ra vagy W-re végződnek.

Lebonthatjuk a problémát egy kisebb problémára: hány különböző módon lehet meghatározott hosszúságú szavakat alkotni:

- 1 darab 1 betűs: (W, kövesd az üres sort, kövesd a W-t, kövesd az üres sort)
- 2 darab 2 betűs: (WW és RU; kövesd az R-t, kövesd az U-t)
- 3 darab 3 betűs: (WWW; WRU; RUW)
- 4 darab 4 betűs: (WWWW; WWRU; RUWW; WRUW)
- 6 darab 5 betűs: (WWWWW; WWWRU; WWRUW; WRUWW; RUWWW; RUWRU)
- 9 darab 6 betűs: (WWWWWW, WWWWRU, WWWRUW, WWRUWW, WRUWWW, RUWWW, WRUWRU, RUWRUW, RUWRU)

De ezek közül csak azok jönnek szóba, amelyek abba a körbe végződnek, ahonnan az „E”-t tartalmazó, „Vége” körre mutató nyíl kiindul.

MIÉRT INFORMATIKA?

A számítógép nagyon jó abban, hogy az összes lehetőségen végigmenjen egy keresés során.

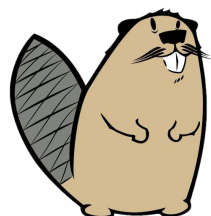
Ebben az esetben tudatosan törekszünk arra, hogy leírjuk az összes szót, de észben kell tartanunk, hogy nem szabad kétszer számolni a szavakat.

Automatának hívjuk az ilyen típusú gráfokat, ahol egy kezdő pontból (állapotból) egy végpontba (állapotba) megyünk át és a nyilakat, mint átmeneteket követjük. Erre példa, amikor betűk sorát („szavakat”) generálunk bizonyos megkötéseket figyelembe véve. Az ilyen típusú rendszereket használják némely esetben jelszavak generálására, pl. rendszámok létrehozására, jelszavak generálására. Ilyenkor a karakterek egyedi sorrendje szükséges, de mindenképpen követnie kell a meghatározott formátumot.

Eggyel nehezebb feladat egy szóról megállapítani, hogy követi-e a szabályokat, azaz előállítható-e egy adott automatával.

WEBOLDAL

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

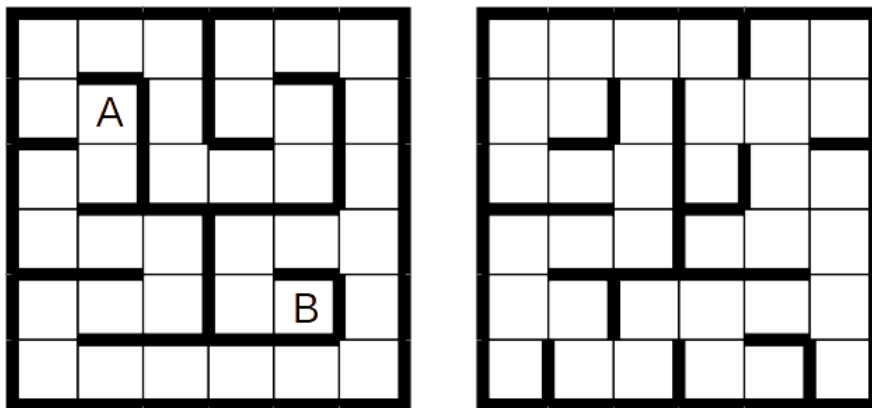


VARÁZSLÓISKOLA (2022-UZ-03)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

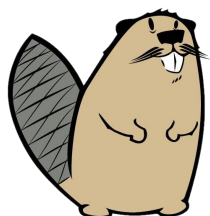
A varázslóiskolának két szintje van. A szintek pontosan egymás felett helyezkednek el. Mindkét szint mezőkre van osztva és bizonyos mezők között falak vannak.



A varázslótanonc Ron 1 másodperc alatt ér át egyik mezőről a másikra (átlósan nem) ugyanazon a szinten. Sajnos azonban Ron elfelejtette a varázsigét, hogyan juthat át a falakon. De arra emlékszik, hogy tud egyik szintről a másikra ugrani (egy mező alatti vagy feletti mezőre). Ehhez 5 másodpercre van szüksége.

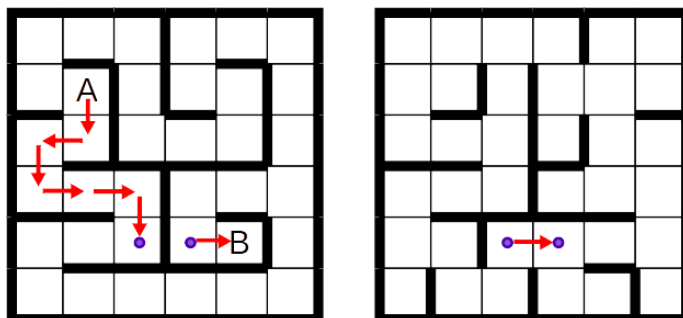
Ron az „A” mezőről a „B” mezőre szeretne eljutni.

Legkevesebb hány másodperc alatt tudja ezt megtenni?



A helyes válasz a 18 másodperc

18 másodperc alatt így jut el Ron „A”-ból „B”-be:



De tényleg ez a leggyorsabb út?

A „legrövidebb időt”, ami alatt Ron az „A” mezőről a „B” mezőre a következőképpen számolhatjuk ki:

Az A mező esetében a legrövidebb idő nyilvánvalóan 0 másodperc. Ezután lépésről lépésre menjünk tovább: Azon mezők közül, amelyeknél már a legrövidebb idő van megadva, válasszuk ki a legalacsonyabb értékűt. (Tehát a legelején az A négyzetet választjuk.) Ebből a kiválasztott négyzetből megnézzük az összes lehetséges következő mezőt, és átgondoljuk, hogyan juthatunk el a kiválasztott négyzetből a lehető leggyorsabban; a számított időket a következő mezőkbe pedig beírjuk. Előfordulhat, hogy egy korábban beírt idő javításra kerül. A kiválasztott mezőt ezután már nem lehet figyelembe venni; ezért a következő lépésekben már nem választható ki.

Itt vannak az ezzel a módszerrel kiszámított legrövidebb idők, az A mezőtől kezdve:

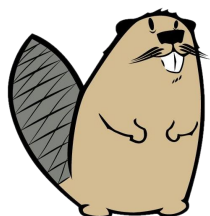
2	3	4	11	12	13
1	0	5	10	9	14
2	1	6	7	8	15
3	4	5	18	17	16
8	7	6	17	18	15
9	10	11	12	13	14

7	6	7	8	11	12
6	5	8	9	10	11
7	6	7	10	11	12
8	9	8	13	12	13
9	10	11	12	13	14
10	11	12	13	14	15

Tehát Ronnak legalább 18 másodpercre van szüksége, hogy az A mezőből a B mezőbe jusson. 6 másodperc (A válasz) lenne a legrövidebb út időtartama, ha nem lennének falak a mezők között. Ha Ron így is oda-vissza mozog az emeletek között, az 16 másodperc lenne (B válasz). Ha csak egy szint lenne az A és B mezőkkel, akkor 20 másodperc (D válasz) lenne a legrövidebb idő az A-ból B-be vezető úthoz.

MIÉRT INFORMATIKA?

A leggyorsabb vagy legrövidebb útvonalakat elég gyakran kell kiszámítani; nyilvánvaló példa az útvonaltervezés a modern térképalkalmazásokban is. A probléma nagymértékben leegyszerűsödik, ha az utak a szomszédos pontok közötti véges lépésekből állnak, és ismert, hogy ezek mennyibe kerülnek: idő, pénz, energiafogyasztás tekintetében – ami a probléma szempontjából fontos.

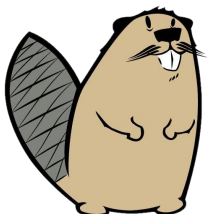


Ebben az esetben a mezők, lépések és a lépések költségei gráfként felrajzolhatóak, amelyben a lépések útvonalakká alakíthatóak. Az informatikában számos algoritmus ismert gráfokra, amelyek segítségével hatékonyan lehet kiszámítani a legrövidebb utakat. Az egyiket Edsger Dijkstra informatikus találta fel; ezt a Dijkstra-algoritmust használtuk fent a helyes válasz magyarázatához.

A legrövidebb utak fontos szerepet játszanak a számítógépek áramköreinek tervezésében is. A kapcsolási pontokat a lehető legalacsonyabb költséggel kell összekötni. A modern áramkörök többszintűek, és a két szint közötti huzalozás drágább, mint az azonos szinten lévő (egyébként összehasonlítható) vezetékezés – hasonlóan az emeletek közötti mozgáshoz ebben a hódfeladatban.

WEBOLDAL

<https://hu.wikipedia.org/wiki/Dijkstra-algoritmus>



HATSZÖGFESTÉS (2022-VN-05A)

KISHÓD – KÖZEPES

BENJAMIN – KÖNNYŰ

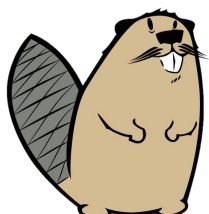
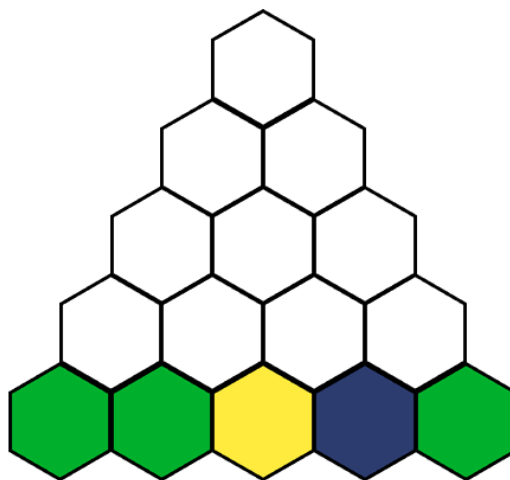
Samu fehér hatszögeket rak egymás mellé, melyeket ezután három különböző színnel fest be az alábbi szabályt követve:

Amikor három hatszög pontosan a képen látható módon van együtt (kettő alul és egy felső középén), akkor vagy mindháromnak azonos színűnek kell lennie, vagy mindháromnak különböző színűnek kell lennie.

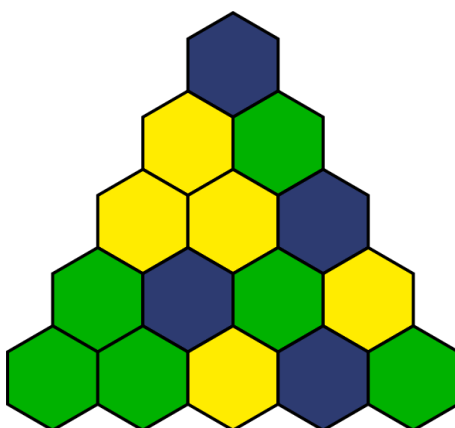
Samu sok hatszöget lerakott, és néhányat már ki is festett.



Fesd ki az összes többi hatszöget a szabályt követve.



A helyes megoldás így néz ki:



Amint felfestünk két hatszöget, amelyek egymás mellett vannak, a felettük közén található hatszög színe egyértelműen meghatározható: ha azonos színűek, akkor a fenti hatszög is ezt a színt kapja. Ha mindkettő különböző színű, akkor a felettük lévő hatszög a harmadik színt kapja. Például az alsó (alulról második sorban található) fehér hatszögnek kéknek kell lennie.

A maradék hatszögeket sorban, alulról felfelé festheted egymás után követve a szabályt.

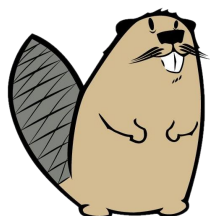
MIÉRT INFORMATIKA?

Hogyan oldjuk meg ezt a hódfeladatot? A hatszög befestéséhez egy műveletet hajtunk végre. A megfelelő művelet (a megfelelő színnel való befestés) kiválasztásához meg kell nézni az alatta lévő hatszögeket, és ellenőrizni kell, hogy milyen feltételnek felelnek meg: azonos színűek vagy különböző színűek. Ezt a tesztet, amelyet egy művelet követ, meg kell ismételni minden olyan hatszögnél, amely még fehér, és két, már festett hatszög felett fekszik.

Műveletek, feltételek, ismétlések: Ezek minden algoritmus alapvető építőkövei, azaz egy pontosan leírt eljárás, amely számítógépre programként is megvalósítható. Tehát amikor megoldottad ezt a hódfeladatot, elkészítettél egy algoritmust. Ez az informatikusok egyik legfontosabb feladata: algoritmusok kitalálása vagy már kitalált algoritmusok alkalmazása és számítógépes programokba való implementálása az automatikus információfeldolgozással kapcsolatos feladatok, problémák megoldása érdekében.

WEBOLDAL

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



HATSZÖGFESTÉS (2022-VN-05B)

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

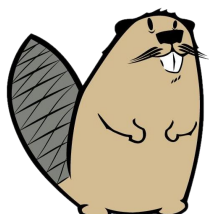
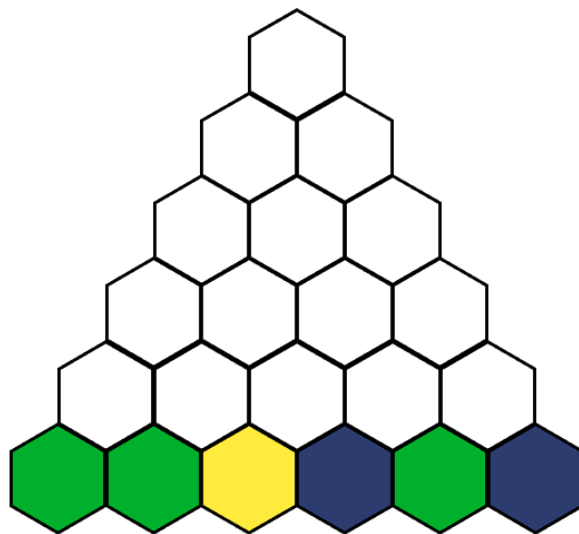
Samu fehér hatszögeket rak egymás mellé, melyeket ezután három különböző színnel fest be az alábbi szabályt követve:

Amikor három hatszög pontosan a képen látható módon van együtt (kettő alul és egy felső középén), akkor vagy mindháromnak azonos színűnek kell lennie, vagy mindháromnak különböző színűnek kell lennie.

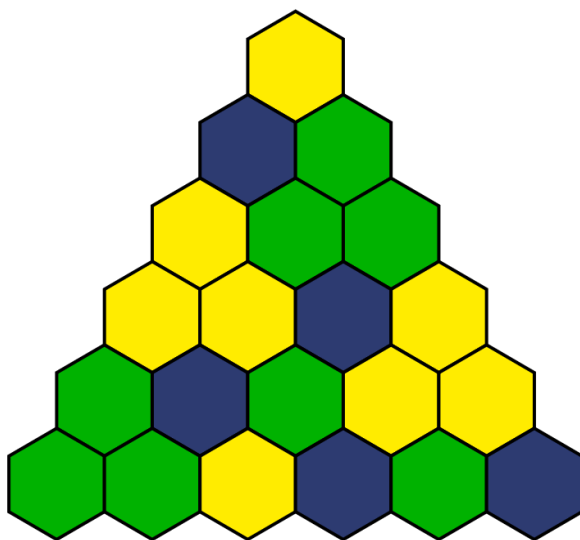
Samu sok hatszöget lerakott, és néhányat már ki is festett.



Fesd ki az összes többi hatszöget a szabályt követve.



A helyes megoldás így néz ki:



Amint felfestünk két hatszöget, amelyek egymás mellett vannak, a felettük középen található hatszög színe egyértelműen meghatározható: ha azonos színűek, akkor a fenti hatszög is ezt a színt kapja. Ha mindkettő különböző színű, akkor a felettük lévő hatszög a harmadik színt kapja. Például az alsó (alulról második sorban található) fehér hatszögnek kéknek kell lennie.

A maradék hatszögeket sorban, alulról felfelé festheted egymás után követve a szabályt.

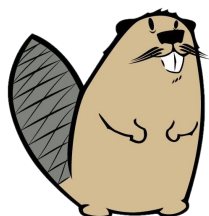
MIÉRT INFORMATIKA?

Hogyan oldjuk meg ezt a hódfelelő feladatot? A hatszög befestéséhez egy műveletet hajtunk végre. A megfelelő művelet (a megfelelő színnel való befestés) kiválasztásához meg kell nézni az alatta lévő hatszögeket, és ellenőrizni kell, hogy milyen feltételnek felelnek meg: azonos színűek vagy különböző színűek. Ezt a tesztet, amelyet egy művelet követ, meg kell ismételni minden olyan hatszögnél, amely még fehér, és két, már festett hatszög felett fekszik.

Műveletek, feltételek, ismétlések: Ezek minden algoritmus alapvető építőkövei, azaz egy pontosan leírt eljárás, amely számítógépre programként is megvalósítható. Tehát amikor megoldottad ezt a hódproblémát, elkészítettél egy algoritmust. Ez az informatikusok egyik legfontosabb feladata: algoritmusok kitalálása vagy már kitalált algoritmusok alkalmazása és számítógépes programokba való implementálása az automatikus információfeldolgozással kapcsolatos feladatok, problémák megoldása érdekében.

WEBOLDAL

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

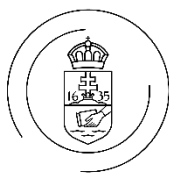
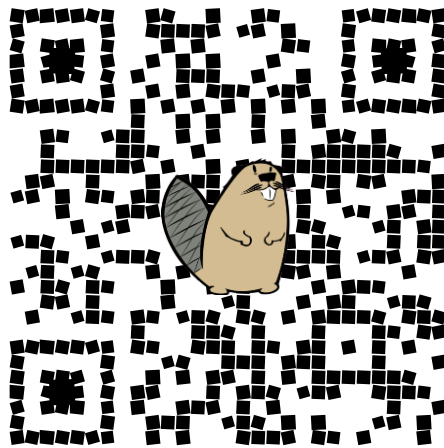


TÁMOGATÓINK, KÖSZÖNETNYILVÁNÍTÁS

Köszönjük a nemzetközi Bebras kezdeményezés országainak, kiemelten a DACH-csapatnak
(Németország, Ausztria, Svájc),
az ELTE IK hallgatóinak, illetve
a kapcsolattartó tanároknak szervezői munkáját.

A HÓD VERSENY MINDEN TARTALMÁRA A CC BY-NC-SA 4.0 LICENSZ VONATKOZIK.

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



ELTE | IK
INFORMATIKAI KAR

