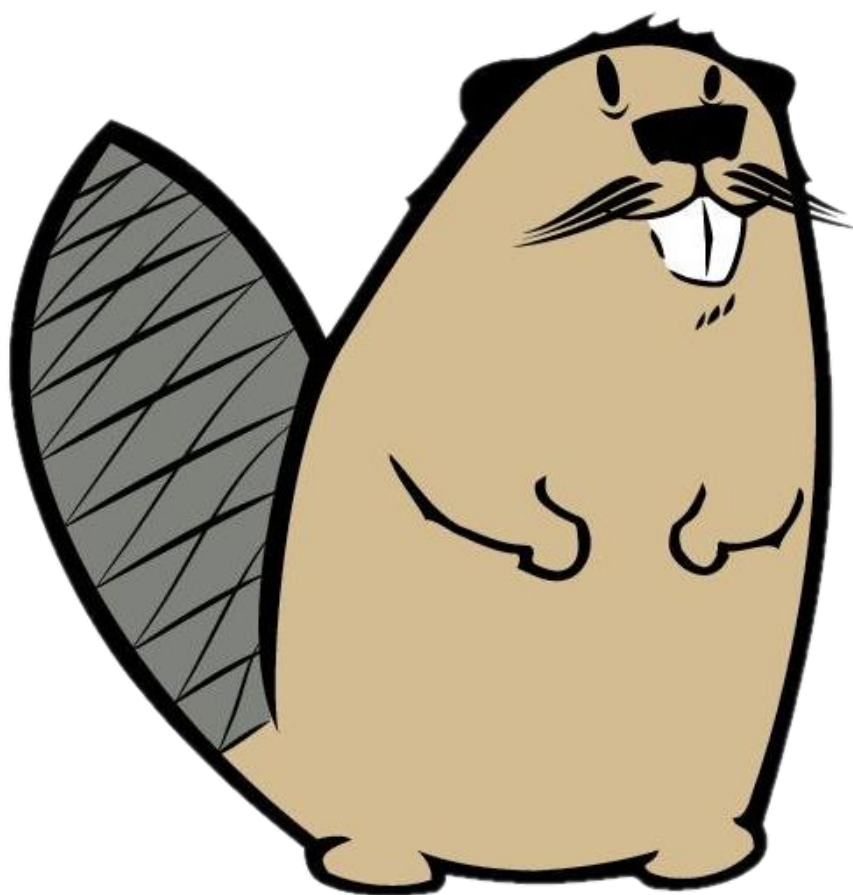


HÓDÍTSD MEG A BITEKET!

**INFORMATIKAI GONDOLKODÁST TÁMOGATÓ, NEMZETKÖZI BEBRAS
KEZDEMÉNYEZÉS MAGYAR MEGVALÓSULÁSA**



2017

MI IS AZ E-HÓD?

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

A nemzetközi Bebras, melyhez 2016-ban 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 keltett verseny, melynek célja, hogy rövid, gyorsan (kb. 3 perc alatt) megérthető és megoldható feladatokkal megvalósítsa az alábbiakat:

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

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

Magyarországon 2017-ben hetedik alkalommal, öt korcsoportban vehettek részt a diákok 4-től 12. osztályig.

A versenyt az ELTE IK T@T Labor és az NJSZT Közoktatási Szakosztálya szervezi.

Az alábbi dokumentumban a 2017-es magyar verseny feladatai és megoldásai találhatóak.

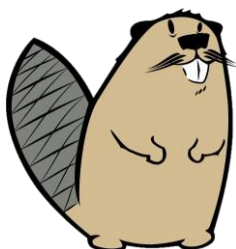
További információkért látogasson el a [HTTP://E-HOD.ELTE.HU/](http://E-HOD.ELTE.HU/) weboldalra, vagy írjon email-t az info@e-hod.elte.hu címre.

RÉSZVÉTEL

A részvétel mindenki számára ingyenes.

A verseny november második 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, 15 feladat három nehézségi szinten: könnyű, közepes és nehéz (legkisebb korosztályban 10 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

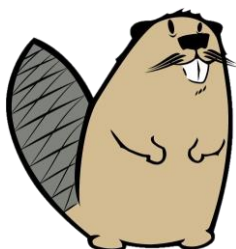
A Kishód korcsoportban 10, minden más korcsoportban 15 feladatot kell megoldani három nehézségi szinten. Minden helyes válasz pontot ér, minden helytelen válaszáért pontlevonás jár.

Nem megválaszolt kérdés esetében az összpontszám változatlan marad.

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

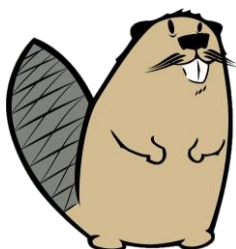
	Könnyű	Közepes	Nehéz
Helyes válasz	6 pont	9 pont	12 pont
Helytelen válasz	-2 pont	-3 pont	-4 pont

Összesen (15 feladat esetében) maximum 135 pontot érhető el.



TARTALOMJEGYZÉK

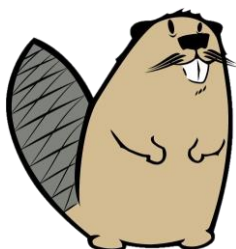
Mi is az E-HÓD?	2
Részvétel	2
Szabályok	3
Értékelés, Pontozás	3
Tartalomjegyzék	4
Feladatok	5
Hírügynökség (2017-CA-14)	6
Madáretető (2017-RO-03a)	8
Karkötők (2015-MY-01)	10
Kapu (2017-AZ-02)	12
Sára cipő (2017-HU-05)	14
Tedd azt! (2011-DE-18)	16
Kártyakód (2017-CZ-02)	18
Mosolyt kérek! (2017-DE-02)	20
Öt pálcika (2017-SK-04)	22
Nyomdagép (2012-AT-01)	24
Kutyák (2017-CA-02)	26
Táncverseny (2017-CA-05)	28
Hírszerkesztés (2017-HU-11)	30
Karperecek (2017-AT-06)	32
Honomakato (2017-DE-06)	34
Nagymama lekvárja (2017-RU-04)	36
Körforgóvár (2017-SI-02)	38
Repülőtér (2013-AT-04)	40
Bibéria Pizzázó (2017-CH-03)	42
Terhelési teszt (2017-HU-04)	44
Frissítő (2017-CA-07)	46
Titkos rendelés (2017-CH-04)	48
Riadó a múzeumban! (2017-DE-03)	50
Labdák (2017-RS-02)	52
Helyettesítés (2017-SI-06)	54



Tiszteletbeli japán név (2017-HR-02).....	56
Az új dal (2017-IR-06)	58
Hódvár járatai (2017-CH-07b)	60
Bringára fel! (2017-DE-09)	63
Fogpiszkálós játék (2017-HU-06)	65
Szótávolság (2017-PL-02)	67
Letöltési lista (2017-RU-01)	69
Öntöző rendszer (2017-AT-05)	71
Pontgyűjtés (2017-CA-12)	73
Képtömörítés (2017-KR-07).....	75
Rövidítés vagy kitérő? (2017-NL-05).....	77
Bontsd részekre a számsort (2017-RU-05)	79
Papírhajtogatás (2012-DE-06)	81
Támogatóink, köszönetnyilvánítás	83

FELADATOK

- Kishód:** 2017-CA-14, 2017-RO-03a, 2015-MY-01, 2017-AZ-02, 2017-HU-05,
2011-DE-18, 2017-CZ-02, 2017-DE-02, 2017-SK-04, 2012-AT-01
- Benjámin:** 2017-AZ-02, 2017-HU-05, 2011-DE-18, 2017-CZ-02, 2017-DE-02,
2017-SK-04, 2012-AT-01, 2017-CA-02, 2017-CA-05, 2017-HU-11,
2017-AT-06, 2017-DE-06a, 2017-RU-04, 2017-SI-02, 2013-AT-04
- Kadét:** 2017-DE-02, 2017-SK-04, 2017-CA-02, 2017-HU-11, 2017-AT-06,
2017-DE-06a, 2017-RU-04, 2013-AT-04, 2017-CH-03, 2017-HU-04,
2017-CA-07, 2017-CH-04, 2017-DE-03, 2017-RS-02, 2017-SI-06
- Junior:** 2017-DE-06a, 2017-RU-04, 2017-SI-02, 2017-HU-04, 2017-CA-07,
2017-CH-04, 2017-RS-02, 2017-SI-06, 2017-HR-02, 2017-IR-06,
2017-CH-07b, 2017-DE-09, 2017-HU-06, 2017-PL-02, 2017-RU-01
- Senior:** 2017-CA-07, 2017-RS-02, 2017-SI-06, 2017-HR-02, 2017-IR-06,
2017-CH-07b, 2017-DE-09, 2017-HU-06, 2017-PL-02, 2017-AT-05,
2017-CA-12, 2017-KR-07, 2017-NL-05, 2017-RU-05, 2012-DE-06



HÍRÜGYNÖKSÉG (2017-CA-14)

KISHÓD – KÖNNYŰ

Viola hódok segítségével szeretne eljuttatni egy üzenetet Leóhoz. Három betűből álló szakaszokra osztja fel az üzenetét, melyet kártyákra ír. Minden hódnak ad egy-egy kártyát.

Viola tudja, hogy sok minden hátráltathatja a hódok útját, így az üzenetdarabkák más-más időpontban érkezhetnek meg. Ezért megszámozza a kártyákat, mielőtt odaadná a hódoknak. Leónak helyes sorrendbe kell tennie a kártyákat, hogy elolvashassa az értelmes üzenetet.

Viola az alábbi kártyákat osztaná ki, ha a "TOLLASBÁL" üzenetet küldené el:

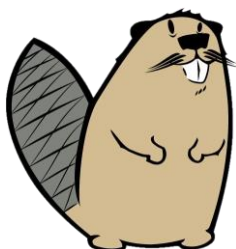
1 TOL	2 LAS	3 BÁL
----------	----------	----------

Leo a következő kártyácskákat kapta meg:

4 NEF	7 ET!	5 ELE	1 NEF
2 ELE	6 JCS	3 DDA	

Mi volt az eredeti üzenet?

- A) ELENEFJCSDDAELENEFET
- B) NEFELEDDAJCSET!NEFELE
- C) NEFELEDDANEFELEJCSET
- D) JCSELENEFNEFDDAELEET



„C” válasz a helyes:

NEFELEDDANEFELEJCSET!

Ha helyes sorrendbe teszed a kártyákat, az alábbi üzenetet kapod:

Ne feledd a nefelejcsset!

MIÉRT INFORMATIKA?

Ha adatokat, pl. e-mailt, képeket vagy videókat küldünk interneten keresztül, ezek kis csomagokra (TCP/IP-csomagok) darabolódnak fel. Minden csomag legfeljebb 65536 jel nagyságú lehet ($2^{16} = 65536 \text{ Jel} = 64 \text{ KB}$). Ezek a csomagok aztán részenként különböző utakon saját, a csomagról szóló (feladó, címzett, a csomag szekvenciaszáma) kiegészítő információikkal kerülnek elküldésre.

Minden ilyen információ gondoskodik arról, hogy az eredeti információ a címzettnél ismét helyesen legyen összeillesztve, még akkor is, ha a részinformációk (részcsomagok) időben elcsúszva érkeznek meg.

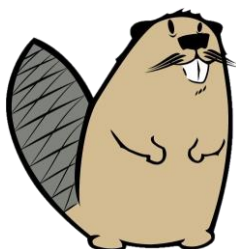
Megjegyzés: IPv6 esetében a csomagok nagyobbak lehetnek.

KULCSSZAVAK

Internet-protokoll, csomagok

WEBOLDALAK

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



MADÁRETETŐ (2017-RO-03A)

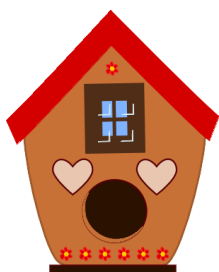
KISHÓD – KÖNNYŰ

Hódmama kislánya születésnapjára ajándékot szeretne venni. Egy madáretetőre gondol, így megkérdezte tőle, hogy milyen madáretetőnek örülne a legjobban.

A lánya ezt válaszolta: "Olyan madáretetőt szeretnék, ami kétablakos és van rajta egy szív".

Hódmama útra kelt, hogy megvegye a kívánt madáretetőt.

Az alábbi madáretetők közül melyiket veheti meg hódmama a kislányának?



A)



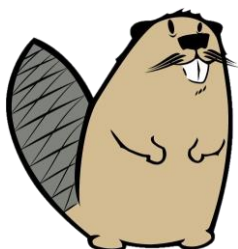
B)



C)



D)



„B” válasz a helyes: Csak ezen a madáretetőn van 2 ablak és 1 szív.

Az A nem lehet, mert ezen a házon csak 1 ablak van és 2 szív.

A C sem lehet, mivel nincs rajta szív, hiába van 2 ablaka.

A D sem megfelelő, mert csak 1 ablaka van.

MIÉRT INFORMATIKA?

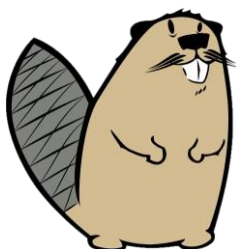
A feladat nehézsége abban rejlik, hogy mind a négy házra meg kell vizsgálni, hogy azok megfelelnek-e a hódikislány két óhajának. A ház színei és egyéb különlegességei nem lényegesek, így a fölösleges információkat egyidejűleg ki is kell szűrni, csak bizonyos tulajdonságokat szabad nézni. A gyakorlatban is a jelentős dolgokra kell koncentrálni.

KULCSSZAVAK

Minta, mintafelismerés, absztrakció

WEBOLDALAK

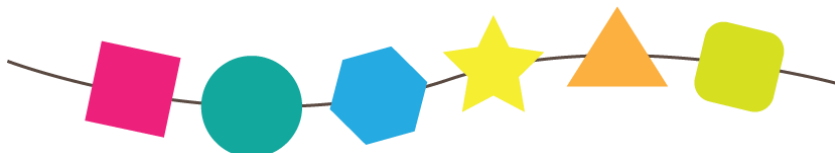
<https://hu.wikipedia.org/wiki/Absztrakció>



KARKÖTŐK (2015-MY-01)

KISHÓD – KÖNNYŰ

Lilinek különböző formájú gyöngyökből álló fűzött karkötője van. Egy nap elszakadt a karkötő és már nem is lehet megjavítani. Az elszakadt karkötő így néz ki:

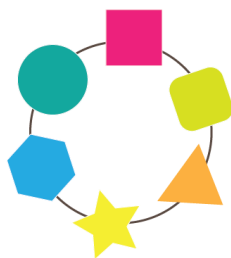


Lili pontosan ugyanilyen karkötőt szeretne. Az üzletben négy különböző karkötőt árulnak.

Melyik teljesen ugyanolyan, mint Lili elszakadt karkötője?



A)



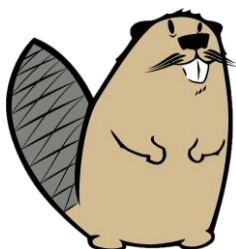
B)



C)



D)



„B” válasz a helyes.

A B karkötő követi az elszakadt karkötő gyöngyeinek sorrendjét.

Az A karkötőnél a narancs háromszög és a sárga csillag felcserélődött.

A C karkötőnél a narancs háromszög és a kék hatszög van rossz helyen.

A D karkötőnél a sárga csillag és a zöld lekerekített sarkú négyzet van rossz helyen.

MIÉRT INFORMATIKA?

Az informatikában fontos, hogy felismerjünk olyan mintázatokat, amelyek hasznosak lehetnek számunkra. A mintázatok felismerése segít megtalálni a hasonlóságokat bizonyos dolgok között, amik első ránézésre különbözőnek tűnhetnek, de van bennük valami közös.

Amikor egy új problémánál észrevesszük, hogy hasonlít egy másik, már megoldott problémához, akkor a hasonló probléma megoldásából megtalálhatjuk az új probléma megoldását: ez a módszer nemcsak informatikai problémákra, hanem matematikai és más tudományos kérdésekre is segít választ adni.

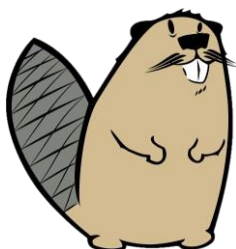
Ez a feladat a megoldás vizsgálatát is igényli: a lehetséges válaszokban szereplő karkötőket össze kell hasonlítani az eredeti karkötő látható változatával, hogy egyezzen az alakzatok sorrendje. A megoldás ellenőrzésének ilyen módja fontos az informatikában, hogy meghatározzuk, hogy egy program kimenetele helyes-e.

KULCSSZAVAK

Mintafelismerés, sorozat, ciklikus permutáció

WEBOLDALAK

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



KAPU (2017-AZ-02)

KISHÓD – KÖZEPES

BENJAMIN – KÖNNYŰ



A hódok gyakran meglátogatják egymást. De néha nincs otthon senki sem.

Ekkor hagynak egy üzenetet a kő kertkapujukon, mellyel a várható érkezésükről adnak hírt. A híradás az alábbiak szerint történik: Legfeljebb 3 faágat illesztenek be a kőkapu szemközti furataiba.

A hódok az alábbi jelentésekben állapodtak meg.



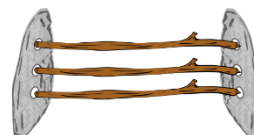
Itthon vagyunk,
gyere csak be.



Csak délben érünk
haza.



Sajnos csak este
jövünk.

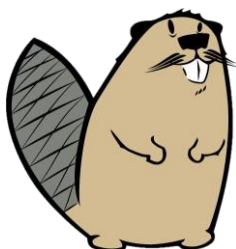


Látogatóban
vagyunk, éjfél
körül érkezünk.

A hódok további üzenetekben is megállapodhatnak anélkül, hogy további faágat vagy furatot használnának.

Hány további üzenetben állapodhatnak meg a hódok?

- A) 2
- B) 3
- C) 4
- D) 5



„C” válasz a helyes: 4 új kódban állapodhatnak meg.

A különböző üzenetek száma az alábbi módon adható meg:

A furatok lehetséges állapotainak számát (2) emeljük a furatpárok számának hatványára (3), vagyis $2^3 = 2 \cdot 2 \cdot 2 = 8$

A 4 meglévő kód mellett tehát 4 további kód készíthető a három faággal.

Az új állapotok a következők:



MIÉRT INFORMATIKA?

Ebben a feladatban a hódok a kettes (bináris) számrendszert használják. Az egymással szemben található furatok az információhordozók, páronként 2 állapottal: vagy van benne faág, vagy nincs.

A hódok hagyományból tudják, hogy mit jelentenek az üzenetek. Ha mégis tévednének, nagy kár nem származna belőle. Az informatika világméretű hálózati rendszerében azonban mindenki mindenkivel közvetlen kommunikációs szomszéd és hibátlanul meg kell érteniük egymást.

A nagy szervezetek gondoskodnak a jelrendszerek szabványosításáról, standardizálásáról. A világ minden tájáról képviselt szakbizottságokban döntenek egy jel kinézetéről és annak jelentéséről. A legfontosabb számrendszereket az országgyűlések törvényileg szabályozzák, így érik el, hogy a világ összes számítógépe (is) jól megértse egymást.

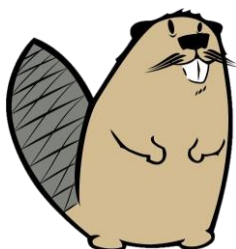
KULCSSZAVAK

Bináris számok, kódolás, információ reprezentálása, permutáció

WEBOLDALAK

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

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



SÁRA CIPŐI (2017-HU-05)

KISHÓD – KÖZEPES

BENJAMIN – KÖNNYŰ

Sára az édesapjával elmegy új lábbelit vásárolni, de nem akarnak 100 eurónál többet kiadni.

Sára kedveli a fekete színt és a csizmákat.

Melyik lábbelit fogja választani?



84 Euro

A)



114 Euro

B)



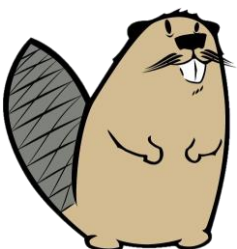
94 Euro

C)



84 Euro

D)



„A” válasz a helyes.

A helyes válasz az első cipő. Fekete, csizma és az ára 100 euró alatt van.

A másik fekete csizma túl drága, a harmadik fekete pedig cipő, nem csizma.

Ebben a feladatban a lábbeliket tulajdonságokkal illetjük (szín, ár, típus). Mindegyik jellemzőnek két állapota van:

- szín: fekete vagy sem
- típus: csizma vagy sem
- ár: olcsóbb vagy drágább 100 eurónál

Azt a lábbelit kell megtalálnunk, mely mindhárom jellemzőnek megfelel.

MIÉRT INFORMATIKA?

Az informatikán belül nem csak a programozásban fontos, hogy kiértékeljünk és feltételeket egyeztessünk.

Többnyire nem csak egy-két feltételünk van, melyek között kapcsolatok vannak.

Ezek a kapcsolatok lehetnek ÉS típusúak (mint ebben a feladatban), ahol minden feltételnek egyszerre kell teljesülnie, hogy az egész feltétel-halmaz igaz legyen. Ha a feltételek közül elég csak az egyiknek teljesülnie, akkor VAGY kapcsolatról beszélünk.

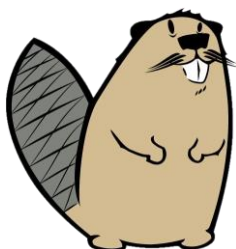
Vannak másfajta kapcsolatok is, mint amikor az egyik tulajdonság kihat egy másikra (pl. jobb minőségű anyagból készülő lábbelik általában drágábbak).

KULCSSZAVAK

Logika, kapcsolatok

WEBOLDALAK

<http://hu.wikipedia.org/wiki/Konjunkció>



TEDD AZT! (2011-DE-18)

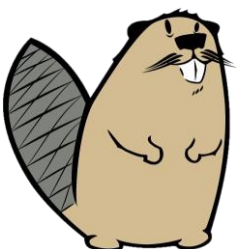
KISHÓD – KÖZEPES

BENJAMIN – KÖNNYŰ

A nagyon egyszerű programok utasítások puszta sorozatából állnak. Egy utasítás egy elvégzendő tevékenységet ír le.

A következő szövegek melyike értelmezhető nagyon egyszerű programként?

- A) „Mi az információ?”
- B) „Kettő meg kettő egyenlő négy.”
- C) „Gyere be és csukd be az ajtót!”
- D) „Üdvözlünk a valóságban!”



„C” válasz a helyes:

A C egy egyszerű program: az 1. parancs az, hogy „gyere be”, a 2. parancs pedig az, hogy „Csukd be az ajtót”. Megfelelően működik, ha kívülről indulsz és az ajtó nyitva van, és a parancsokat egymás után végzed el. Mi történik, ha az ajtó zárva van?

Az A nem egy program, hanem egy kérdés. A B egy egyenlet, de nincs tennivaló – Orwell az igazság alapvető állításaként használta. A D egy üzenet valakinek, aki kilépett a Mátrixból, de nincs tennivaló.

MIÉRT INFORMATIKA?

Számos tennivalót kifejezhetünk programként az emberek, robotok vagy számítógépvezérelt kütyük számára. Ahhoz, hogy valamit kifejezhessünk, egy nyelvre van szükségünk.

Sokan vitatják, hogy a természetes emberi nyelvek – legyenek beszélt vagy írott nyelvek – alkalmasak-e hosszútávon programozásra. Az informatika ezzel a problémával foglalkozik.

Ezidáig a programozási nyelvek szigorú és pontos megfogalmazást követeltek egy jól felépített, logikus elrendezésű szókinccsel. Ha programozni tanulsz, meg kell tanulnod átfogalmazni a természetes nyelven kigondolt feladatokat és utasításokat egy szintaktikusan helyes és szemantikailag majdnem hibamentes utasítássorozattá.

A fentiek miatt sokan állítják, hogy a programozás nem csak mesterség, hanem művészet is.

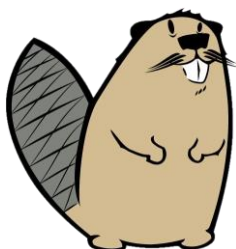
KULCSSZAVAK

Programozási nyelv, utasítás

WEBOLDALAK

http://wiki.prog.hu/wiki/Mi_az_a_programozas

<https://hu.wikipedia.org/wiki/Számítógép-programozás>



KÁRTYAKÓD (2017-CZ-02)

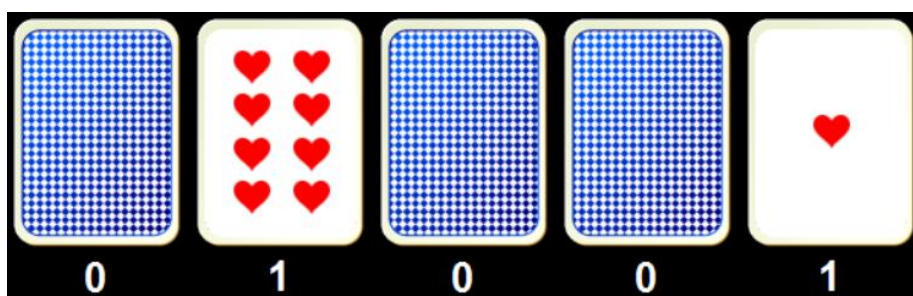
KISHÓD – NEHÉZ

BENJAMIN – KÖNNYŰ

5 kártyát fektettünk egymás mellé. Balról jobbra a kártyákon szereplő szívek száma: 16, 8, 4, 2, 1.

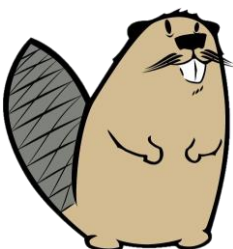
Minden kártya alatt az 1 vagy a 0 szám szerepel. Ha a szám 1, akkor a kártya képpel felfele néz, azaz a szívek látszanak. Ha a szám 0, akkor képpel lefele van a kártya és a szívek rejtve maradnak.

Így a kártyákat számok kódolására használhatjuk. Például a képen 9 szív látható, így a 9-es számot a 01001 kódolja.



Melyik kód felel meg a 26 szívnek?

- A) 11010
- B) 11101
- C) 10011
- D) 11001



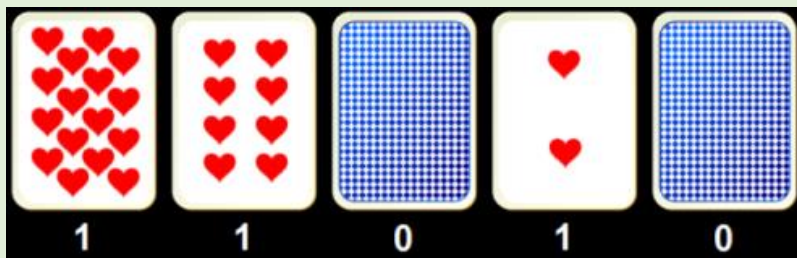
„A” válasz a helyes.

A kép is mutatja, hogy a kód a következő $16 + 8 + 2 = 26$ szív.

Ez az egyetlen helyes válasz. Hogy ezt bizonyítsuk, nézzük meg, hogy a $8 + 4 + 2 + 1 = 15$ és ahhoz, hogy 26 szívet kapjunk, fel kell fordítanunk a legbalrább található kártyát. $26 - 16 = 10$, melynek kirakásához az utolsó 4 kártya mindegyikére szükségünk van (mivel $4 + 2 + 1 = 7$).

Így fel kell fordítanunk a 8 szívet ábrázoló kártyát is. Ekkor $10 - 8 = 2$ szív maradt, amit meg kell jelenítenünk. A 4 szívet ábrázoló kártya már túl sok, így azt lefele fordítjuk.

Az 1 szívet tartalmazó magában túl kevés, így azt is lefele fordítjuk és a 2 szívet ábrázoló kártyát kell felfele fordítanunk.



MIÉRT INFORMATIKA?

A számok ehhez hasonló ábrázolását bináris, azaz kettes számrendszerbeli ábrázolásnak nevezzük.

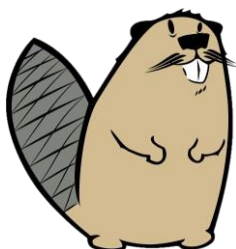
Minden tetszőleges szám (vagy akár szöveg, kép, videó) leírható bináris kódokkal. A bináris (kettes vagy duális) számrendszer szinte minden modern számítógépben használatos. Ennek praktikus oka, hogy a logikai áramköröket, melyek a számítógépbe vannak beépítve, könnyebb megvalósítani kettes (bináris) rendszerben, mint tízesben (decimálisban).

KULCSSZAVAK

Bináris számrendszer, bit

WEBOLDALAK

http://hu.wikipedia.org/wiki/Gépi_számaábrázolás



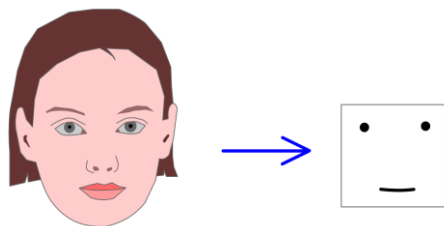
MOSOLYT KÉREK! (2017-DE-02)

KISHÓD – NEHÉZ

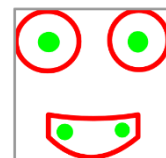
BENJAMIN – KÖZEPES

KADÉT – KÖNNYŰ

A hódok kifejlesztettek egy alkalmazást, amely felismer egy mosolygó arcot. Az alkalmazás először egy modellt készít az arcképről, mely hasonlít egy szmájlira. Két pontból és egy vonalból áll, amelyek a szemek és a száj pozícióját ábrázolják.



Ezt követően az alkalmazás egy zöld pontokból és piros vonalakból álló sablon segítségével megvizsgálja a modellt.



Mosolygós arcként azonosítja, ha annak pontjai és vonala:

1. minden zöld pontot érintenek, és
2. nem metszenek egy piros vonalat sem.

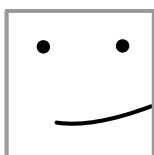


Mosolygós arc

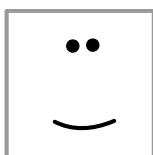


Nem mosolygó arc
A száj metszi a piros vonalat.

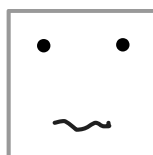
Az alkalmazás 4 képről készített modellt. Ezekből egyet azonosított mosolygós arcként. Melyiket?



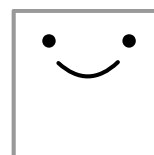
A)



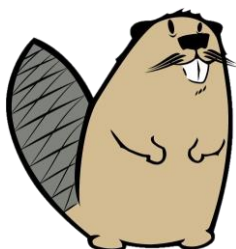
B)



C)



D)



„C” válasz a helyes: itt a szemek jó helyen vannak és a száj is a megadott keretben van. Az már más kérdés, hogy az arc láthatóan nem mosolyt ábrázol.

Az A kép esetében a száj kilóg a megadott keretből.

A B kép esetében a szemek távolsága nem megfelelő.

A D kép esetében a száj nincs a kívánt tartományban.

MIÉRT INFORMATIKA?

Az informatikusok olyan rendszereket fejlesztenek, amelyek kameraképeken arcokat, tárgyakat ismernek fel. Ez a technika igen nagy hatalommal bír, az azonosítást ma már mesterséges intelligencia végzi. Azonban az egyszerűbb szabályok - mint például ennek a feladatnak a kikötései - is nagyon fontosak.

A feladat a képfelismerő rendszerek nehézségeit mutatja be. Jelen esetben a szabályok túl egyszerűek ahhoz, hogy minden bizonyossággal mosolygós arcok kerüljenek kiválasztásra. A rendszer által mosolygósnak értelmezett két arc közül az egyik inkább mogorva arcot ábrázol. Ehhez hasonló hibákat a való életben használt arcfelismerő rendszerek is elkövethetnek például egy automatikus útlevél-ellenőrzéskor vagy egy hatósági eljárás során. 2016-ban egy ázsiai diák útlevélkérelmét utasították el, mert a számítógép nem ismerte fel a szemét az igazolványképen.

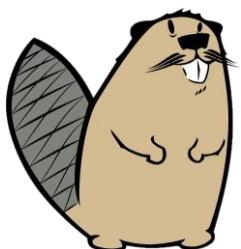
Forrás: <http://www.telegraph.co.uk/technology/2016/12/07/robot-passport-checker-rejects-asian-mans-photo-having-eyes/>

KULCSSZAVAK

Képfelismerés, algoritmus alkalmazása, gépi látás

WEBOLDALAK

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



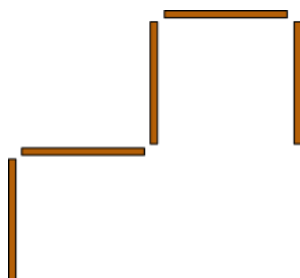
ÖT PÁLCIKA (2017-SK-04)

KISHÓD – NEHÉZ

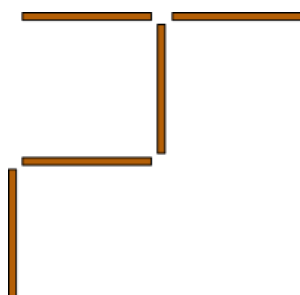
BENJAMIN – KÖZEPES

KADÉT – KÖNNYŰ

Egy asztalon öt pálcikát helyeztünk el az alábbi módon:

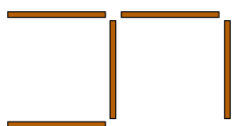


Nóra megfog egy pálcikát és áthelyezi, ahogy az ábrán is látható:

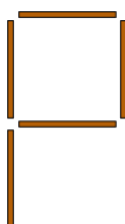


Ezt követően Béla ragad meg egy pálcát és helyezi át máshova.

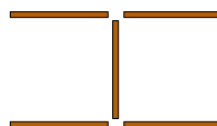
Béla átrendezése után mely módon NEM helyezkedhetnek el a pálcák az asztalon?



A)



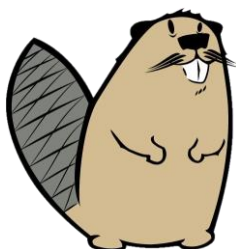
B)



C)



D)



„D” válasz a helyes.

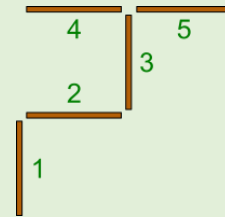
Rendeljünk egy-egy számot a pálcikákhoz! Nóra rendezése után a pálcikák a következő módon helyezkednek el:

Ahhoz, hogy az A jelzésű képet kapjuk, Bélának az 1-es pálcikát kell áthelyeznie máshova.

Ahhoz, hogy a B jelzésű képet kapjuk, az 5-ös pálcikát kell áthelyeznie.

Ahhoz, hogy a C jelzésű képet kapjuk, az 1-es pálcikát kell áthelyeznie.

Ahhoz, hogy a D jelzésű képet kapjuk, Bélának az 1-es és az 5-ös pálcikát kellene áthelyeznie. A D jelzésű kép tehát nem kapható meg egy pálcika áthelyezésével, így ez a helyes megoldás.



MIÉRT INFORMATIKA?

A pálcikák helyzetének megváltoztatására Nóra és Béla egyszerű műveletet talált ki: pontosan egy pálcikát helyeznek át. Ezáltal a lehető legkisebb módon változik meg a pálcikák helyzete, illetve állapota. Ha egy barátjuk határozná meg a feladatukat, megfelelő utasítást kellene adnia: Helyezz át egy pálcát! - de azt is meg kell adnia, hogy melyiket, hova. Amennyiben a barátjuk komolyabban át szeretné rendeztetni a pálcikákat, akkor megteheti úgy is, hogy ezt az egy utasítást adja ki egymás után többször, így további utasítások mint pl. „Helyezz át kettő pálcikát!”, „Helyezz át három pálcikát!” stb. nem szükségesek.

Aki számítógépes programokat ír, utasítások sorozatát fogalmazza meg, amelyet a számítógép már ismer. Lehet azonban, hogy új utasításokat talál ki, azonban ekkor is törekedni kell az egyszerűsége, azaz hogy a számítógép állapota a lehető legkisebb módon változzon meg. A bonyolult utasítások tehát egyszerű utasításokból épülnek fel, melyek az alábbiak lehetnek:

- több utasítás egymásutánisága,
- egy utasítás egymás után többszöri végrehajtása, vagy
- egy utasítás végrehajtása meghatározott feltétel teljesülése esetén.

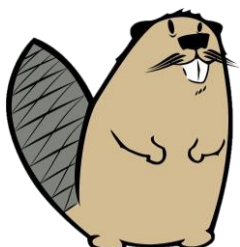
Ez a programozók három legfontosabb módszere, ha a kezdetben egyszerű utasításokból egyre bonyolultabbakat szeretnének felépíteni. Nem is olyan bonyolult!

KULCSSZAVAK

Utasítások, állapotváltozás, programozás

WEBOLDALAK

<http://www.logikaifeladatok.hu/gyufas.html>



NYOMDAGÉP (2012-AT-01)

KISHÓD – NEHÉZ

BENJAMIN – KÖZEPES

Egy egyszerű nyomdagépet kártyákkal vezérelhetünk. A gépnek egy piros papírt kell átszíneznie.

A programkártyák parancsait egymás után (1-2-3-4) kell végrehajtani:

1. színezd a papír alsó felét kékre (ez lesz az ég)
2. fordítsd meg a papírt 180 fokkal
3. színezd a papír alsó felét zöldre (ez lesz a fű)
4. a jobb felső sarokba nyomdázz egy sárga foltot (ez lesz a nap)

Az utasítások hatására így változik meg a piros papír:



A kártyák sajnos összekeveredtek, így a parancsok más sorrendben (3-1-2-4) kerültek végrehajtásra.

Hogy nézett ki ezek után a piros papír?



A)



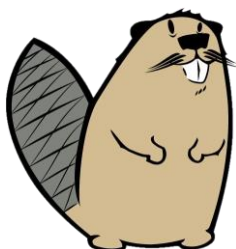
B)



C)



D)



„B” válasz a helyes:

Az A kép hibás, mivel a második utasításban a zöld területet kékre kell színezn.

A C és D képek hibásak, mert az első két utasítás (3 és 1) elvégzése után a papírnak piros és kék színűnek kellene lennie.

MIÉRT INFORMATIKA?

A feladat az algoritmikus gondolkodáson alapszik. Adott egy lépésről lépésre vezető útmutató – mint egy számítógépes program. A feladat példa arra, hogy egy bizonyos program utasításainak sorrendjét megváltoztatva egészen más kimenetelt kapunk.

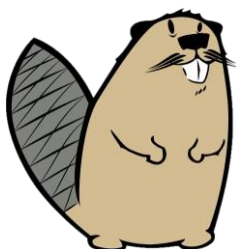
A végrehajtási sorrend megváltoztatása nem csak véletlenül történhet. Előfordulhat, hogy egy programot módosítani kell, hogy lefedjen olyan feladatokat, amikre nem gondoltunk a program készítésekor. Az ilyen esetekben nagyon óvatosnak kell lennünk, hogy a program kimenetelét ne változtassuk meg a feladatok sorrendjével. Emiatt szükséges egy program megírása előtt pontos tervet készíteni és ha lehetséges, akkor olyan módon megírni azt, hogy később további funkciókkal bővíthető legyen.

KULCSSZAVAK

Programtervezés, algoritmus

WEBOLDALAK

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

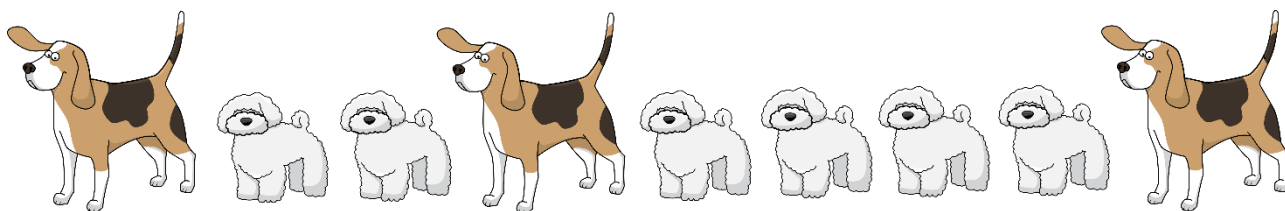


KUTYÁK (2017-CA-02)

BENJAMIN – KÖNNYŰ

KADÉT – KÖNNYŰ

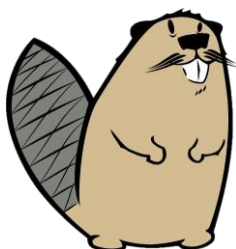
Két különböző fajtájú kutyát állítottunk egy sorba. Egy csere akkor zajlik le, ha két egymás mellett álló kutya helyet cserél. Egyszerre csak egy csere történhet.



Néhány csere után a három nagy kutya egymás mellé került.

Legkevesebb hány csere szükséges ahhoz, hogy a három nagy kutya egymás mellé kerüljön?

- A) 5
- B) 6
- C) 7
- D) 8



„B” válasz a helyes: 6

Balról az első kutya 2, jobbról az első kutya pedig 4 cserével kerül a középső nagy kutya mellé.

Mindegyik kiskutya egyszer cserél helyet, ugyanis mindegyiket közvetve nagykutya fognak közre. Nem jutunk közelebb a megoldáshoz, ha egy kiskutyát másik kiskutyával cserélünk ki. A cserének akkor van értelme, ha minden lépésben egy kis- és egy nagykutya helyzetét cseréljük fel. Összesen 6 kiskutya van, ami azt jelenti, hogy legkevesebb 6 csere szükséges.

MIÉRT INFORMATIKA?

Az adatok a számítógép memóriájában tárolódnak. Ez magába foglalja a belső tárat (RAM) és a külső memóriát (pl. merevlemez, pendrive). A számítógép gyorsabban hozzáfér belső tárban tárolt adatokhoz, azonban a külső táruk költségkímélőbbek ezeknél. Ebből az következik, hogy a számítógépek rendszerint sokkal nagyobb külső memóriával rendelkeznek, azonban a szakemberek keresik a lehetőséget, hogy a belső memóriát maximálisan kihasználhassák.

Jelen esetben a feladat két elem cseréjéről szól. Ha a kutyákra úgy tekintünk, mint a memóriában tárolt adatokra, akkor egy csere két adategység memóriában történő felcserélését jelenti. Amennyiben külső memóriáról van szó, akkor a feladat megoldásához a lehető legkevesebb cserét kell végrehajtani.

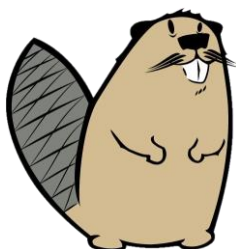
A cserék különböző rendezési algoritmusoknál is előjönnek. A rendezési folyamat annál hatékonyabb, minél kevesebb cserét igényel. Például a buborékos rendezés minden lépésben szomszédos elemeket cserél fel, még akkor is, ha ezek még nem a kívánt sorrendben kerülnek.

KULCSSZAVAK

Rendezés, algoritmikus gondolkodás, csere

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Rendezés_\(programozás\)](https://hu.wikipedia.org/wiki/Rendezés_(programozás))



TÁNCVERSENY (2017-CA-05)

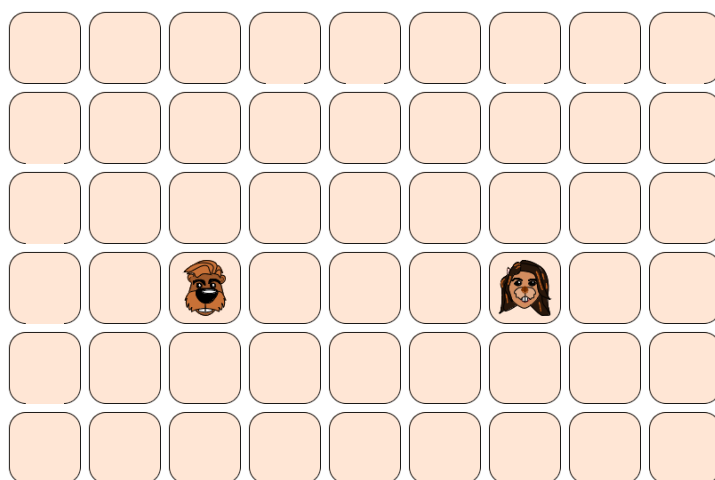
BENJAMIN – KÖZEPES

Egy mókus és egy hód táncversenyen vesznek részt. A közönség reakciói alapján meghatározott tánclépéseket mutatnak be. Az alábbi táblázat azt mutatja, hogy az adott közönségreakcióra hogyan mozognak:

	Váóóó!	Sikítás	Taps	Igeen!
	← ↑	↑ ←	← ← ↑	↓ ↓
	↑ →	→ ↓	↑ ↑ ↑	← ←

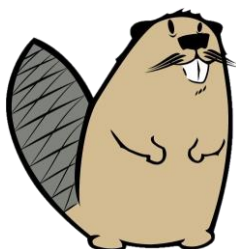
Például, ha a közönség sikít, akkor a mókus felfelé táncol egy mezőt, majd balra szintúgy egyet. Ugyanakkor a hód egyet sasszézlik jobbra, majd egyet le.

A kőr elején az alábbi kiindulópályákat vették fel:

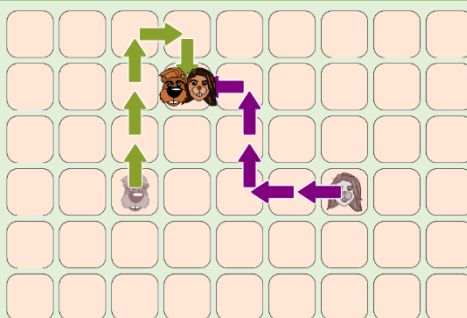


Az alábbiak közül melyik nézőreakciók után kerülnek ugyanarra a mezőre mindketten?

- A) Igeen! Sikítás
- B) Váóóó! Sikítás
- C) Sikítás Sikítás
- D) Taps Sikítás



„D” válasz a helyes: Taps Sikítás:



Minden más esetben különböző mezőre érkeznek táncosok:

A) Igeen! Sikítás:	B) Váóó! Sikítás:	C) Sikítás Sikítás:

MIÉRT INFORMATIKA?

Ez a feladat jól szemlélteti a párhuzamos műveleteket. Jelen esetben a két táncos mozgása egymástól független, de függ egy előre meghatározott szabálytól. Ha két szereplő (pl. RAM és egy csatlakoztatott készülék) „ütközik”, azaz egyidejűleg szeretnének hozzájutni ugyanahhoz az erőforráshoz, szabályoznunk kell, hogy melyik férhet hozzá először. A programozás területén ezt a feladatot végzi például a szemafor, a számítógép szintjén pedig az operációs rendszer. Ezek lehetőséget adnak egy szereplőnek, hogy az erőforrást lefoglalja, használja, majd ezek után felszabadítsa. Amíg foglalt az erőforrás, más szereplő nem férhet hozzá.

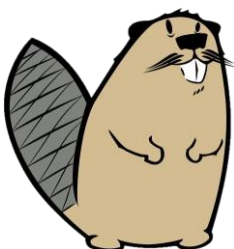
Amikor te a tánclépések sorrendjén gondolkoztál a képernyőn, tulajdonképpen egy kétszereplős szimulációt hajtottál végre. A számítógépes szimulációk nagyon sok területen segítik az emberek munkáját, gondoljunk csak az időjárás-előrejelzésre, ami ugyancsak szimuláción alapul.

KULCSSZAVAK

Optimalizálás, párosítás

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Utastitas_\(informatika\)](https://hu.wikipedia.org/wiki/Utastitas_(informatika))
[https://hu.wikipedia.org/wiki/Szal_\(programozas\)](https://hu.wikipedia.org/wiki/Szal_(programozas))



HÍRSZERKESZTÉS (2017-HU-11)

BENJAMIN – KÖZEPES

KADÉT – KÖNNYŰ

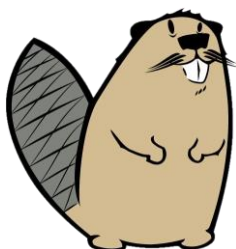
10 diák dolgozik az iskolaújságon.

Minden pénteken saját cikkeiket írják és szerkesztik. Az alábbi táblázatban a zöld cellák mutatják, hogy a diákoknak mikor van szükségük számítógépre. A számítógépek mind egyformák, és minden órában csak egy diák használhat egy adott gépet.

Diák \ Idő	8	9	10	11	12	13	14	15
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								

Legalább hány számítógépre van szükségük a diákoknak, hogy mindenki a fenti beosztás szerint dolgozhasson (és ne kelljen megszakítania a munkáját)?

- A) 4 számítógépre
- B) 5 számítógépre
- C) 6 számítógépre
- D) 8 számítógépre



„B” válasz a helyes:

9:00-kor és 10:00-kor 5 diáknak van szüksége számítógépre, tehát 5 gépnél kevesebbel nem oldható meg a probléma.

Ha megfelelően rendezzük az adatokat, ahogy azt a következő táblázat mutatja, láthatjuk, hogy 5 számítógép elég is.

Diák \ Idő	8	9	10	11	12	13	14	15
1		PC 3	PC 3					
2			PC 1	PC 1	PC 1	PC 1		
3	PC 1	PC 1						
4					PC 3	PC 3	PC 3	
5		PC 4	PC 4					
6				PC 2	PC 2			
7			PC 5	PC 5		PC 5	PC 5	
8		PC 5						
9	PC 2	PC 2	PC 2					
10						PC 2	PC 2	

Amikor egy diák megérkezik, leül az első szabad számítógéphez. Amikor befejezte a munkát, egy másik diák jön és leül a géphez.

MIÉRT INFORMATIKA?

Ahhoz, hogy nagy mennyiségű adatot és a különböző típusú adatok közti kapcsolatokat megértsük, a legjobb egy adatábrázolást készíteni, mint például egy táblázatot vagy diagramot. A szóban forgó információktól függően, ugyanazt az adathalmazt többféleképpen ábrázolhatjuk.

A jelenlegi feladatban egy másik ábrázolási mód a következő:

PC \ Idő	8	9	10	11	12	13	14	15
1	Diák 3	Diák 3	Diák 2	Diák 2	Diák 2	Diák 2		
2	Diák 9	Diák 9	Diák 9	Diák 6	Diák 6	Diák 10	Diák 10	
3		Diák 1	Diák 1		Diák 4	Diák 4	Diák 4	
4		Diák 5	Diák 5					
5		Diák 8	Diák 7	Diák 7		Diák 7	Diák 7	

Az időbeosztás és a rendelkezésre álló erőforrás optimális felhasználása a valós életben is gyakori probléma. Erre jó példa egy hotel szobafoglalási rendszere, ahol két foglalás nem ütközhet, és a szobák beosztását hatékonyan kell végezni. Ugyanígy, ha egy konferenciateremben többféle előadást tartanak, egy hatékony beosztás hasznos lehet.

Egy másik mód az adatok ábrázolására, ha gráfokat használunk, ahol a csúcsok az időtartamok és két csúcs között az élek az ütköző időtartamokat jelzik. Az ilyen gráfot intervallumgráfnak nevezzük.

KULCSSZAVAK

Időbeosztás, intervallumgráf, ábrázolás táblázatban

WEBLDALAK

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

ELTE IK T@T Labor – CC BY NC-SA

<http://e-nod.elte.hu>

KARPERECEK (2017-AT-06)

BENJAMIN – NEHÉZ

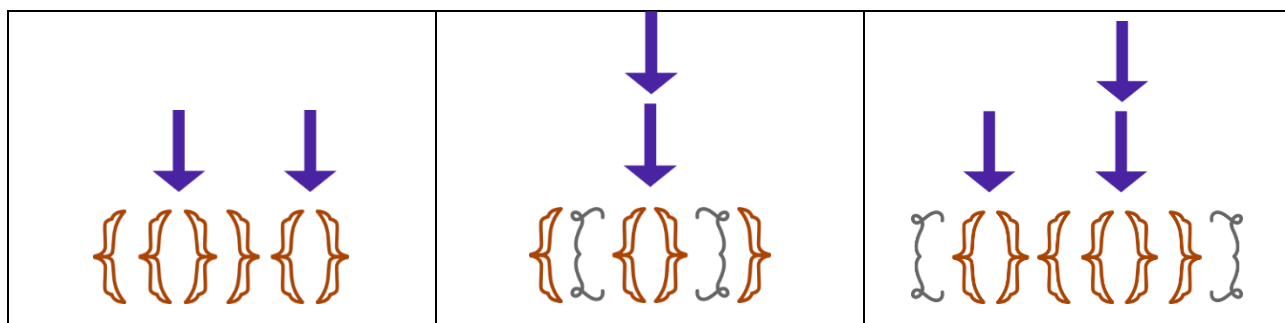
KADÉT – KÖNNYŰ

Hód család saját készítésű karpereceket készít egy közép kort felidéző ünnepségre. Ehhez zárójelre hasonlító motívumokat használ, melyek azonban mindig párban állnak: nyitó és záró motívum.

Egy karperec elkészítése az első pár kiválasztásával kezdődik, ugyanis a család két dísz közül is választhat:



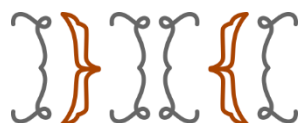
Ezután újra és újra további motívumpárokat szúrnak be tetszőleges helyre, ahogy az alábbi példa is mutatja:



Az alábbi karperecek közül melyik készült a fent leírt módon?



A)



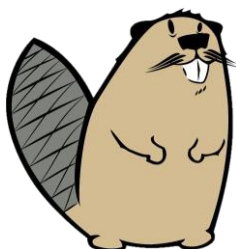
B)



C)



D)



„D” válasz a helyes:

A legelső zárójelpárba egy másikat illesztettek be, amelybe pedig egy újabbat.

A többi karperec nem a fent leírt módon készült:

A) Az első hiba balról nézve a 3. helyen található, mielőtt még zárulna a 2. helyen álló motívum.

B) Balról nézve már az első helyen lévő motívum is hibás: záró motívum áll ott, pedig maga a motívum még „ki sem lett nyitva”.

C) Az első hiba balról nézve a 4. helyen áll: olyan zárómotívum áll ott, ami meg sem lett nyitva.

MIÉRT INFORMATIKA?

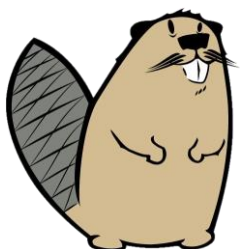
A karpereckészítés szabálya megegyezik a matematikában vagy az informatikában ismert zárójelezéssel. Helyesen zárójelezett kifejezéseket szintaktikailag helyesnek is mondjuk abban az értelemben, hogy eleget tesznek az előírt szintaktikai szabálynak (egy formális nyelv nyelvtanának). Szintaktikai hibákat sokkal könnyebb kijavítani, mint a szemantikai hibákat, melyek formailag ugyan helyesek, de hibás gondolatmenet írnak le.

KULCSSZAVAK

Szabályos zárójelezés, szintaxis, szemantika

WEBOLDALAK

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



HONOMAKATO (2017-DE-06)

BENJAMIN – NEHÉZ

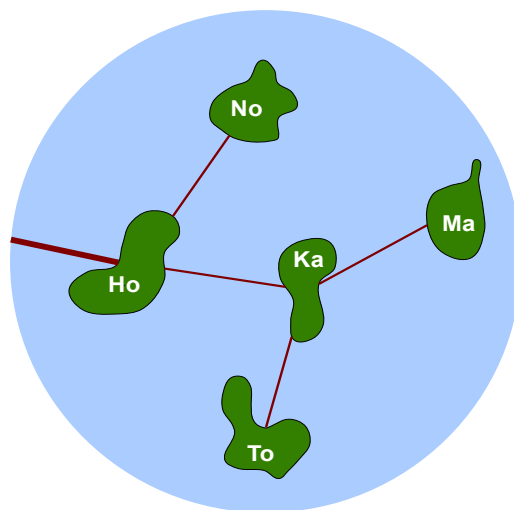
KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

A Honomakato szigetcsoport öt szigetből áll: Ho, No, Ma, Ka és To.

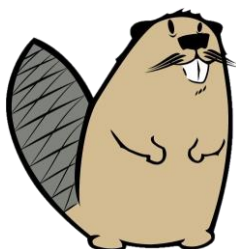
A Ho fősziget üvegszálas kábellel csatlakozik az internethez. Ezen kívül optikai kábelek kötik össze a következő szigetpárokat: Ho és No, Ho és Ka, Ka és Ma, illetve Ka és To. Így minden kábel összeköttetésben áll a Ho szigettel és csatlakoztatva vannak az internethez.

Honomakato lakosai szeretnék stabilabbá tenni a kapcsolatot azáltal, hogy ha egy optikai kábel megsérülne, akkor is kapcsolatban maradjon minden sziget.



Az alábbi kábelpárok melyike teszi stabilabbá a kapcsolatot?

- A) A Ho-To és No-Ma kábelek.
- B) A Ho-To és Ma-To kábelek.
- C) A Ka-No és No-Ma kábelek.
- D) Két kábel nem elég a hálózat stabilá tételéhez.



„A” válasz a helyes: a Ho-To és No-Ma kábelek.

Ha van egy Ho-To kábel,

- * To csatlakoztatva van, ha a Ho-Ka vagy a Ka-To kábel elszakad. Továbbá,
- * Ka is csatlakozik To-n keresztül, ha a Ho-Ka kábel szakad el;

Ha van egy No-Ma kábel,

- * Ma csatlakozik No-n keresztül, ha a Ho-Ka vagy a Ka-Ma kábel elszakad;
- * No csatlakozik Ka-n és Ma-n keresztül, ha a Ho-No kábel elszakad;
- * Ka csatlakozik No-n és Ma-n keresztül, ha a Ho-Ka kábel elszakad.

A Ma-To kábel nem védi No-t a lecsatlakozástól, ha a Ho-No kábel elszakad.

A Ka-No kábel nem védi To-t a lecsatlakozástól, ha a Ho-Ka vagy a Ka-To kábel elszakad.

MIÉRT INFORMATIKA?

A Honomakato szigetcsoporton épített kábelhálózat nem csak az egész internet egy kis része. A felépítése az internet globális hálózatára példa: Minden számítógépet, mobiltelefont, televíziót, hűtőt és bármit, amit manapság az internetre csatlakoztatunk, tekinthetünk egy-egy hálózati pontnak, mint a Honomakatao-hálózat szigeteit.

Az 1960-as években az internet megalkotásának egyik fő mozgatórugója egy ilyen stabil hálózat létrehozása volt. Bizonyos hálózati pontok közötti kapcsolódási hibák nem okozhatják az egész hálózat meghibásodását. A legtöbb más típusú hálózatnál, mint a közlekedési hálózat, ellátóhálózat, stb., ugyanilyen fontos, hogy ne lehessen egyetlen pont egy teljes meghibásodás okozója.

A számítógéptudománnyal foglalkozók felhasználják a gráfelméleti ismereteket, amikor hálózatokat vizsgálnak. Egy gráf egy pontokból és a pontok közötti összeköttetésekből (élek) áll. A gráfot összefüggőnek mondjuk, ha bármelyik pontból elérhetjük az összes többi. A gráf olyan éleit, melyek szükségesek ahhoz, hogy a gráf összefüggő maradjon, „hídnak” nevezzük. A stabil hálózatokban nem lehetnek ilyen hidak. szerencsére a számítógéptudomány ismer eljárásokat arra, hogy megtalálja a hidakat. Robert Tarjan számos gráf-algoritmus mellett egy hatékony híd-kereső algoritmust is kitalált.

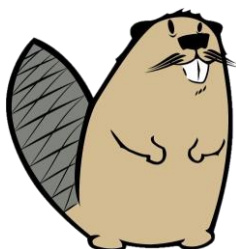
KULCSSZAVAK

Dinamikus adatstruktúra, gráf, híd (informatika)

WEBOLDALAK

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

<https://hu.wikipedia.org/wiki/Gráf>



NAGYMAMA LEKVÁRJA (2017-RU-04)

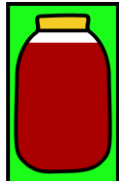
BERNJAMIN – NEHÉZ

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

Nagymama lekvárt főz, melyet Anna, Peti és Liza üvegekbe tölt.

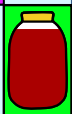
Ehhez mindig az alábbi lépéseket kell ismételniük, szigorúan az adott sorrendben:

		
Üveg elmosása	Megtöltés lekvárral	Üveg lezárása
Szükséges idő: 3 perc	Szükséges idő: 2 perc	Szükséges idő: 1 perc

Anna, Peti és Liza fel szeretné osztani a munkát, ezért egy tervet készítenek.

Amire figyelniük kell: egy lépés csak akkor kezdhető el, ha az előtte lévő befejeződött. Egy üvegnek tisztának kell lennie, mielőtt lekvárt töltünk bele, és az üveg csak akkor zárható le, ha teletöltötték lekvárral.

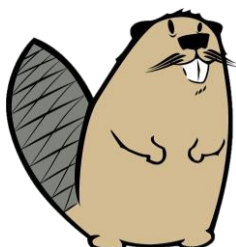
Ez a terv nem megvalósítható:

ANNA				
PETI				
LIZA				

Anna, Peti és Liza **10 perc alatt** a lehető legtöbb üveget szeretnék megtölteni.

Hány üveget tudnak megtölteni a gyerekek, ha jól terveznek?

- A) 4 üveget B) 5 üveget C) 6 üveget D) 7 üveget



„B” válasz a helyes: 5 üveget

Az alábbiakban egy lehetséges megoldási tervet láthatunk, amellyel Anna, Peti és Liza 10 perc alatt 5 üveget tud megtölteni.

ANNA					
PETI					
LIZA					

Más tervek is léteznek, amellyel ez ugyancsak kivitelezhető. Ezekben a tervekben a 3 sor, tehát a munkafolyamat lépései, melyet a gyerekeknek külön-külön el kell végezniük, éppúgy, mint a megadott ábrán, de más személyhez rendelődnek.

A gyerekek 10 perc alatt 5-nél több üveget nem tudnak megtölteni. Egy lekváros üveg megtöltése 6 percet vesz igénybe, mert $3 + 2 + 1 = 6$. Mindegyik gyereknek 10 perc áll a rendelkezésére, ezáltal az összes munkaidő 30 perc. Lehetetlen 10 perc alatt 5-nél több üveget megtölteni, mert $30/6=5$. Öt üveg tehát az elméleti optimális érték, melyet az ábrán látható megoldás is szemléltet.

MIÉRT INFORMATIKA?

Ebben a feladatban Anna, Peti és Liza egyszerre dolgozik, azaz az egyes munkafolyamatok párhuzamosan zajlanak. A párhuzamos feldolgozás megtalálható az informatikában is, például a nagymennyiségű adatok feldolgozása adatközpontokban egyidejűleg, párhuzamosan történik. Ebben az esetben, éppúgy, mint a megoldott feladatban, az egyes munkafázisokat ésszerűen kell felosztani. Egy darab számítógépben vagy okostelefonban is több adatfeldolgozó egység (processzormag) található, amelyek párhuzamosan dolgozzák fel a munkafolyamat egyes lépéseit. Az informatikában ütemezésnek (angolul schedulingnak) is nevezik annak a munkatervnek az elkészítését, amellyel a lehető legtöbb feladat befejezhető.

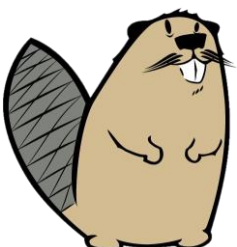
Az informatikán kívül is találhatóak hasonló problémák. Nagy projektek lebonyolítási tervéhez hálódigramot használnak. A hálódigramra épülő tervek meghatározzák, hogy az egyes munkafolyamatokat milyen sorrendben kell végrehajtani, s ezek a folyamatok milyen hosszúak. Az informatikai ismeretek tehát akkor is segítenek, ha nem is informatikusok vagyunk.

KULCSSZAVAK

Hálózati terv, algoritmus, párhuzamos folyamatok, projektmenedzsment

WEBOLDALAK

http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2010-0019_Projektmenedzsment/ch29.html
<https://hu.wikipedia.org/wiki/Projektmenedzsment>



KÖRFORGÓVÁR (2017-SI-02)

BENJAMIN – NEHÉZ

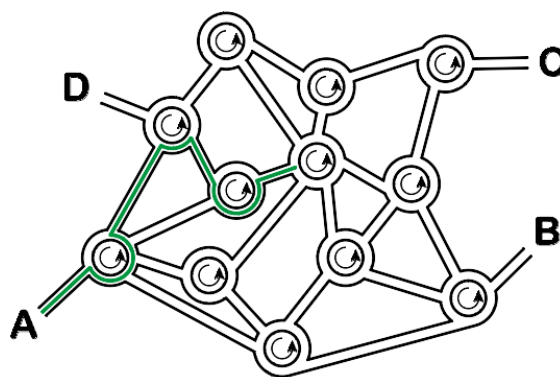
JUNIOR – KÖNNYŰ

Körforgóváron minden kereszteződésben csak körforgalmak találhatóak. Itt a navigációs rendszer nem ad olyan utasításokat, mint

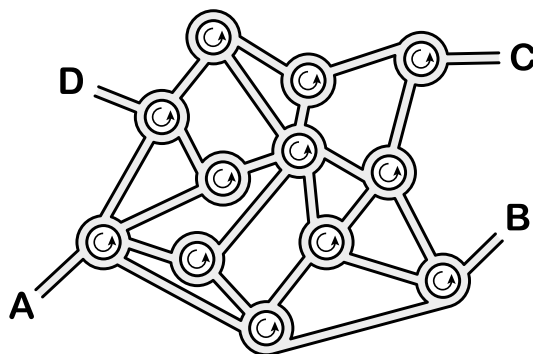
- a következő körforgalomnál hajts ki a 4. kijáraton, majd
- a következő körforgalomnál hajts ki az 1. kijáraton, majd
- a következő körforgalomnál hajts ki a 2. kijáraton;

helyette számok sorozatát adja meg, mint "4 1 2".

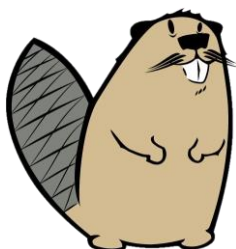
Ez jelenti a képen a zölddel jelölt útvonalat.



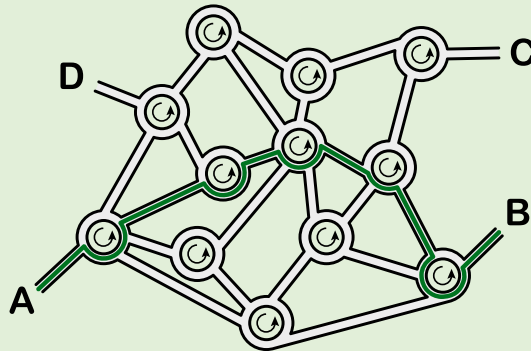
Ha A-ból indulunk és a "3 1 3 2 3" sorozatot követjük, hova fogunk érkezni az utunk végén?



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



„B” válasz a helyes:



MIÉRT INFORMATIKA?

Ez a feladat jó példa az információ formalizálására. Egy navigációs rendszernek szükséges az útvonalterv általa érthető nyelvre történő lefordítása. Az útvonaltervet számsorral való átkódolása, ahogy a feladatban láttuk, az információ olyan formába öntése, melyet a számítógép utasítások sorozataként tud értelmezni. Az ilyen utasítássorozatot szekvenciának nevezzük, és ez a programozási nyelvek alap építőköve.

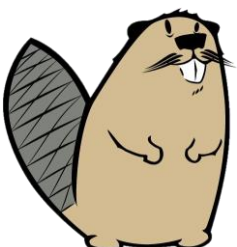
Ebben a példában könnyítés, hogy az úthálózat egyformán kialakított: minden kereszteződés körforgalom. Az ilyen azonos típusú elemekből álló struktúrát homogénnek nevezzük, míg a különböző elemekből állót heterogénnek. A homogén struktúrákat a számítógépes programok könnyebben feldolgozzák, ezért az informatikában előnyben részesítjük őket, ha csak nincs nyomós okunk másképp tenni.

KULCSSZAVAK

Utasítássorozat, programvégrehajtás

WEBOLDALAK

<http://wiki.prog.hu/wiki/Utasítás>
[https://hu.wikipedia.org/wiki/Kód_\(informatika\)](https://hu.wikipedia.org/wiki/Kód_(informatika))

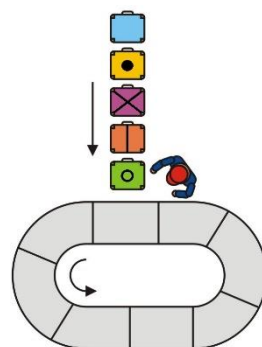


REPÜLŐTÉR (2013-AT-04)

BENJAMIN – KÖZEPES

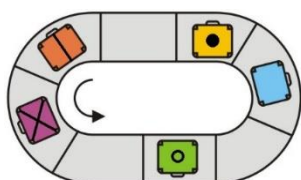
KADÉT – KÖZEPES

A repülőtéri szállítószalag a nyíl irányába forog körbe és 8 hely van rajta. Egy dolgozó 5 bőröndöt rak sorban a szállítószalagra. A következő bőröndöt mindig a harmadik szabad helyre rakja, tehát a már foglalt helyeken kívül két szabad helyet hagy tovább fordulni.

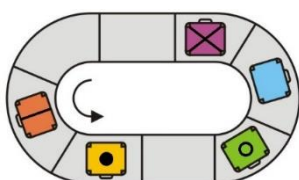


Akkor van kész, ha mind az 5 bőröndöt felrakta a szállítószalagra.

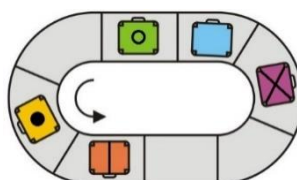
Hogy néz ki a szállítószalag a munka végeztével?



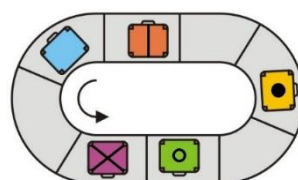
A)



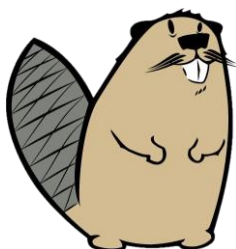
B)



C)



D)



„B” válasz a helyes:

Elsőként a körrel jelölt bőrönd kerül fel a szállítószalagra. Utána 3 hellyel hátrébb a függőleges vonallal, újabb 3 hellyel hátrébb pedig a kereszttel jelölt bőrönd. Ezután a ponttal jelölt bőröndnek a harmadik szabad helyre kell kerülnie. Mivel azonban a körrel jelölt bőrönd ekkorra már körbeért, a harmadik szabad hely a kereszttel jelölt bőröndtől számított negyedik hely. Utoljára a jelöletlen bőrönd kerül fel a szállítószalagra. Ennek a függőleges vonallal jelölt bőröndöt, két szabad helyet és a kereszttel jelölt bőröndöt kell elengednie.

MIÉRT INFORMATIKA?

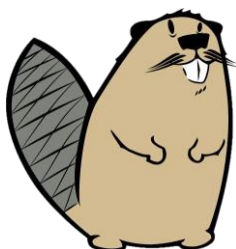
Ha a táskák végrehajtandó műveleteket, a szállítószalagon lévő helyek pedig szabad kapacitást jelölnének, a feladat arról szólna, hogy hogyan lehet a végrehajtandó műveleteket a számítógép szabad teljesítményének függvényében elosztani (angolul „scheduling”). A műveletek elosztását bonyolítja, hogy a feldolgozási idejük és a tárolásukhoz szükséges hely változó. Ráadásul a műveletek nem függetlenek egymástól, és különböző részrendszerekre, ill. csatlakozó eszközökre van szükségük, amelyek gyakran csak korlátozott számban állnak rendelkezésre. Érdeemes megjegyezni, hogy az itt ábrázolt elosztás – minden harmadik szabad helyre kerül egy elem – nem a leghatékonyabb.

KULCSSZAVAK

Algoritmus, eljárás, modulus, periódus

WEBOLDALAK

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

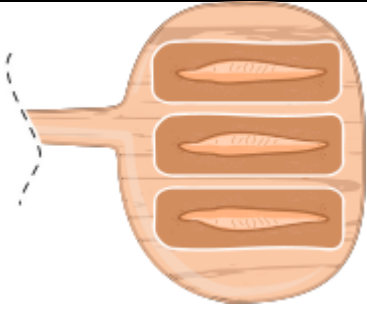
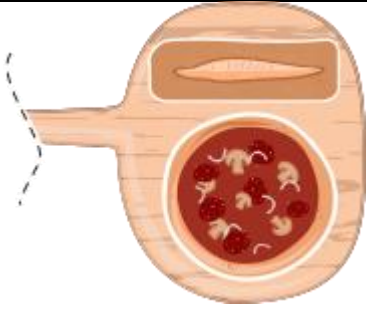
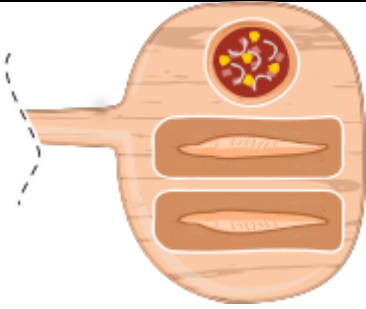


BIBÉRIA PIZZÁZÓ (2017-CH-03)

KADÉT – KÖZEPES

A Bibéria Pizzériának csak egy kemencéje van, így egyszerre csak kevés ételt tudnak elkészíteni.

Lehetséges kombinációk:

		
3 ciabatta kenyér	1 ciabatta kenyér 1 nagy pizza	2 ciabatta kenyér 1 kis pizza

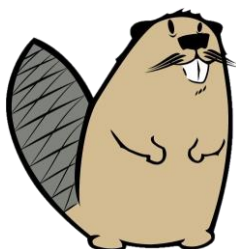
Az ételek sütési ideje különböző: egy kis pizzának 10 percet, egy nagy pizzának 15 percet, egy ciabatta kenyérnek pedig 20 percet kell a kemencében töltenie. A szakács addig nem veheti ki az ételleket, amíg meg nem sülték.

A mai nap igen forgalmas: egy kis pizzát, két nagy pizzát és négy ciabatta kenyeret rendeltek a vendégek.

Igencsak éhesek, így a lehető leghamarabb szeretnék megkapni a rendelésüket.

Hány perc alatt készülhet el leggyorsabban az összes rendelés?

- A) 40 perc
- B) 50 perc
- C) 60 perc
- D) 80 perc

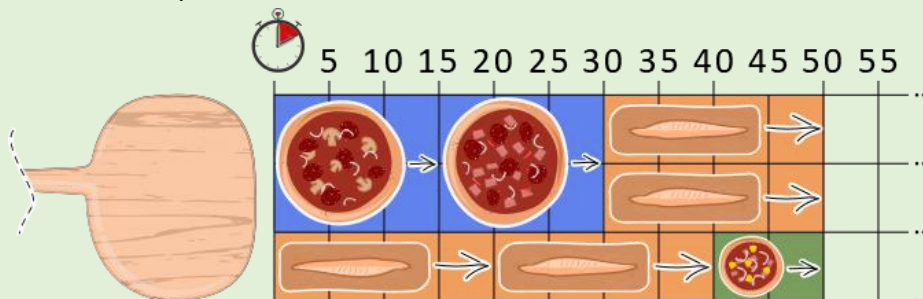


HÓDÍTSD MEG A BITEKET – 2017-ES FELADATSOR

BIBÉRIA PIZZÁZÓ (2017-CH-03)

„B” válasz a helyes:

Több optimális megoldás létezik. Legegyszerűbben az alábbi módon adható meg egy ilyen megoldás: a két nagy pizzát egymás után süti meg a pizzaszakács, melyekkel egy időben egy-egy ciabatta kenyér is sül. A két nagy pizza elkészülése után bekerül az utolsó két ciabatta kenyér, melyhez az előző adag ciabatta kenyér megsülése után a kis pizza társul:



Ez a megoldás 50 percet igényel. Gyorsabban nem lehet kisütni ezeket, mert ezalatt az 50 perc alatt végig teljesen foglalt a kemence.

MIÉRT INFORMATIKA?

Amikor megpróbálunk a korlátozott erőforrások (jelen esetben a kemence) kihasználására egy ütemtervet készíteni, akkor rendszerint az optimális megoldást szeretnénk megkapni. Gyakran fordul elő, hogy költségeket, vagy jelen esetben magát az időt kell optimalizálni. Ha jól használjuk ki az erőforrást (kemence), akkor a legrövidebb sütési időt kapjuk eredményül.

Az optimális ütemezés széles körben elterjedt megoldási folyamatát angolul Round-Robin-nak nevezik. Lényege, hogy az összes folyamat, amelynek le kell zajlania, egy várólistán szerepel. Az erőforrás pedig egy kis időre mindegyik folyamat rendelkezésére áll. Ha a folyamat ez alatt az idő alatt nem fejeződik be, akkor visszakerül a várólistára.

Ebben a feladatban nincs sok értelme ennek a stratégiának, ugyanis senki sem vesz ki félkész pizzát a kemencéből, hogy később visszategye, így a mohó stratégiával kell dolgoznunk: kezdjük a sütést a legnagyobb pizzával (a nagy pizzák egyikével, mert ez foglalja a legtöbb helyet), és próbáljuk meg vele együtt kisütni a hozzá méretben legközelebb állót, és így tovább. Mivel a kemencében egyszerre több étel is lehet, az elkészült étel mindig kivehető és a következő elkészítendő étel betehető a helyére.

Megtaláltuk végül a legjobb megoldást? Ez nem nyilvánvaló, az ember találhat egy jobb megoldást, amelyben pl. három ciabatta kenyérrel kezd.

Azonban ha ezt választjuk, minden erőforrás mindig teljesen ki van használva (tehát ebben az esetben a kemence mindig tele van), biztosak lehetünk abban, hogy nincs jobb megoldás

KULCSSZAVAK

Ütemezés, erőforrás, várakozás, mohó algoritmus

WEBOLDALAK

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

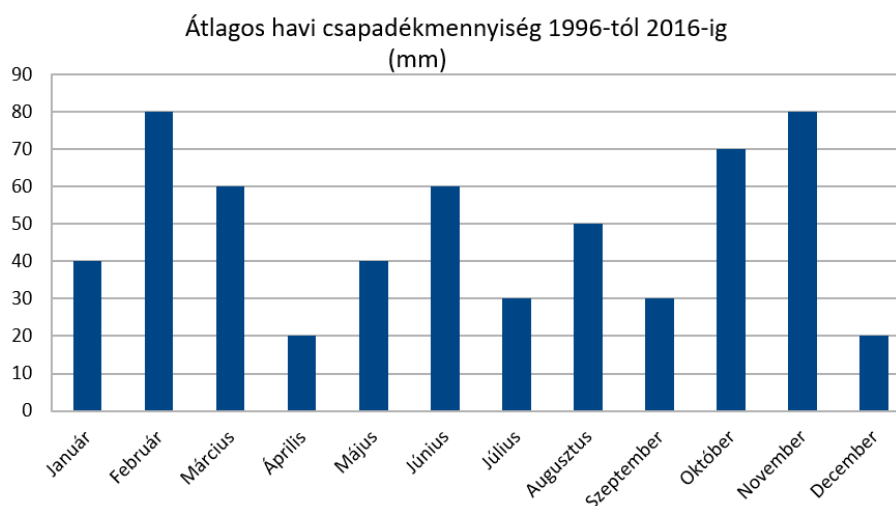
TERHELÉSI TESZT (2017-HU-04)

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

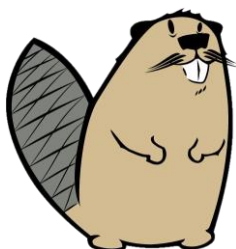
Egy új gátat terveznek. A mérnökök tanulmányozták az elmúlt húsz év csapadékeloszlását (eső, hó, köd), hogy megtalálják a legjobb hónapot az építkezéshez. A következő feltételekben értenek egyet:

- Az egyik legcsapadékosabb hónapban kell a maximális terhelést tesztelni, mielőtt a gát megnyitását engedélyeznék.
- A gátat egy vagy két hónappal a teszt előtt kell megépíteni.
- Az építkezés hónapja a lehető legszárazabb legyen



Melyik hónapot válasszák a mérnökök (az építkezés elkezdéshez)?

- A) Január
- B) Április
- C) Szeptember
- D) December



„D” válasz a helyes:

Először egyeztessük a legcsapadékosabb hónapot (teszt időpontja): februárban és novemberben esik a legtöbb csapadék, ezért ezek egyikében kell tesztelni.

Mivel az építkezés egy vagy két hónappal a tesztelés előtt történhet, négy hónap jöhet szóba: február előtt január és december, illetve november előtt szeptember és október. Ezek közül a legszárazabb hónap december.

MIÉRT INFORMATIKA?

Számos példát láthatunk arra, hogy adatelemzéssel keressük meg a legjobb értéket:

- megtalálni a legkedvezőbb éjjel-nappali benzinkutat,
- megtalálni az interneten a legkedvezőbb árú HD-kamerát, amihez visszavásárlási és gyártói garancia jár,
- megtalálni a legjobb helyet egy szélkerék építéséhez.

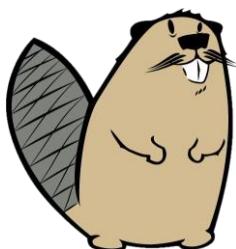
A feladatban egy kisebb adathalmazt diagramon ábrázoltunk. Így lehetővé válik az adatok elemzése, egyszerű átlátása és könnyebben megtaláljuk a feltételeknek megfelelő értéket (jelen esetben időpontot a gátépítésre). Nagyobb adathalmazokat, mint például egy HD-kamera keresése esetén adódó árlista, számítógéppel hatásosabban tudnánk kiértékelni.

KULCSSZAVAK

Adatelemzés, diagram, oszlopdiagram

WEBOLDALAK

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



FRISSÍTŐ (2017-CA-07)

KADÉT – NEHÉZ

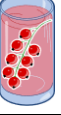
JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Négy barát kirándulni ment. Útközben megálltak egy italboltnál, hogy felfrissüljenek kicsit. Mindenkinek megvan a maga ízlése. Az előnyben részesített innivalók a táblázatban láthatók. A szívek száma azt jelenti, hogy mennyire szívesen fogyasztaná azt az adott illető. Például Anna a

körtelevet () szívesen inná (3 szív), ribizli szörpre () most azonban nem vágyik (1 szív).

Dániel nagyon szívesen inná ribizli szörpöt () (4 szív), körtelevet () azonban nem (1 szív).

				
Anna				
Bea				
Krisztina				
Dániel				

A bolt nagyon népszerű, így mind a négy italból már csak egy-egy maradt.

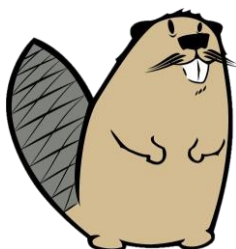
Milyen ital jut Annának, ha azt szeretnénk, hogy a szívek száma a lehető legmagasabb legyen?

A) Almalé 

B) Körtelevé 

C) Narancs szörp 

D) Ribizli szörp 



„B” válasz a helyes: Körtelé

A maximálisan megszerezhető szívek száma 14, melyet például ez a megoldás mutat:

				
Anna				
Bea				
Krisztina				
Dániel				

Ahhoz, hogy ezt a megoldást kapjuk, ideális, ha Dániel választásából indulunk ki. Ő a ribizli szörpöt szereti leginkább (4 szív), amit azonban a többiek nem (mindenhol 1 szív). Ha Beának vagy Krisztinának az almalevet adjuk, akkor az utoljára maradt két személy (Anna és Krisztina, másik esetben Anna és Bea) a második kedvenc szörpjüket ihatják.

A négy barát közül hárman leginkább almalevet innának. Mivel azonban csak egy van belőle, két személynek muszáj a második kedvencével beérnie, ezért a $3+3+4+4=14$ szívnél nem gyűjthető több.

Csak ezeknél a választásoknál lehetséges 14 szívet összegyűjteni, mert minden további választásnál legalább egy személynek a harmadik kedvencét kellene fogyasztania, így az összegyűjtött szívek száma $2+3+4+4=13$ lenne.

MIÉRT INFORMATIKA?

Ez a feladat arról szól, hogy miként lehet elérni a lehető legtöbb szívet, ezáltal a lehető legmagasabb, azaz optimális elégedettségi szintet. Az optimalizálás az informatika és a matematika fontos kutatási területe. Sokszor kerül elő, ugyanis gyakran sok időbe telik megtalálni egy feladat optimális megoldását. Ebben az esetben egy egyszerű algoritmusnak, amely minden lehetséges és lehetetlen megoldás megvizsgál, több, mint 65000 különböző lehetőséget kellene kipróbálnia. Egy kis gondolkodással azonban jelentősen csökkenthető ez a szám (csak 24 lehetséges megoldás van, amire a szívek számát ki kell számolni), azonban nem mindig könnyű rálelni a gondolatmenetre.

Jelen esetben a párosítás különleges módja jelenti a nehézséget: a négy barát közül mindenkinek pontosan egy szörpöt szabad rendelni úgy, hogy minden italból pontosan egy van.

Hasonló problémák a való életben is felmerülnek, elég csak szervtranszplantációs várólistákra gondolni. Itt egy páciensnek egy szervet adnak úgy, hogy egyidejűleg figyelembe veszik a korlátozó feltételeket (pl. várakozási idő, átültetés sürgőssége, befogadás).

KULCSSZAVAK

Optimalizálás

WEBOLDALAK

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

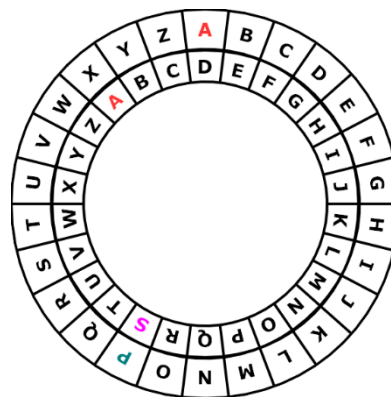
TITKOS RENDELÉS (2017-CH-04)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

A Hódügynökségnél dolgozó Anna titkos üzenet formájában rendel egy étteremben.

A rendelést csak Cézár, a szakács érti. A kódoláshoz egy kicsi és egy nagy, egymásba illeszthető korongot használ, melyen az angol ABC betűi vannak. Alaphelyzetben a belső korong „A” betűje a külső korong „A” betűje alatt áll, a belső „B” betű a külső „B” alatt és így tovább.



A titkosítás módja a következő: Anna először felírja a rendelést, pl. PIZZA. Ezután minden betűre az alábbi műveleteket hajtja végre:

1. A betű alá egy léptetési számot ír.
2. Alaphelyzetbe állítja a belső korongot, majd azt léptetési számnak megfelelően az óramutató járásával ellentétes irányba lépteti.
3. A kódolt üzenetbe azt a betűt írja, ami a rendelés betűjéhez illeszkedik.

Példa: Ha Anna a PIZZA szót szeretné lekódolni és a forgatási számok 3, 1, 4, 3 és 1, akkor a kódolt üzenet SJDCB lesz. A képen azt látod, ahogy a két korong 3-as léptetésszám után áll: a P-hez az S illeszkedik.

Rendelés	P	I	Z	Z	A
Léptetésszám	3	1	4	3	1
Kód	S	J	D	C	B

Egy másik rendeléshez Anna forgatási számai: 3, 5, 1, 7, 2, 4 és 8, a kód pedig SFDHN.

Rendelés	?	?	?	?	?
Léptetésszám	3	5	1	7	2
Kód	S	F	D	H	N

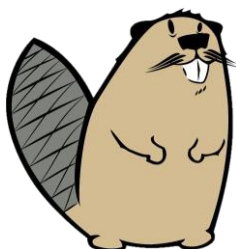
Mit rendelt Anna?

A) HAMBI

B) PACAL

C) FAGYI

D) PARAJ



„B” válasz a helyes:

Rendelés	P	A	C	A	L
Léptetésszám	3	5	1	7	2
Kód	S	F	D	H	N

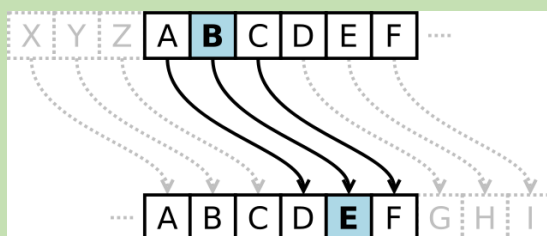
A rendelés megadható a korongok segítségével. A kód minden egyes betűjétől lépünk léptetésszámnak megfelelően az óramutató járásával ellentétesen, így megkaphatjuk a rendelés adott betűjét.

Pl. S-től 3 lépésre óramutatóval ellentétes irányba a P van, F-től 5 lépésre az A stb. A helyes megoldás tehát PACAL.

MIÉRT INFORMATIKA?

Anna titkosítja a rendelését, hogy azt csak a kedvenc szakácsa értse meg. A titkosítás az emberiség egyik legrégebbi vágya. Mindig akadt indok arra, hogy úgy váltsunk üzenetet, hogy csak a fogadó értse meg. Nagyon sok különböző titkosítási folyamat létezik, mindegyikhez 2 algoritmus tartozik. Az egyik titkosításra, a másik pedig a titkosított szöveg visszafejtésére. Mindkét eljárásnak ismernie kell az üzenet titkosítási kulcsát.

Az egyik legrégebbi titkosítási eljárás Julius Caesarhoz kötődik: a titkosítás kulcsa egy szám, amely a betűk ABC szerinti elmozdítását adja meg. A 3-as kulcs például azt jelenti, hogy az eredeti üzenet A betűjét D-vel, a B-t E-vel (stb.) titkosítjuk. A kódolás-dekódolás menetében segíti a „Caesar-korong”, melynek működését a feladatban részleteztük.



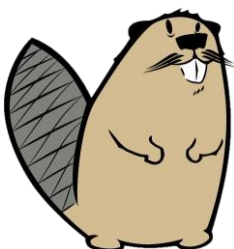
Az egy titkosítási kulcsot használó folyamatok azonban nem biztonságosak. Ezt feltehetően Anna is tudja, mert minden betűhöz más kulcsot (jelen esetben más számot) használ, Vigenère titkosítási folyamatához hasonlóan. Ennél a folyamatnál a forgatási számok hosszú üzenet esetén ismétlődnek, hogy a titkosítási kulcs ne legyen túl hosszú.

KULCSSZAVAK

Titkosítás, kriptográfia, polialfabetikus kód, Caesar-rejtjel, Vigenère-rejtjel

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Kriptográfia>
<https://hu.wikipedia.org/wiki/Caesar-rejtjel>
<https://hu.wikipedia.org/wiki/Vigenère-rejtjel>



RIADÓ A MÚZEUMBAN! (2017-DE-03)

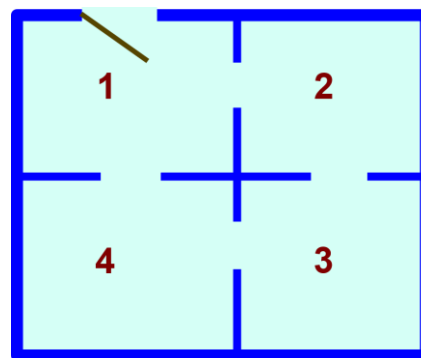
KADÉT – NEHÉZ

A Fafaragványok Múzeumában intelligens biztonsági rendszer működik. A rendszer minden percben megállapítja, hány személy tartózkodik az egyes termekben és a számokat egy táblázatban rögzíti. A táblázat adataiból a rendszer megállapítja, hogy van-e behatoló a múzeumban.

A behatoló olyan személy, aki a múzeumban tartózkodik, de nem használta a bejáratot.

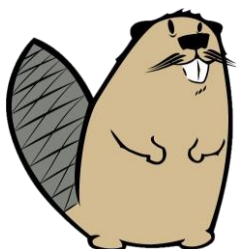
Amint a rendszer behatolót észlel, beindítja a riasztót. Balra látható a rendszer 10:01 és 10:07 között készített táblázata. Jobbra a múzeum alaprajza látható a termek számával.

Idő	Terem 1	Terem 2	Terem 3	Terem 4
10:01	2	0	0	0
10:02	3	0	0	0
10:03	2	1	0	0
10:04	4	1	1	0
10:05	2	2	3	0
10:06	5	2	2	1
10:07	4	1	2	2



A táblázatban szereplő percek egyikében megszólal a riasztó. Melyikben?

- A) 10:03
- B) 10:04
- C) 10:05
- D) 10:06



„C” válasz a helyes:

A rendszer 10:05-kor riaszt. Ebben az időpontban a Terem 3-ban két személlyel több volt, mint az előtte levő percben. A szomszédos termekben akkor (vagyis 10:04-kor) csak egy személy volt, nevezetesen a Terem 2-ben. Következésképp valaki kívülről mehetett csak a Terem 3-ba, anélkül, hogy használta volna a múzeum bejáratát.

Más időpontokban nem fedezünk fel behatolót:

10:01-kor 2 személy van a Terem 1-ben, ők használhatták a múzeum bejáratát.

10:01-től 10:02-ig 1 személy bejöhett a múzeum bejáratán.

10:02-től 10:03-ig 1 személy átmehetett a Terem 1-ből a Terem 2-be.

10:03-tól 10:04-ig 1 személy a Terem 2-ből a Terem 3-ba és 1 személy a Terem 1-ből a Terem 2-be mehetett és 3 személy a bejáraton keresztül érkezett.

10:05-től 10:06-ig 1 személy a Terem 3-ból a Terem 4-be ment és 3 személy bejött a múzeumba.

10:06-tól 10:07-ig 1 személy a Terem 1-ből a Terem 4-be és 1 személy a Terem 2-ből a Terem 1-be mehetett át, miközben 1 személy a bejáraton keresztül elhagyta.

MIÉRT INFORMATIKA?

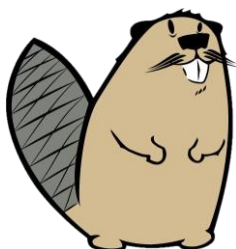
Olyan biztonsági rendszert, mely fontos (kritikus) területen az ott tartózkodó személyek számát felügyeli, például reptereken találhatunk. A felügyeleti kamerák élőképet a számítógépprogram feldolgozza, felismeri a személyeket, megszámlolja azokat és kiértékeli a számokat. Ezek a programok mesterséges intelligenciát alkalmaznak, például az emberi arcok felismerésére. Ahhoz, hogy a biztonsági problémákat észleljék időnként olyan egyszerű szabályokat is definiálnak és figyelnek, mint ebben a feladatban.

KULCSSZAVAK

Intelligens biztonsági rendszer, szabály, felügyeleti eszköz

WEBOLDALAK

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



LABDÁK (2017-RS-02)

KADÉT – NEHÉZ

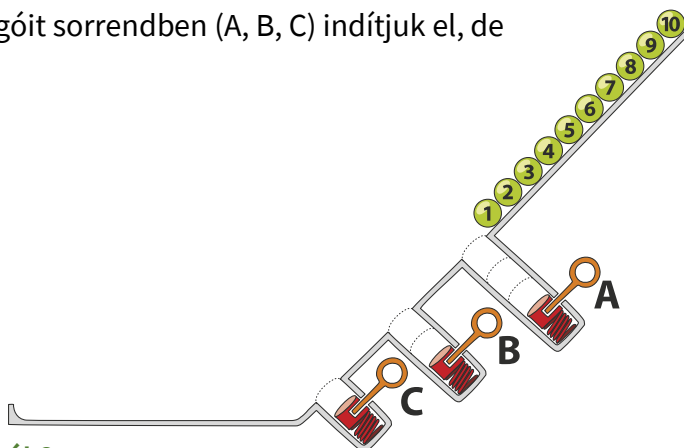
JUNIOR – KÖZEPES

SENIOR – KÖZEPES

Számozott golyók gurulnak le egy rámpán. A golyók sorrendje aszerint változik, ahogy beesnek lyukakba. Ha egy golyó egy lyukhoz ér, amelyben van még hely, akkor a golyó beleesik. Máskülönben a golyó átgurul a lyuk fölött. A lyuk alján elhelyezett rugókkal kilőhetők a lyukba begurult golyók. Például:

5 golyó előtt kezd gurulni	Miután a golyók megálltak	A rugó indítása utáni végső helyzet

Tíz golyó gurul az alábbi rámpán. A három lyukba (A, B, C) 3, 2 és 1 golyónyi hely van, ahogy az ábra is mutatja. A lyukak rugóit sorrendben (A, B, C) indítjuk el, de csak azután, hogy minden golyó megállt.



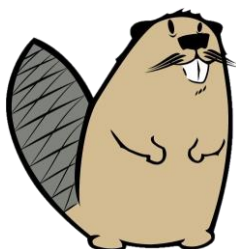
Milyen sorrendben állnak legvégül a golyók?

A)

B)

C)

D)



„D” válasz a helyes: 

Az „A” lyukba 3 golyó fér, vagyis a 4-től 10-ig számozott golyók átgurulnak fölötte. A „B” lyukba 2 golyó fér, vagyis a 6-tól 10-ig számozott golyók gurulnak át fölötte. A „C” lyukba csak 1 golyó esik bele, tehát a 7-től 10-ig számozott golyók ilyen sorban gurulnak végig. (A C választ kizárhatjuk.)

Ezután kilökjük a golyókat az „A” lyukból a következő számozási sorrendben: 3, 2, 1. Ezek a golyók le is gurulnak alulra. Ekkor már a 7, 8, 9, 10, 3, 2, 1 golyósorozatunk végleges. (A-t és B-t is kizárhatjuk).

A „B” lyukban levő golyók kilökésekor az 5-ös, majd a 4-es golyó is legurul a rámpa aljára, végül pedig a „C” lyukból kilökjük az utolsó, 6-os golyót is, ezáltal a D válaszban mutatott 7, 8, 9, 10, 3, 2, 1, 5, 4, 6 sorrend adódik, tehát a D valóban helyes válasz.

MIÉRT INFORMATIKA?

A lyukak vermekként működnek, mely egy adatszerkezet, vagyis adatok egy rendezési módja. Ez az „utolsó-be első-ki” szabályon alapszik (LIFO, Last-In, First-Out). Például, az első golyó, amelyik beleesik egy lyukba, kerül utoljára kilökésre. Annak ellenére, hogy ez egy nagyon egyszerű ötlet, számos helyzetben hasznos.

Segítségével megvizsgálható például, hogy a zárójelek a $((1+2)*3)$ kifejezésben lezártak, viszont a $((4+5)*(6-7))$ kifejezésben nem. Az ötlet az, hogy minden kezdőzárójelet egy veremben helyezünk el (úgynevezett push műveletet hajtunk végre), majd a hozzátartozó záró-zárójelnél kivesszük onnan (pop művelet).

Ahelyett, hogy az adatokat kifinomultabb adatszerkezetbe rendeznénk vagy kitalálnánk egy bonyolultabb algoritmust, látható, hogy a feladat megoldásához elégséges az „utolsó-be első-ki” szabály alkalmazása.

KULCSSZAVAK

Verem, LIFO

WEBOLDALAK

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

HELYETTESÍTÉS (2017-SI-06)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

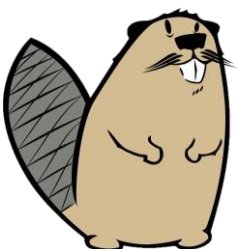
Ma úr, egy távol-keleti vállalat egyik munkatársa, hirtelen lebetegedett, ezért minden feladatát MaMa úr vette át.

Szerencsére Ma úr 2 hét múlva meggyógyult és újra munkába állt. MaMa úr azonban olyan jól besegített neki, hogy megbeszélték, Ma úr átveszi MaMa úr korábbi munkáját, MaMa úr pedig Ma úrét.

A projektdokumentációt úgy kell módosítaniuk, hogy a szövegben a Ma név helyett MaMa szerepeljen, és fordítva. A felhasznált projekttervben tetszőleges karaktersorozat tetszőleges karakterekkel helyettesíthető.

Melyik eljárási mód értelmes és célravezető, ha feltételezzük, hogy a szövegben sehol sem fordul elő „#” jel?

- A) Először minden „Ma”-t „MaMa”-ra cserélünk, majd minden „MaMa”-t „Ma”-ra.
- B) Először minden „MaMa”-t „Ma”-ra cserélünk, majd minden „Ma”-t „MaMa”-ra.
- C) Először minden „Ma”-t „#” jelre, majd minden „#”-t „MaMa-ra”, végezetül minden „MaMa”-t „Ma”-ra cserélünk.
- D) Először minden „Ma”-t „#” jelre, majd minden „MaMa”-t „Ma”-ra, végezetül pedig minden „#” jelet „MaMa”-ra cserélünk.



„D” válasz a helyes:

A: Ebben az esetben a szövegben csak „Ma” maradna, az összes „MaMa” eltűnne, mert az első cserélés után már csak „MaMa” maradna, amit végül az összes helyen „Ma”-ra cserélnénk

B: Ebben az esetben a szövegben csak „MaMa” maradna, az összes „Ma” eltűnne, mert az első cserélés után már csak „Ma” maradna, amit végül az összes helyen „MaMa”-ra cserélnénk.

C: Ebben az esetben a szövegben csak „Ma” maradna, az összes „MaMa” eltűnne. Miután az összes „Ma”-t „#” jelre, a „#”-t pedig azonnal „MaMa”-ra cserélnénk, végezetül az összes „MaMa”-t visszaírnánk „Ma”-ra, akkor csak „Ma” maradna.

D: Ez az egy eljárás mód működőképes, mert a „Ma” előfordulásokat „#” jelre cserélnénk, így megőrződnének, amíg a „MaMa”-t „Ma”-ra cseréljük. Végezetül már csak a „#” jelet kellene „MaMa”-ra cserélni.

MIÉRT INFORMATIKA?

Annak ellenére, hogy a csere egy egészen egyszerű folyamat, az informatikában fontos szerepet játszik. Sorozatos cserélésekkel komplex feladatokat lehet végrehajtani. Az elméleti informatikában is előfordul a csere, melyekkel például a formális nyelvtan cserélési szabályok listájaként felírható.

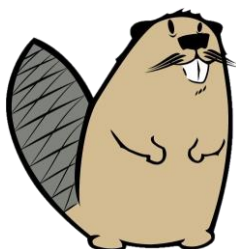
Ebben a feladatban az a nehéz, hogy két fogalmat kell egymással kicserélni. Ez csak akkor lehetséges, ha egy harmadik, egyébként nem használt fogalmat, mint segédfogalmat vezetünk be.

KULCSSZAVAK

Szövegszerkesztő, utasítássorozat, változók cseréje

WEBOLDALAK

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



TISZTELETBELI JAPÁN NÉV (2017-HR-02)

JUNIOR – KÖNNYŰ

SENIOR – KÖNNYŰ

Egy japán világjáró arról mesélt, hogy régi japán szokás szerint a keresztnévek betűiből tiszteletbeli japán neveket alkothatunk.

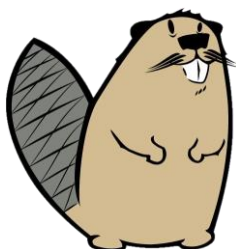
Mindössze annyit kell tennünk, hogy a keresztnévek betűit egy-egy szótagnak feleltetjük meg.

A → ka	D → te	G → ji	J → zu	M → rin	P → mor	S → si	V → ru	Y → fu
B → pi	E → ku	H → ri	K → me	N → to	Q → ke	T → chi	W → mu	Z → zi
C → mi	F → lu	I → ki	L → ta	O → mo	R → shi	U → do	X → na	

A világjáró magyar barátjának tiszteletbeli japán neve: "Rikushimedotakusi".

Mi a magyar barát keresztnéve?

- A) Herman
- B) Herold
- C) Herkules
- D) Herbert



HÓDÍTSD MEG A BITEKET – 2017-ES FELADATSOR

TISZTELETBELI JAPÁN NÉV (2017-HR-02)

„C” válasz a helyes: Herkules

Előlről haladva az első szótag a „Ri”, ami a H-nak feleltethető meg. Második szótag a „ku”, amely dekódolva „E”. Az összes szótag megfeleltetését az alábbi táblázat mutatja:

Ri	ku	shi	me	do	ta	ku	si
H	E	R	K	U	L	E	S

A feladat gyorsan megoldható, ha az utolsó szótaghoz (si) keresünk magyar megfelelőt (S). Egyedül a Herkules név végződik S-re.

A lehetséges nevek mindegyike az MTA Nyelvtudományi Intézete által anyakönyvi bejegyzésre alkalmasnak minősített utónevei közé tartozik 2017. szeptember 1-től.

MIÉRT INFORMATIKA?

Minden bizonnyal találkoztál már ehhez hasonló játékokkal, melyek titkos nyelvként is használhatóak. Az eljárás alapja, hogy egy betű egy másik betűvel vagy egy egyértelműen azonosítható szótaggal helyettesíthető. Ezt az informatikában betű-betű megfeleltetésnek vagy semi-thue rendszernek hívjuk.

Ezek a rendszerek nem alkalmasak a szövegek biztonságos titkosítására, mert a monoalfabetikus kódolást veszik alapul, így könnyen feltörhetők számítógép nélkül is.

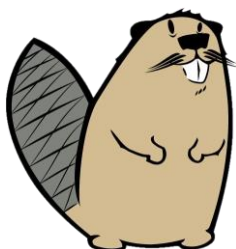
A megfeleltetési rendszerben szereplő szótagok nagyon hasonlítanak azokra, amelyeket Japánban is használnak, ezért hangzik japánosan a kódolt név.

KULCSSZAVAK

Betű-betű megfeleltetés, semi-thue rendszer, monoalfabetikus kódolás

WEBOLDALAK

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



AZ ÚJ DAL (2017-IR-06)

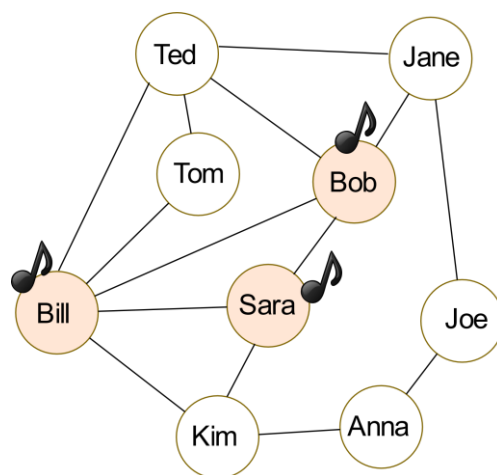
JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Az alábbi diagram azt mutatja, hogy az osztályban ki kinek a barátja. Két személyt csak akkor tekintünk barátnak, ha a nevük vonallal össze van kötve, különben nem.

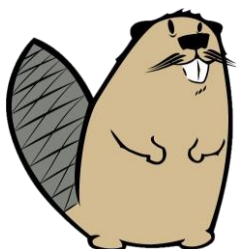
Hétfőn megjelent egy új lemez, melyet még ezen a napon Bill, Bob és Sara megvásárolt. Az ő nevüket hangjeggyel jelöltük.

Keddtől a következők történtek: Csak azok a személyek vásárolták meg a lemezt, akikre érvényes, hogy **barátainak legalább fele** előző nap rendelkezett vele. Kedden például Tom megvásárolta, azonban Jane már nem.



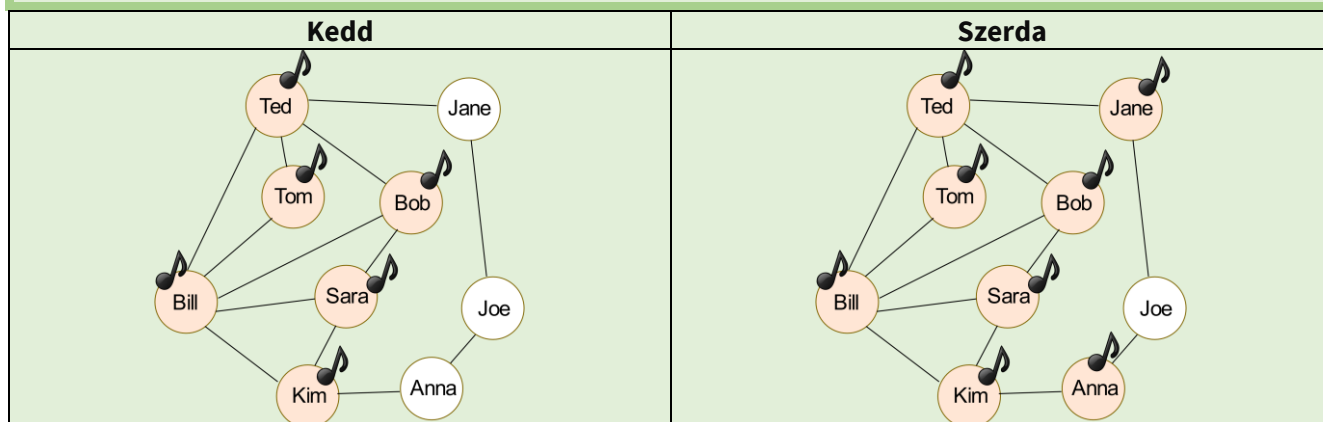
Legkorábban melyik nap vette már meg az osztály minden tagja a lemezt?

- A) Szerda
- B) Csütörtök
- C) Péntek
- D) Szombat



„B” válasz a helyes:

Tom, Ted és Kim kedden, Anna és Jane szerdán vásárolja meg a lemezt.



Joe csak csütörtökön tudja megvenni, így ezen a napon lesz meg mindenkinek az osztályból.

MIÉRT INFORMATIKA?

Az úgynevezett szociális hálózatoknak néha milliárd tagja van, ezért nem csak a felhasználók számára érdekesek, hanem termékek/szolgáltatások kínálóinak is. A marketingkampányok arra használják a szociális hálózatok(at) / hálózatok diffúziós folyamatát, hogy az új termékeket megismertessék a hálózat tagjaival. Erről a folyamatról szól ez a feladat.

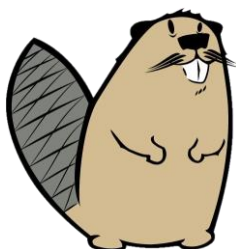
Jelen esetben egy hullámmodellt használunk, hogy a sláger terjedését szimuláljuk. A feladat diagramja egy gráf, a nevek a gráf pontjai, a „barátságvonalak” pedig a gráf élei. Két pont akkor szomszédos, ha egy él segítségével közvetlen kapcsolatban állnak egymással. A fertőzés hulláma, jelölése q , egy 0 és 1 közé eső szám, a feladatban $q=0,5$. Egy személy B viselkedést mutat (pl. megveszi a lemezt), ha szomszédjainak legalább q része ugyanezt a B viselkedést produkálja.

KULCSSZAVAK

Ismeretségi hálózat, gráf

WEBOLDALAK

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

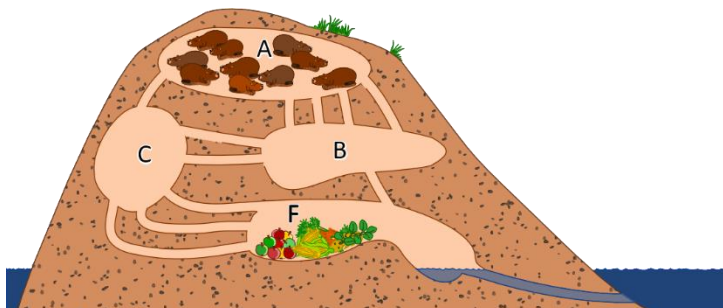


HÓDVÁR JÁRATAI (2017-CH-07B)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

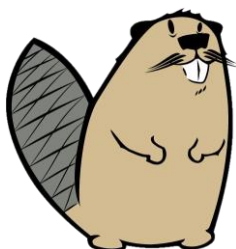
Az A jelzésű barlangban 10 hód található, akik a lehető leggyorsabban szeretnék eljutni az F barlangba az ennivalóhoz. Egy hódnak egy percbe telik, hogy egy járaton átérjen a szomszédos barlangba, azonban egy járatot egyszerre csak egy hód használhat. (Közvetlenül egymás mögött nem osonhat több hód egyszerre egy járatban.)



Az A, B, C, F barlangokban minden hód számára elegendő hely van, és a barlangon való áthaladás nem igényel plusz időt.

Legkevesebb hány perc szükséges ahhoz, hogy mind a 10 hód az F barlangba érjen?

- A) 2 perc
- B) 3 perc
- C) 4 perc
- D) 5 perc



„C” válasz a helyes:

Mind a 10 hód 4 perc alatt ér az F barlangba. A hódvárban 2 legrövidebb út vezet az F barlangba. Mindkét járat 2 perc alatt mindig egy-egy hódot juttat a célhoz, a 3. percben ez mindenkor két hód.

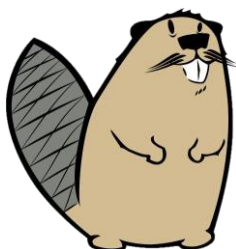
$$A \rightarrow B \rightarrow F$$

$$A \rightarrow C \rightarrow F$$

Az $A \rightarrow B \rightarrow C \rightarrow F$ útvonal 2 hódnak biztosít helyet, de az útvonal 3 perc alatt járható be. Tehát 3 perc után összesen 6 hód jutott az F barlangba. Kell egy 4. perc, hogy mindegyikük eljusson az ételhez. Az alábbi táblázat azt tartalmazza, miként juthat el a 10 hód az A barlangból az F-be.

Leírás / változás	A barlangokban lévő hódok száma (lépés/változás után)			
	A	B	C	F
Alaphelyzet	10	0	0	0
3 hód A-ból B-be megy (kevesebb, mint amennyi lehetséges lenne)				
1 hód A-ból C-be megy				
Eloszlás 1 perc után	6	3	1	0
3 hód A-ból B-be megy (kevesebb, mint amennyi lehetséges lenne)				
1 hód B-ből F-be				
2 hód B-ből C-be				
1 hód C-ből F-be				
1 hód A-ból C-be				
Eloszlás 2 perc után	2	3	3	2
1 hód A-ból B-be (legrovidebb úton)				
1 hód B-ből F-be				
2 hód B-ből C-be				
1 hód A-ból C-be (legrovidebb úton)				
3 hód C-ből F-be				
Eloszlás 3 perc után	0	1	3	6
1 hód B-ből F-be				
3 hód C-ből F-be				
Eloszlás 4 perc után	0	0	0	10

Különböző megoldási lehetőségek léteznek arra, miként juthatnak el a hódok 4 perc alatt az F barlangba. Az általunk felvázolt megoldásban egy teremben sem kell várakozniuk a hódoknak, egyből továbbmehetnek.



MIÉRT INFORMATIKA?

A járatok hálózata úgynevezett „áramként” (flow-network) fogható fel. A két barlang közötti járatok száma meghatározza, hány hód juthat át egy perc alatt egy másik barlangba. A két barlang közt az úgynevezett kapacitás korlátozza az érintett barlangok közti folyamatot.

Az „áram” a gráfelméletben egy olyan irányított gráf, amelynek minden útja rendelkezik egy áteresztő képességgel (jelen esetben ez a barlangok száma). Az áramlást, amely a gráf útjain át zajlik le, az utak ezen képessége korlátozza. Egy hálózatban lezajló folyamatokkal szimulálhatunk például egy számítógépes vagy egy közlekedési hálózatot, továbbá a közlekedés áramlásában fellépő szűkületek is felfedezhetőek.

A folyamatgráfokban különösen az érdekes, hogy legfeljebb mekkora áramlás lehetséges két csúcs között. Feladatunkban ez nem más, mint a 4 hód, amelyek A-ból az F barlangba szeretnének eljutni úgy, hogy a továbbhaladásra egy barlangban se kelljen várakozniuk és percenként egy barlang távolságot tehetnek meg. A maximális áram nagysága az úgynevezett Ford-Fulkerson algoritmussal számolható ki.

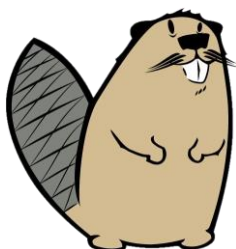
KULCSSZAVAK

Gráf, folyam

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Áram_\(gráfelmélet\)](https://hu.wikipedia.org/wiki/Áram_(gráfelmélet))

https://hu.wikipedia.org/wiki/Maximális_folyam-minimális_vágás



BRINGÁRA FEL! (2017-DE-09)

JUNIOR – NEHÉZ

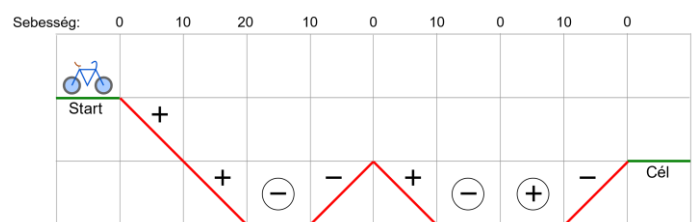
SENIOR - KÖZEPES

A bringázásnak vannak különlegesebb módjai. Egy ilyen, amikor speciális kerékpárokkal különböző pályákon kell végigmenni. Mindegyik pálya szakaszok sorozatából áll. Egy szakasz iránya „hegyre fel”, „hegyről le” vagy pedig „vízszintesen” lehet.

Egy pálya teljesítése az alábbi szabályok szerint történik:

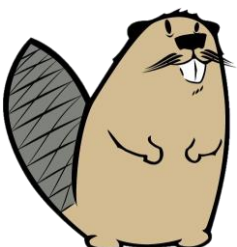
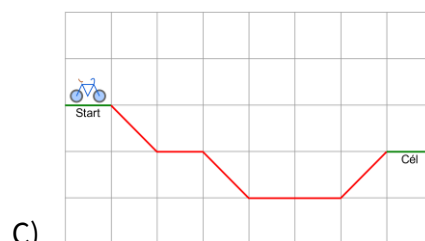
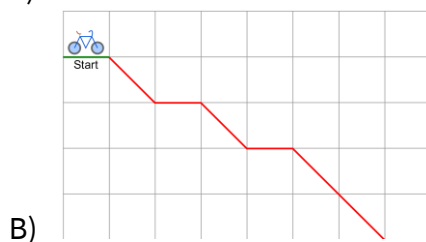
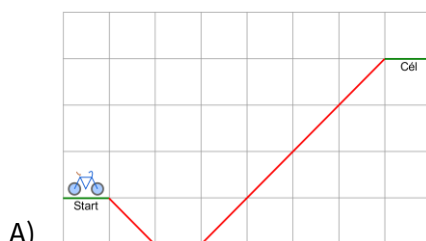
- A rajtnál 0 km/h a kerékpáros sebessége.
- Lefelé menetben 10 km/h-val nő a kerékpár sebessége.
- Hegyre felfelé tartó szakaszon a kerékpár sebessége 10 km/h-val csökken.
- Vízszintes szakaszon Te magad döntheted el, hogy a sebesség nőjön vagy csökkenjen 10 km/h-val.
- A célban újra 0 km/h kell legyen a kerékpáros sebessége.

Menet közben lehet 0 km/h a sebesség, azonban azonnal növelni kell, különben a kerékpáros elesik és nem tudja befejezni a pályát. Itt látható egy pálya vázlata, mely ezek alapján a szabályok alapján megtehető.



A célbaéréshez a vízszintes szakaszokon úgy kell változtatni a sebességet, ahogy a körökben meg van adva.

Az alábbiakban további pályák vázlatát láthatod. Csak egy fejezhető be a megadott szabályok szerint. Melyik az?



„C” válasz a helyes:

Egyedül csak a C pályán lehet végigmenni.

A „C” pálya teljesítéséhez a vízszintes szakaszokon az alábbi sebességváltoztatásokat kell végrehajtani: + – –, azaz gyorsítás, lassítás majd újra lassítás. Célbaéréskor így lesz 0 km/h a kerékpár sebessége. Akkor is 0 km/h, ha az alábbi változások történnek: – + – vagy – – +.

A többi pálya nem teljesíthető:

- „A” pálya: Ha minden lehetséges helyen nő a kerékpár sebessége, akkor is csak az emelkedő feléig jut a kerékpáros.
- „B” pálya: Célbaéréskor a kerékpár sebessége nagyobb, mint 10 km/h, akkor is, ha a vízszintes szakaszokon lassít a kerékpáros.
- „D” pálya: Ahhoz, hogy a célban 0 km/h legyen a sebessége, a kerékpárosnak az utolsó szakaszon lassítania kellene, de hegyről lefelé a sebesség nő.

MIÉRT INFORMATIKA?

A zárójelek fontos szerepet játszanak a műveletek világában. A matematikai kifejezésekben, mint például $[n(n-1)]/2$ vagy $(a+b)(a-b)$ meghatározzák a zárójelek, hogy milyen sorrendben kell kiértékelni az egyes részkifejezéseket. A zárójelek mindig párosával fordulnak elő: egy nyitó és egy záró zárójel. Egy zárójeles kifejezés akkor helyes, ha minden nyitó zárójelet pontosan egy csukó zárójelhez, minden csukót pedig egy nyitó zárójelhez tudunk rendelni.

Nem csak a matematikában, hanem az informatikában is vannak zárójeles műveletek, melyeket a számítógépeknek fel kell dolgozniuk. Jó példa erre a HTML nyelv, amely a weboldalak leíró nyelve. HTML kódban egy bekezdést `<p>` és `</p>` tag-ek határolnak. Minden HTML tag-ben van egy `<` és `>`. A zárójelpárok nagyon népszerűek az informatikában, mert a számítógép könnyen fel tudja dolgozni őket a verem (egyszerű és hatékony adatszerkezet) segítségével.

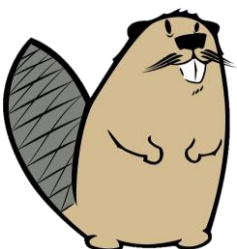
Az egyes pályaszakaszok olyanok, mint a zárójelek. Egy „hegyről le” szakasz megfeleltethető a nyitó, a „hegyre fel” pedig a csukó zárójelnek. A vízszintes szakasz a joker terület, ami lehet nyitó vagy csukó zárójel. Egy pálya akkor teljesíthető, ha a pályaszakaszok zárójelre cserélése után helyes zárójeles kifejezés jön létre. A „C” pálya az alábbi módon szemléltethető: `(?())`. Ez a pálya megtehető, mert a `?` helyére tudunk olyan zárójelet írni, hogy helyes zárójeles kifejezés szülessen. Ezek a következők lehetnek: `((()))` vagy `()()` vagy `()()`. Az elméleti informatikában azokat a nyelveket, amelyek csak helyes zárójeles kifejezésekből állnak, „Dyck-nyelvnek” nevezzük.

KULCSSZAVAK

Formális nyelvek, környezetfüggetlen nyelvek, Dyck nyelv

WEBOLDALAK

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



FOGPISZKÁLÓS JÁTÉK (2017-HU-06)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

Helga és Béla egy társasjátékot játszanak: A játékban két halom fogpiszkáló van az asztalon és a játékosok felváltva kerülnek sorra.

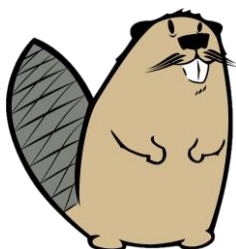
A soron következő játékosnak a következő lépéseket kell megtennie:

- Teljesen elveszi az egyik kupacot az asztalról.
- Szétosztja az asztalon maradt fogpiszkálókat két kisebb kupacra.

Az a játékos nyer, aki két, pontosan egy-egy fogpiszkálóból álló kupacot hagy az asztalon. Helga kezdi a játékot.

A játék 24 fogpiszkálóval indul két kupacba osztva. Melyik kezdő elosztásban tud nyerni Helga?

- A) 11 és 13
- B) 8 és 16
- C) 7 és 17
- D) 5 és 19



„B” válasz a helyes:

A győztes stratégia az, ha két páratlan számú fogpiszkálóból álló kupacot készítünk. Ha a kezdőkupacok páros számú fogpiszkálót tartalmaznak, a második játékos fog nyerni (feltéve, hogy mindketten hiba nélkül játszanak).

(Nem tudunk 24 fogpiszkálót egy páros és egy páratlan számú kupacra osztani.)

Miért?

Ha a játékosnak két páratlan számú fogpiszkálóból álló kupaca van, csak egy páros és egy páratlan számú kupacra oszthatja bármelyiket. A következő játékos elveszi a páratlan számú kupacot és a másikat két páratlan kupacra osztja szét.

A játék akkor ér véget, amikor két kupacban 1-1 fogpiszkáló van. Ha páratlan kupacok vannak, amikor valaki köre jön, az biztosan tud nyerni a fenti stratégiával.

Ebben a játékban nem elég egy jó nyerő stratégia, hanem egy jó kezdő pozícióra is szükségünk van.

MIÉRT INFORMATIKA?

Ahhoz, hogy jó stratégiát alakítsunk ki egy szituációban, egy állandó célt keresünk: teljesüljön a körünk után, de a másik játékos ne érhesse el. Alkalmazd ezt és a játék végén is igaz lehet.

Van néhány olyan játék, ahol a kezdő pozíció mindig ugyanaz és készíthetünk egy nyerő stratégiát, de általában nagyon bonyolult kiszámolni minden lehetséges lépést.

Ebben a feladatban (és a legtöbb játékban is) csak egy “rossz” lépés vagy helyzet után lehet jó stratégiánk.

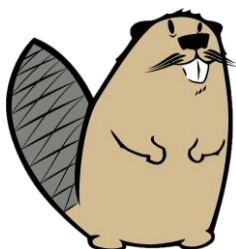
A programozók (informatikusok) szeretnek stratégiákat kidolgozni és mesterséges intelligenciával (MI) kombinálni, majd ezek alapján számítógépes játékokat fejleszteni. A lehetséges lépéseket általában fákkal ábrázolják, és a „számítógép” ebben a fában keres számára megfelelő lépéseket.

KULCSSZAVAK

Játékelmélet, logika

WEBOLDALAK

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



SZÓTÁVOLSÁG (2017-PL-02)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

Két szó közötti távolság a legkevesebb olyan lépések száma, amelyeken keresztül az egyik szótól eljuthatunk a másikig.

Ahhoz, hogy két szó közötti távolságot meghatározzuk, a következő lépéseket kell megtennünk:

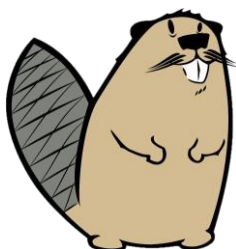
- Egy tetszőleges helyre beszúrunk egy betűt.
- Egy tetszőleges helyről elveszünk egy betűt.
- Egy tetszőleges helyen álló betűt kicserélünk egy másikra.

Például a „bennem” és a „teremt” közötti távolság: 4, mert

1. bennem → tennem („b”-t „t”-re cseréltük)
2. tennem → ternem („n”-t „r”-re cseréltük)
3. ternem → terem („n”-t kivettük)
4. terem → teremt („t”-t beszúrtuk)

Mi a távolság az „Emil” és az „Erich” szavak között?

- A) 3
- B) 4
- C) 5
- D) 6



„A” válasz a helyes:

„Emil” és „Erich” közötti távolság 3, mely például állhat az alábbi lépésekből:

Emil → *Eril* → *Eric* → *Erich*

Kevesebb lépésből ez nem megoldható, mivel az „Erich” eggyel több betűből áll, így mindenképpen szükséges egy beszúrás (egy lépés) és az „Erich”-ben nem szerepel „m” és „l”, így azokat minimum ki kell cserélni (2 lépés).

MIÉRT INFORMATIKA?

A szavak közötti ilyen különbséget Levensthein-távolságnak nevezik. Vladimir Levensthein, orosz matematikus vezette be ezt a fogalmat 1965-ben. Például arra használják, hogy egy helyesírás-ellenőrző a rosszul leírt szóra helyesen leírtat tudjon javasolni: ha a rosszul leírt szó és a javaslat szerint jó szó között kicsi a távolság, akkor valószínűleg elgépelés történt.

Hasonló távolságvizsgálatokat használnak gyakran pl. a DNS-vizsgálatoknál, képek vagy az automatizált szövegfordítások esetében, plágiumkeresésnél is.

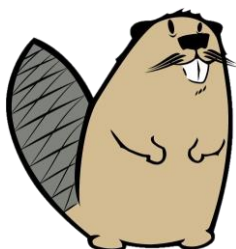
A számítógéppel kiszámolhatjuk a Levenshtein-távolságot úgy, hogy minden lehetséges helyettesítést kipróbálunk. Mivel a számítógép ehhez nem alkalmaz feleslegesen hosszú szavakat, határokat állíthatunk fel a vizsgálatban.

KULCSSZAVAK

Levenshtein távolság, minta, algoritmus

WEBOLDALAK

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

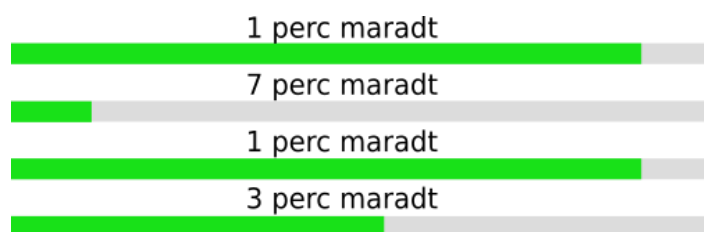


LETÖLTÉSI LISTA (2017-RU-01)

JUNIOR – NEHÉZ

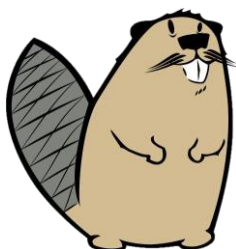
Több internetes állomány egyidejű letöltése esetén a hálózat kapacitása megoszlik. 10 állomány egyidejű letöltésekor mindegyik állomány a hálózati kapacitás egytized részét használhatja.

Ebben a feladatban tekintsünk el a kiszolgáló sebességétől, mert ezek leggyakrabban nagyobbak, mint a fogadó gépe. Egy felhasználó éppen 4 állományt tölt le egyidejűleg. A hátralévő időt az aktuális letöltési sebességből számolhatjuk ki:



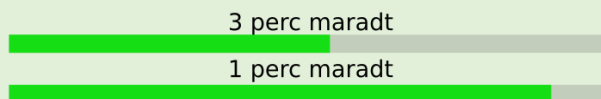
Mennyi időbe telik, míg mindegyik állomány letöltődik?

- A) 3 perc
- B) 4 perc
- C) 5 perc
- D) 6 perc

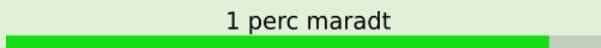


„A” válasz a helyes:

Egy perc múlva két fájl töltődik le. A másik kettőnél a hátralévő idő 6 ill. 2 percre csökken. A letöltési sebesség megkétszereződik, köszönhetően a két befejezett letöltésnek, azaz 3 és 1 perc marad hátra:



A következő perc elteltével a harmadik fájl is letöltődik. Az utolsó fájlnek még 2 perc kellene, de a letöltési sebesség újra megduplázódik, így csak 1 percet kell még várni:



Összesen tehát 3 perc alatt töltődik le a négy állomány.

MIÉRT INFORMATIKA?

A *folyamatjelző sáv* (angolul: progress bar) egy vezérlőelem, ami folyamatosan mutatja egy feladat (pl. telepítés, letöltés) feldolgozottsági szintjét. Közben meg is becsüli, hogy mennyi ideig tart még a folyamat. Kivételt képeznek azok a folyamatjelzők, amelyeknél a hátralévő idő nem ismert. Ebben az esetben feltétlenül jelezni kell, hogy a számítógép továbbra is dolgozik, nem fagyott le. Ilyenkor a „hátralévő idő: ismeretlen” jelzés, egy homokóra vagy egy forgó szimbólum jelenik meg.

Az *adatátviteli ráta*, amelyet más néven átviteli sebességnek, vagy nem túl találóan sávszélességnek is hívunk, az a digitális adatmennyiség, amely egy időegység alatt az átviteli csatornán átvitt adatmennyiséget jelenti. Egy csatorna lehető legmagasabb adatátviteli rátáját csatornapacitásnak nevezzük. Egy csatorna fontos jellemzője továbbá az úgynevezett válaszkésleltetés vagy a látensidő, ezzel lehet megadni, hogy az adatoknak mennyi időre van szükségük a feladótól a vevőig. (időnként a feladóhoz vivő visszautat is beleszámolják)

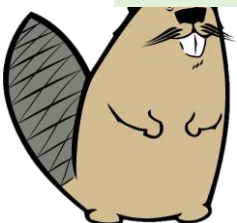
Az interneten számos tényező befolyásolja két számítógép közötti maximális adatátviteli rátát: először is egy szerver csak egy maximális sebességgel képes adatot szállítani, ami a hardver- és szoftverállománytól függ. Ezenkívül ott vannak azok a szerverek, amelyek korlátozott adatátviteli rátával (manapság gyakran csak 1 GBit/s-mal) rendelkező sávval kapcsolódnak az internethez. Az összes, egyidejűleg lezajló adatátvitelnek ezen a sávon kell osztoznia, majd az adatoknak az interneten különböző sávokon kell eljutniuk a vevőhöz, aki néha a világ másik felén van. Habár ezek a sávok nagy teljesítményre képesek, mégis az összes adat ezeken a sávokon közlekedik. További lassító tényező az internetszolgáltató és az ügyfél közötti kapcsolat. Az adatátviteli ráta gyakran itt a legkorlátozottabb, manapság gyakran 100 MBit/s vagy még lassabb. Végezetül pedig az adatnak el kell jutnia a router-től magáig a számítógépig. Kábeles összeköttetés rendszerint gyorsabb, mint az internetre kapcsolódás sebessége. Amennyiben WLAN hálózatról van szó, további korlátozás lehet, pl. 27 MBit/s.

KULCSSZAVAK

Letöltés, folyamatjelző sáv

WEBOLDALAK

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

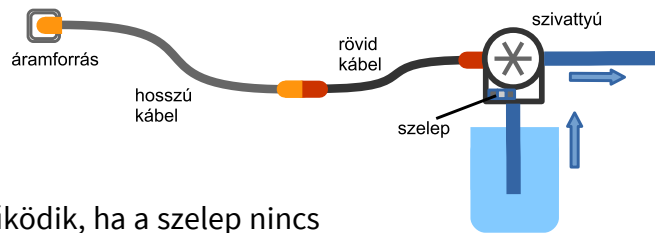


ÖNTÖZŐ RENDSZER (2017-AT-05)

SENIOR – KÖNNYŰ

Egy hódnak van egy virágágyása és egy zöldségeskertje, melyekhez két ugyanolyan öntözőberendezést telepített. Az ábrán ezek közül az egyik látható. Az öntözőrendszer hálózati áramforrásról működik és az alábbi alkatrészekből áll:

- egy hosszú kábel
- egy rövid kábel
- egy szivattyú
- és a szivattyúszelep. (A szivattyú nem működik, ha a szelep nincs a helyén.)

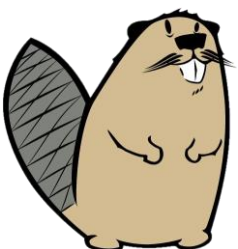


Egy napon a virágágyás öntözőrendszere leállt. A hód megállapította, hogy a leállás oka nem a víztartály vagy a vízvezeték hibájából fakad. A zöldségek öntözése ellenben tökéletesen működik, így ez a készülék felhasználható a hiba megtalálására.

A rendszerben egyetlen egy elem ment tönkre, melyet a hód meg szeretne találni. Ehhez a zöldségeskertben található locsolóberendezést alkatrészeire tudja bontani. Úgy szeretné megtalálni a hibát, hogy a két locsolóberendezés elemeit cserélgeti.

Az alábbi javaslatok közül melyiket fogadja meg, hogy minél biztosabban és gyorsabban megtalálja a hibás alkatrészt?

- A) A tesztelést a szivattyúval kell kezdenie, mert ez a berendezés legfontosabb része.
- B) Minden lépésben csak egy alkatrészt cserélhet ki. Ha a rendszer újra működik, akkor az utoljára kicserélt elem volt a hibás.
- C) Inkább új alkotórészekről kell gondoskodnia, mert a zöldségeskert locsolója használt.
- D) Minden lépésben legalább két alkatrészt kell kicserélnie, így hamarabb eléri a célját.



„B” válasz a helyes:

- A) Egy alkatrész fontosságából nem tudunk következtetni annak hibavalószínűségére.
- C) A vadonatúj alkatrészek is hibásak néha. A zöldségeskert locsolórendszere azonban mindenképp ép, ugyanis a másik öntözőrendszer jelen esetben működik.
- D) Valóban gyorsabb lenne a folyamat, ha arról lenne szó, hogy minél hamarabb javítsa ki. Sőt, a hód egyszerűen kicserélhetné a zöldségeskert locsolójával, azonban a hód a hibás alkatrészt szeretné megtalálni. Ha egyszerre több alkatrészt cserél ki, akkor további teszteket kell még elvégeznie, hogy a hibásat megtalálja.

MIÉRT INFORMATIKA?

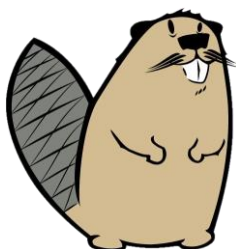
A hibakeresés - más néven debuggolás – egy fontos feladat, mert az összetett szerkezeteknél előbb vagy utóbb hibák léphetnek fel. A hiba megtalálása igazi kihívás, különösen akkor, ha egyszerre kettő vagy több is fennáll. Ennek elkerülése érdekében a fejlesztők lehetőleg minden egyes változtatás után tesztelik a gépet. Ezzel növelhetjük annak esélyét, hogy csak egyetlen egy hibát kelljen keresnünk.

KULCSSZAVAK

Hibakeresés, debuggolás, keresés

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Hibajavítás>
[https://hu.wikipedia.org/wiki/Bug_\(informatika\)](https://hu.wikipedia.org/wiki/Bug_(informatika))
<https://hu.wikipedia.org/wiki/Szoftvertesztelés>



PONTGYŰJTÉS (2017-CA-12)

SENIOR – NEHÉZ

Egy népszerű gondolkodós játék leírása következik.

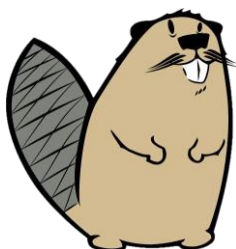
Adott egy táblázat (lásd lentebb), amelyben az **S** (start) mezőről kell eljutni a **C** (cél) mezőre úgy, hogy minden lépésben csak jobbra vagy felfele léphetsz, ahogy a nyilak is mutatják.

Minden cellában van egy szám, ami az adott mezőn megszerezhető pontszámot jelzi. Válaszd úgy ki a célba vezető utat, hogy a lehető legtöbb pontot gyűjtsd össze!

	2	0	1	1	C
	1	2	0	2	3
	2	2	0	2	1
	3	1	0	2	0
↑	S	0	1	3	0
					→

Maximálisan hány pont érhető el?

- A) 10
- B) 12
- C) 14
- D) 16



„C” válasz a helyes:

Legfeljebb 14 pont gyűjthető össze. A következő táblázat az adott mezőig maximálisan megszerezhető pontszámot tartalmazza, a jobb felső sarokban pedig a végső, helyes megoldás látható. Ezt a táblázatot akkor kapod meg, ha minden egyes mezőnél megvizsgálod a bal és alsó szomszéd maximumát, melyhez hozzáadod az eredeti táblázat mezőértékét. A helyes megoldást szürkével jelöltük:

	8	9	10	12	14
	6	9	9	11	14
	5	7	7	9	10
	3	4	4	6	6
	0	0	1	4	4

MIÉRT INFORMATIKA?

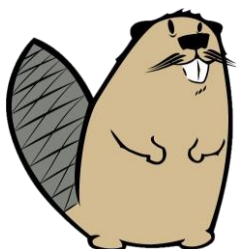
A leírt módszer nem feltétlen a leggyorsabb megoldási mód. Az agyunk nagyon hatékony a megoldást már-már magától adó feladatok megoldásában, ugyanis az eredmény néha azonnal látható. Ez a módszer azonban nem hibamentes, és csak az átlátható méretű táblázatok esetén működik. Ezzel szemben a fent leírt folyamat mindig helyesen működik, amennyiben az egyes lépéseket helyesen írták le. A program ezeket a lépéseket tetszőleges méretű táblázaton (informatikában mátrixon) végre tudja hajtani. Ez egy jó program ismerve.

KULCSSZAVAK

Algoritmikus gondolkodás

WEBOLDALAK

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



KÉPTÖMÖRÍTÉS (2017-KR-07)

SENIOR – NEHÉZ

A csak fekete és fehér képkockákból álló képek 0 és 1 számjegyekkel leírhatóak: 0 jelenti a fehér, 1 pedig a fekete képkockát. Egy négyszer négy képkockából álló képet így 16 számjeggyel (0 vagy 1) le tudunk írni.

Sok képet kevesebb jellel is ábrázolhatunk, ha a „negyed” eljárást alkalmazzuk.

Ehhez a rajzhoz egy négyzetrácsot rendelünk. A „negyed” eljárás így alkalmazható egy ilyen rácsra:

1. Ha a négyzetrács minden számjegye 0, akkor az eredmény „0” (lásd a bal képet). Ha a négyzetrács minden számjegye 1, akkor az eredmény „1”.
2. Minden más esetben a négyzetrácsot négy azonos nagyságú részre osztjuk. A „negyed” eljárást ezután az egyes negyedekre alkalmazzuk a bal felső negyedről kezdve, az óramutató járásával megegyező irányba haladva. A négy (rész)eredményt egymás után írjuk le zárójelek közé téve. Két különböző példát láthatsz a középső és jobb oldali képen.

0 0 0 0	1 1 0 0	1 1 0 0
0 0 0 0	1 1 0 0	1 1 0 0
0 0 0 0	1 1 1 1	1 1 0 1
0 0 0 0	1 1 1 1	1 1 0 1
0	(1011)	(10(0110)1)

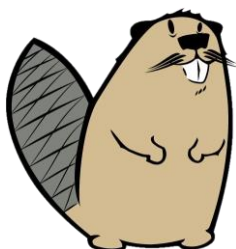
Figyelem: Ha egy rács már csak egyetlen számból áll, az eredmény pontosan ez az egy szám.

Itt egy 8x8-as kép leírása látható.

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

A „negyed” eljárást alkalmazva, milyen eredményt fogunk kapni?

- A) (1 1 1 0)
- B) (1 1 (1 0 1 1) 1)
- C) (1 1 1 (1 1 0 1) 1 1))
- D) (1 1 1 (1 (1 0 1 1) 1 1))



„D” válasz a helyes:

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

(111(1(1011)11))

MIÉRT INFORMATIKA?

A körülöttünk levő adatok rendszerint analóg jelekből állnak, vagyis folytonos értékeket vesznek fel. Gondolhatunk olyan tulajdonságra, mint az emberek magassága, vagy a képek színárnyalatai. Az adatok számítógépes tárolásakor azonban ezeket a folyamatos értékeket csak digitális jelként, vagyis megadott értékek halmazaként rögzíthetjük.

A számítógépek működéséből eredően a tárolás kettes számrendszerben történik, vagyis minden értékhez hozzárendelünk egy kettes számrendszerbeli számot. A fenti képtömörítési eljárásban a kép kockáihoz (pixel) először a színüknek megfelelő értékeket rendelünk hozzá (több szín esetében nem csak 0 és 1, hanem 10, 11, 100, 101, 110, stb. számokat). Ezzel létrehoztuk a kép digitális jelét, azaz digitalizáltuk.

Az adatok részletességüktől függően nagyon sok helyet elfoglalhatnak a tárolóegységeken, ezért sokan foglalkoznak azzal, hogy minél jobban tömörítsék őket. A példában szereplő 64 pixelből álló képet a megadott eljárással a zárójeleket leszámítva sikerült 10 számjegyű kettes számrendszerbeli számmal eltárolnunk.

A tömörítési eljárások rendkívül sokszínűek, de a legtöbbnek ugyanaz a célja: minél kisebb helyen tárolhassuk az adatokat és ha szükséges, vissza tudjuk alakítani a kezdeti állapotba.

Csak bizonyos típusú képekre alkalmazható.

KULCSSZAVAK

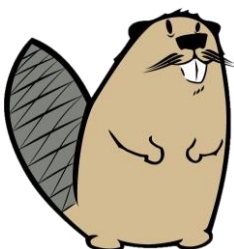
Képtömörítés, négyágú fa

WEBOLDALAK

http://www.agt.bme.hu/tutor_h/terinfor/t23.htm

<http://tudasbazis.sulinet.hu/hu/informatika/informatika/informatika-9-12-evfolyam/keptomorites-veszteseges-es-nem-veszteseges-tomorites-fourier-transzformacio/grafikus-kepek-vesztesegmentes-kodolasa-2>

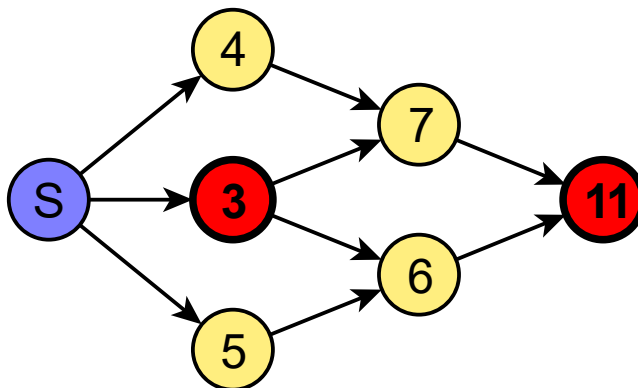
<http://www.karinthy.hu/home/informatika/jel/index.html>



RÖVIDÍTÉS VAGY KITÉRŐ? (2017-NL-05)

SENIOR – NEHÉZ

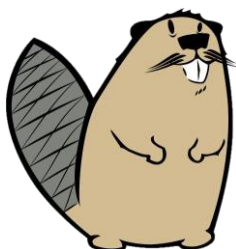
Az alábbi ábra egyirányú utak kapcsolatát ábrázolja.



A kereszteződésekben található szám azt jelzi, hogy az S pontból hány egység hosszú a legrövidebb út az adott kereszteződésig.

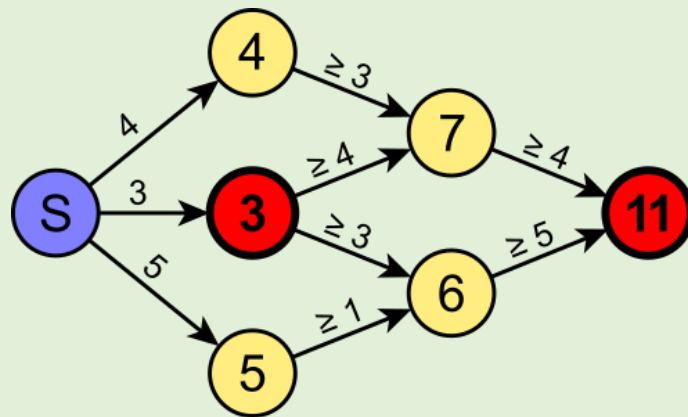
Az alábbi állítások közül melyik igaz mindig a két vastagon kiemelt (piros) kereszteződésről?

- A) A két kereszteződés közötti legrövidebb út hossza pontosan 8 egység.
- B) A két kereszteződés közötti legrövidebb út hossza 8 vagy 8-nál kevesebb egység.
- C) A két kereszteződés közötti legrövidebb út hossza 8 vagy 8-nál több egység.
- D) Semmi sem állapítható meg az adott kereszteződések közötti legrövidebb út hosszáról.



„C” válasz a helyes:

Ha a legrövidebb út 8 egységnél kevesebb lenne, akkor az S és a „11”-es kereszteződés közötti legrövidebb út $3+8=11$ -nél kevesebb lenne. A legrövidebb út azonban 8 egységnél nagyobb lehetne, mivel az S és a „11”-es kereszteződés közötti út a „4” és „7” vagy az „5” és „6” egységnyi hosszú utakon vezetne. Ezért hamis az összes többi állítás.



A fenti ábra azt mutatja, hogy legalább milyen hosszúnak kell lenniük az egyes szakaszoknak ahhoz, hogy a megadott számok helyesek legyenek.

MIÉRT INFORMATIKA?

Első ránézésre úgy tűnhet, mintha ebben a feladatban csupán két pont közötti legrövidebb út kiszámításáról lenne szó. Ehhez azonban ismerni kellene két kereszteződés közötti tényleges távolságot, amelyet rendszerint a szakaszok fölé mint „szakaszhossz” feljegyezhetnénk.

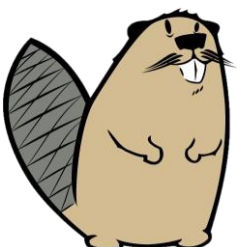
Ebben a feladatban azonban csak az S pont és az adott kereszteződés közti legrövidebb út van feljegyezve, így lehetséges lenne, hogy a „4”-től „7”-ig tartó szakasz 10 egység hosszú, ha „3”-tól „7”-ig tartó szakasz 4 egységnyi. Ebből következik, hogy további információk hiányában nem állapítható meg, hogy melyik a lehetséges utak közül a legrövidebb.

KULCSSZAVAK

Legrövidebb út

WEBOLDALAK

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



BONTSD RÉSZEKRE A SZÁMSORT (2017-RU-05)

SENIOR – NEHÉZ

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

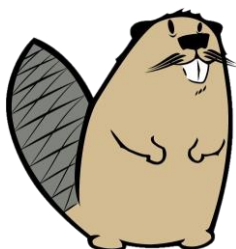
Egy példa: Az X betűt 12-ként kódolja. Ekkor az Y-t kódolhatja 2-ként. Így sem a 12 nem kezdődik 2-vel, sem a 2 12-vel. Most a Z-t 11-ként kódolhatja, mivel sem 12 sem 2 nem kezdődik 11-gyel és a 11 sem kezdődik 2-vel vagy 12-vel. A 21 már nem engedélyezett a Z kódolására, hiszen 2-vel kezdődne, ami viszont már az Y-hoz tartozó kód.

A BEBRAS szót a rendszer így kódolta:

12112233321

Melyik számsor jelölheti az A betűt?

- A) 22
- B) 23
- C) 33
- D) 233



„C” válasz a helyes:

A teljes **BEBRAS** szó kódfeosztása csak a következő lehet:

1 – 21 – 1 – 22 – 33 – 321

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

Nem lehet, hogy a **B** betűt több, mint **2** számként kódoljuk (**121**, **1211**, **12112**, stb.), mert még egyszer meg kellene ismétlődnie. Ezáltal a **B** betűt kizárólag az **1**-es kóddal láthatjuk el.

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

Tekintsük a szó végén álló **S** betűt. Nem lehet a kódja **1**, sem **21**, mivel ezek a számsorok már foglaltak a **B** és **E** betűk kódolásánál. Következésképp ezek lehetnek a lehetséges kódok az **S**-hez: **321**, **3321**, **33321**, **233321** és **2233321**. **S**-t nem kódolhatjuk **2233321**-ként, hiszen akkor a szó mindössze **BEBS** lenne. Ugyanilyen okból nem lehet **233321** sem, mert akkor is hiányozna még az **R** vagy az **A** betű (ha csak egy-egy számjeggyel kódolnánk is). Ha **S**-t **3321**-ként kódoljuk, az **RA** párost **223**-ként kell. Azonban az **R** nem lehet **2**, és az **A** sem lehet csak **3**, hiszen más kódszavak már kezdődnek ezekkel a számokkal. Ezek alapján az **S** csakis **321**-ként kódolható. Az **RA** párosra marad a középső számsor: **2233**. A már ismert okok miatt **R**-t **22**-ként, **A**-t **33**-ként kódoljuk.

MIÉRT INFORMATIKA?

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

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

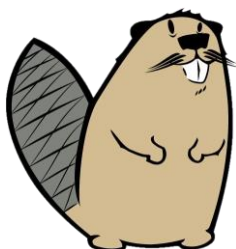
A Huffman-kódolás egy módszer hatékony előtagkódolás keresésére. Széles körben elterjedt és olyan adatformátumok használják, mint pl. a JPEG és az MP3.

KULCSSZAVAK

Kódszöveg, prefix kód, előtag

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Huffman-kódolás>



PAPÍRHAJTOGATÁS (2012-DE-06)

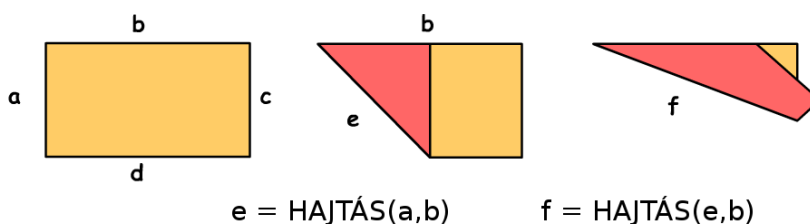
SENIOR – NEHÉZ

A hódok kifejlesztették a papírhajtogatás nyelvét. A nyelv leírása az, ahogyan egy papírral egyenes éleket hajtunk.

Ebben a nyelvben az utasításokat HAJTÁS-nak hívják.

$z = \text{HAJTÁS}(x, y)$ azt jelenti: hajts egy darab papírt úgy, hogy az x éle pontosan az y élére kerüljön. Így egy új él áll elő, melyet z -nek nevezünk.

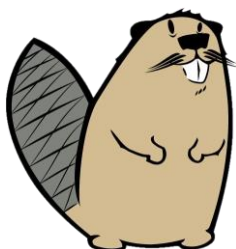
Például két utasítással egymásután: $e = \text{HAJTÁS}(a, b)$, $f = \text{HAJTÁS}(e, b)$



Vegyél egy négyszögletes papírt, melynek a b, d élei kétszer olyan hosszúak, mint az a, c élei. A papírt a hajtogatás alatt az asztalon kell hagynod (elfordítás nélkül). Hajtsd végre a következő utasításokat egymás után:

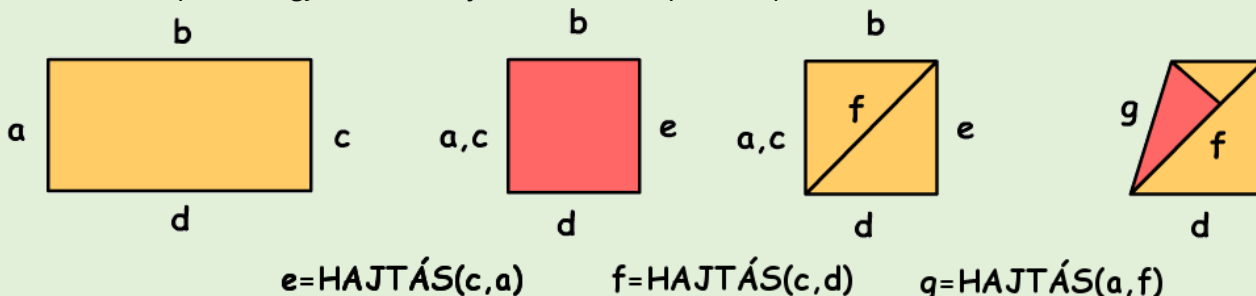
$$e = \text{HAJTÁS}(c, a) \quad f = \text{HAJTÁS}(c, d) \quad g = \text{HAJTÁS}(a, f)$$

Hogy néz ki a papír a hajtogatások után?



„A” válasz a helyes:

A következő képek elmagyarázzák a hajtás-sorozatot lépésről lépésre:



MIÉRT INFORMATIKA?

A feladatban szereplő *hajtás*(a, b) utasítás bemeneti adatai két oldal betűjele, mely oldalakat aztán egymásba kívánjuk hajtani. A papírlap hajtogatási állapotát eközben a számítógépnek követnie kell, vagyis az új oldalon kívül az adott esetben az a oldal változását is fel kell jegyeznie.

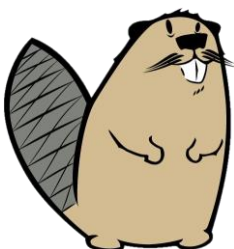
A matematikai függvény fogalomtól ez abban tér el, hogy matematikailag a bemeneti adatok mellett rendszerint a kimeneti adat is egy-egy érték vagy számhalmaz. Az informatikában gyakori, hogy egy objektum tulajdonságait változtatjuk egy függvénnyel. Most a papírlapunk az objektum és a *hajtás* függvényt meghívva, a bemeneti adatoknak megfelelő módosításokat elvégezve, a hajtás utáni állapotát adjuk tovább. Ezt a függvény visszatérési értékének szokás nevezni.

Amennyiben egy programozási függvény nem tér vissza értékkel (csak módosításokat végez), akkor függvény helyett eljárásról beszélünk.

KULCSSZAVAK

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Függvény_\(programozás\)](https://hu.wikipedia.org/wiki/Függvény_(programozás))



TÁMOGATÓINK, KÖSZÖNETNYÍLVÁNÍTÁS

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

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

