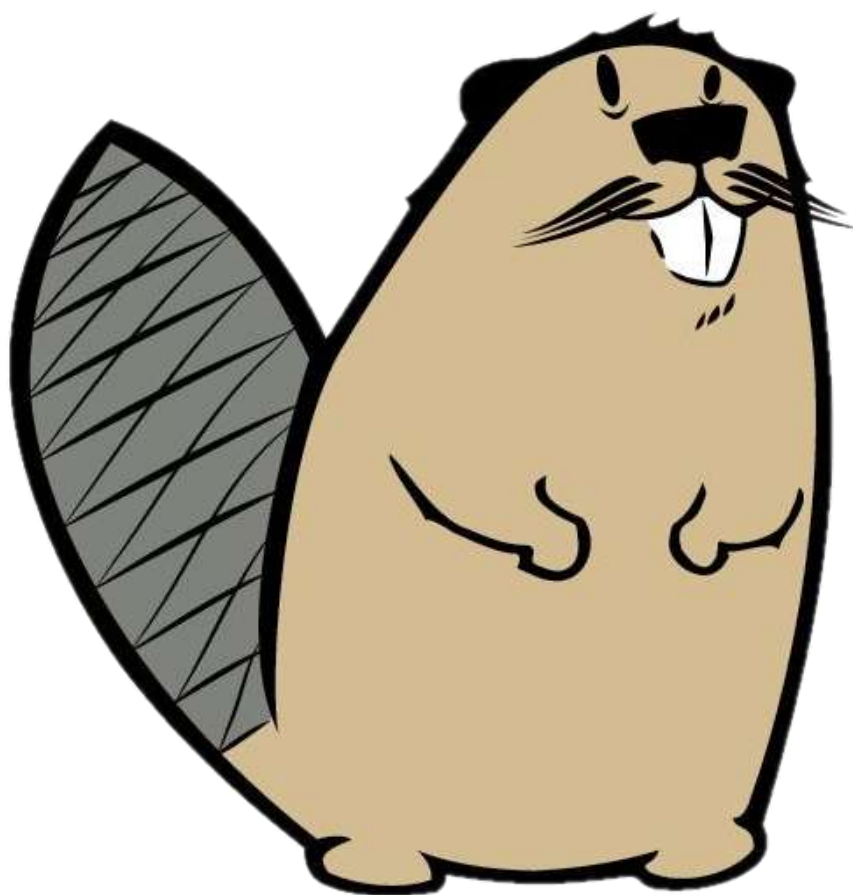


HÓDÍTSD MEG A BITEKET!

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



2019

MI IS AZ E-HÓD?

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

A nemzetközi Bebras, melyhez 2017-ben már közel 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 megválaszolásukhoz. A feladatok érdekes problémákat mutatnak be. Nem tesztek, inkább szórakoztató gondolkodtató feladványok.

Magyarországon 2019-ban kilencedik 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 2019-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, 18 feladat három nehézségi szinten: könnyű, közepes és nehéz (legkisebb korosztályban 12 feladat);
- A verseny alatt semmilyen más számítógépes program, alkalmazás nem használható;
- A verseny során nyugalmas környezetet kell biztosítani;
- A terem a verseny során nem hagyható el;
- Az esetleges számítógéppel, internettel kapcsolatos észrevételeket a kontakt személynek kell összegyűjtenie és továbbítani a szervezők felé;
- A verseny célja: minél több pont összegyűjtése helyes válaszok megjelölésével, helytelen válaszok esetén pontlevonás történik;
- A kérdések tetszőleges sorrendben megválaszolhatók;
- A kérdések, problémák megértése a feladat részét képezi. Ezért a feladatok megbeszélése és értelmezéssel kapcsolatos kérdések nem megengedettek;
- A megoldások a verseny befejezése után, a hód hetet követően kerülnek nyilvánosságra.



ÉRTÉKELÉS, PONTOZÁS

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

Nem megválaszolt kérdés esetében az összpontszám változatlan marad.

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

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

Összesen (18 feladat esetében) maximum 162 pont érhető el.



TARTALOMJEGYZÉK

Mi is az E-HÓD?.....	2
Részvétel	3
Szabályok.....	4
Értékelés, Pontozás.....	5
Tartalomjegyzék.....	6
Feladatok	8
Golyópálya (2019-AT-03).....	9
Négy Hal (2019-AT-04).....	11
Minták (2019-AT-05)	13
Fűzfák és nyárfák (2019-BE-03)	15
Fűrésztelep (2019-BE-06).....	17
Különleges tornyok (2019-CA-01)	19
Nyelvhatározó (2019-CA-03).....	21
Nyári Munka (2019-CA-04)	23
Hódérmék (2019-CH-03b)	25
Az őshódok üzenete (2019-ch-10)	27
Füstjelek (2019-CH-11c).....	29
Füstjelek (2019-CH-11d)	31
Nyomda (2019-CH-13d).....	33
Strand (2019-CH-18).....	35
Színes kínai karakterek (2019-CN-03a)	37
Parkolás (2019-DE-02).....	39
Monitorok (2019-DE-03).....	41
Takarítás (2019-DE-04).....	43
A csapda (2019-DE-05)	45
Celeb-helyzet (2019-DE-08)	47
Könyvesbolt (2019-HU-01).....	49



Bombonos zacskó (2019-HU-02)	51
Könyvtári betörés (2019-HU-03)	53
Alkimisták (2019-HU-04)	55
Díszítsünk(2019-HU-05)	57
Hódanyó és Turing (2019-HU-06)	59
Rangoli (2019-IN-09).....	61
Szép, hogy vannak fák (2019-IT-01)	63
Kipu (2019-JP-03)	65
Karcpapír (2019-KR-01).....	67
Hamburger hozzávalók (2019-KR-07).....	69
Hangyák az ingoványban (2019-LT-06)	71
Nyalóka(2019-PK-01)	73
Igaz vagy hamis (2014-RU-02)	75
Tányérok (2019-RU-01)	77
Videó tömörítés (2019-RU-02)	79
Erdei fafélék (2019-SI-02)	81
Utazás a világegyetemen keresztül (2019-SI-03).....	83
Torony (2019-SK-03).....	85
Robot (2019-SK-04)	87
Hódháló (2019-Th-08)	89
Kincses térkép (2019-VN-04)	91
Támogatóink, köszönetnyilvánítás	93



FELADATOK

- Kishód:** 2019-CH-03b, 2019-CH-11c, 2019-CH-13d, 2019-CH-18
2019-DE-02, 2019-DE-04, 2019-KR-01, 2019-PK-01
2019-SI-02, 2019-SI-03, 2019-HU-01, 2019-HU-05
- Benjámin:** 2019-CA-01, 2019-CH-03b, 2019-CH-10, 2019-CH-11c, 2019-CH-18, 2019-CN-03a
2019-DE-02, 2019-DE-04, 2019-DE-08, 2019-IN-09, 2019-LT-06, 2019-RU-01
2019-SI-02, 2019-SI-03, 2019-SK-03, 2019-HU-01, 2019-HU-05, 2019-AT-05
- Kadét:** 2019-CA-01, 2019-CH-10, 2019-CH-11d, 2019-CN-03a, 2019-DE-03, 2019-DE-05
2019-DE-08, 2019-HU-02, 2019-IN-09, 2019-JP-03, 2019-KR-07, 2019-LT-06
2019-RU-01, 2019-SI-02, 2019-SK-03, 2019-TH-08, 2019-HU-03, 2019-AT-05
- Junior:** 2019-CH-11d, 2019-CN-03a, 2019-DE-03, 2019-DE-05, 2019-DE-08, 2019-HU-02
2019-IT-01, 2019-JP-03, 2019-KR-07, 2019-LT-06, 2019-RU-02, 2019-SK-04
2019-TH-08, 2019-HU-04, 2019-HU-06, 2019-AT-05, 2019-BE-03, 2019-CA-03
- Senior:** 2019-AT-03, 2019-AT-04, 2019-BE-06, 2019-CA-04, 2019-CH-11d, 2019-DE-05
2019-HU-02, 2019-IT-01, 2019-JP-03, 2019-KR-07, 2019-RU-02, 2014-RU-02
2019-SK-04, 2019-TH-08, 2019-VN-04, 2019-HU-04, 2019-HU-06, 2019-AT-05



GOLYÓPÁLYA (2019-AT-03)

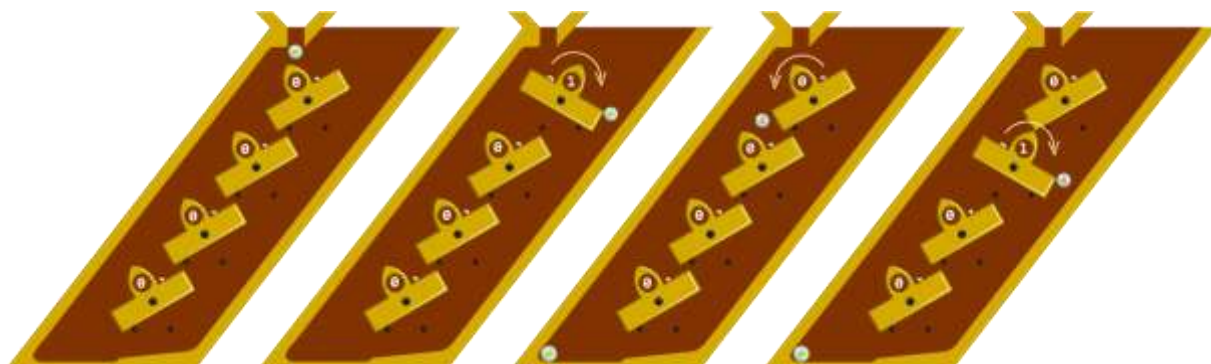
SENIOR – NEHÉZ

Egy golyópálya négy libikókából áll, melyek mindegyike kétféle állapotban lehet:

- ha a libikóka balra dől, akkor az a 0-s állapot
- ha a libikóka jobbra dől, akkor az a 1-es állapot.

Amikor egy golyó ráesik a libikókára, megváltoztatja az állapotát és a golyó legurul.

Amikor két golyót gurítunk egymás után a pályára a libikókák a következőképpen változnak:



A végén a libikókák **lentről felfele** olvasva a következő állapotokban vannak: 0, 0, 1, 0.

Kezdetben minden libikókát a 0-s állapotba állítunk.

Milyen állapotban lesznek a libikókák (lentől felfele olvasva), ha 10 golyót gurítunk egymás után a pályára?

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



„D” válasz a helyes

A legfelső libikóka minden golyónál megváltoztatja az állapotát. Tehát ha, mint most is, páros számú golyóval találkozunk, az állapota 0 lesz. Így az utolsó számjegyünk a válaszban 0.

Felülről a második libikóka csak akkor változtatja az állapotát, amikor a felette lévő 1-es állapotban volt. Tehát ebben az esetben ezt ötször teszi meg, így az 1-es állapotban lesz a végén. Ezért az utolsó előtti számjegyünk 1.

Felülről a harmadik libikóka csak akkor változtatja az állapotát, ha a felette lévő (felülről a második) libikóka(?) 1-es állapotban volt és „golyó éri”, azaz a legfelső libikóka is 1-es állapotban volt. Ezért ez a libikóka csak minden negyedik golyónál változtatja az állapotát, azaz a negyedik és a nyolcadik golyónál. Tehát kétszer, így hátulról a harmadik számjegyünk is 0 lesz.

A legalsó libikóka csak akkor változtatja az állapotát, ha mindhárom felette elhelyezkedő libikóka 1-es állapotban volt. Ez csak a nyolcadik golyó esetében lesz így, tehát csak egyszer történik meg. Ezért az első számjegyünk 1 lesz.

Összességében a libikókák állapota: 1, 0, 1 és 0, tehát a D a helyes válasz.

MIÉRT INFORMATIKA?

A golyópálya libikókái egy elektronikus kapcsolóegységet jelenítenek meg, mely két állapot között tud ide-oda kapcsolni. Az ilyen kapcsolóegységek az elektronikus eszközök alap építőkövei, ebben az esetben egy úgynevezett Flip-flop változat.

A libikókák együtt a kettes számrendszerben számoló számológépként működnek. Itt a flip-flop félösszeadóként van összekapcsolva. Egy ilyen félösszeadó bemenete a flip-flop eltárolt állapota és egy bejövő impulzus. Kimenete az új, eltárolt állapota és egy átviteli értéke.

eltárolt állapot	impulzus	újonnan tárolt állapot	átvitel
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A kapcsolásnak biztosítania kell, hogy a tárolt állapotok változásai megfelelő sorrendben következnek be és egy változást követően önállóan nem változnak ismét. A libikókák állapotai tehát egy kettes számrendszerbeli (bináris) számot reprezentálnak, melyet minden golyó eggyel növel meg.

KULCSSZAVAK

Bináris számrendszer, félösszeadó, flip-flop, számológép

WEBOLDALAK

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



NÉGY HAL (2019-AT-04)

SENIOR – NEHÉZ

Az informatikában az olyan műveletek, mint az + vagy a *, végrehajtásának módja részben attól függ, milyen típusú adatok szerepelnek benne. A következő táblázat a kifejezések különböző típusokkal való kombinációit mutatja:

Általános leírás	Példa
Szám + Szám → Szám (összeadás)	2+3 → 5
Szám + Szöveg → Hiba	2+"3" → Hiba
Szöveg + Szám → Hiba	"2"+3 → Hiba
Szöveg + Szöveg → Szöveg (összefűzés)	"2"+"3" → "23"
Szám * Szám → Szám (szorzás)	2*3 → 6
Szám * Szöveg → Szöveg (Szám-szor a Szöveg összefűzve)	2*"3" → "23"
Szöveg * Szám → Szöveg (a Szöveg Szám-szor összefűzve)	"2"*3 → "222"
Szöveg * Szöveg → Hiba	"2"*"3" → Hiba

Amikor „Hiba” áll eredményként, az azt jelenti, hogy arra a kombinációra nincs meghatározva művelet. Ha egy kifejezésen belül hiba lép fel, akkor az egész kifejezés hibával ér véget.

A műveletek sorrendjénél a * előbb kerül végrehajtásra, mint az + (mint a matematikában megszokott). Zárójelekkel azonban ezeket módosíthatjuk, hiszen mindig a zárójelben lévő kifejezést hajtjuk végre előbb.

Az alábbi kifejezések közül melyiknek lesz a következő szöveg az eredménye?

"...>(((°>.....>(((°>.....>(((°>.....>(((°>..."

- A) $(3^{**}."+">"+3^{**}("+"^{\circ}>"+3^{**}."))^{**}2^{**}2$
- B) $(3^{**}."+">"+3^{**}("+"^{\circ}>")^{**}2^{**}2+3^{**}."$
- C) $(3^{**}."+">"+3^{**}("+"^{\circ}>"+3^{**}."))^{**}2^{**}2$
- D) $(3^{**}."+">"+3^{**}("+"^{\circ}>"+3^{**}."))^{**}2^{**}2$



„D” válasz a helyes

Az öt „összeadandó” a zárójelben sorrendben az alábbiakat adja:

- $3 * "." \rightarrow "..."$
- $"><" \rightarrow "><"$
- $3 * "(" \rightarrow "(((("$
- $"^>" \rightarrow ">"$
- $3 * "." \rightarrow "..."$

Ezeket összefűzve: $"...>(((^>..."$

Az ezt követő $2*2$ szorzás azt jelenti, hogy a szövegünket kétszer összefűzzük, majd mindezt ismét kétszer egymás után fűzzük: $"...>(((^>.....>(((^>.....>(((^>.....>(((^>..."$

Az A) válaszban hibát kapunk, mivel a zárójelben létrejött kifejezés egy szöveg, amit egy szöveggel akarunk szorozni (*).

A B) válasz a következő szöveget hozza létre: $"...>(((^>...>(((^>...>(((^>...>(((^>..."$

Ez elég közel van a kívánt megoldáshoz, de a halak közötti pontok száma nem egyezik.

A C) válasz hibát eredményez, mivel a $3 * "("$ rész eredménye hiba. Tehát az egész kifejezés hibás lesz.

MIÉRT INFORMATIKA?

A műveletekhez (vagy akár az alprogramokhoz is) az operandusoktól (vagy paraméterektől) függően különböző végrehajtást rendelhetünk hozzá. Ezt hívják műveleti túlterhelésnek (operator overloading). Mindenekelőtt olyan műveleteknél fordul elő, melyek egy programozási nyelven kívül is gyakran használatosak. A feladatban leírt + és * műveletek túlterhelése sok programozási nyelvben előfordul. Amilyen praktikus a műveletek túlterhelése, annyira veszélyes is, a programkód olvashatóságának rovására megy.

A következő kis program ugyanazt a szöveget állítja elő, mint a feladatban (a halat), de azt, hogy melyik változóban milyen típusú adatot tárolunk és melyik művelet hogyan működik, csak fáradtságosan kibogozható.

$a = 3$	$b = "."$	$c = "><"$
$d = "("$	$e = ">"$	$f = 2$

Feladat: $(a*b+c+a*d+e*a*b)*f*f$

De a kifejezések különböző jelentéstartalommal való „terhelése” nem csak az informatikában figyelhető meg. A természetes nyelvben ezt polisziemiának (többjelentésű szavaknak) nevezzük. Például a „futó” szavunk környezettől függően más jelentéssel bír: a sportoló, bábu a sakkban, szőnyeg, ...

A feladatban használt hal, egy klasszikus példa az ASCII művészetre is. A „vörös hering” egyik változata, melyet az interneten sokszor arra használnak, hogy jelezzék a trollkodást.

KULCSSZAVAK

Műveleti túlterhelés, polisziémia

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Polimorfizmus>, https://en.wikipedia.org/wiki/ASCII_art



MINTÁK (2019-AT-05)

BENJAMIN – NEHÉZ

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

SENIOR – KÖNNYŰ

Hódani egy olyan rendszerrel játszik, amelyik utasítások alapján mintákat rajzol ki. Az alapértelmezett kezdeti rajzirány a jobb. Ezt az alábbi utasítások követhetik:

- `előre(XX)`: XX cm-t halad a jelenlegi rajzolási irányba
- `balra()`: a rajzolási irányt 90° fokkal balra fordítja
- `jobbra()`: a rajzolási irányt 90° fokkal jobbra fordítja

A következő utasítássorozat: `“előre(20), balra(), előre(10), jobbra(), előre(10)”` eredménye (az alapértelmezett kezdeti irányból):



Adott az alábbi mintarajzoló utasítássorozat:

`“..., előre(20), balra(), előre(10), balra(), előre(10), balra(), előre(20), ...”`

Az alábbi minták közül melyik készülhetett annak a mintának a felhasználásával, amit a fent megadott részlet (utasítássorozat) rajzol ki?

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



„A” válasz a helyes

A kiinduló irányból az utasításokat végrehajtva, ehhez hasonló mintát kapunk:



A helyes mintában ennek a részletnek kell megjelennie, akár a 90° fok többszörösével elforgatva.

Az A válasz az egyetlen, amelyikben egymást keresztező vonalak szerepelnek. Valójában, a kirajzolt részlet 90° fokkal balra, illetve 180° fokkal jobbra elforgatva is megjelenik az ábrán, tehát a helyes válasz az: A.).

MIÉRT INFORMATIKA?

A szoftverfejlesztő munkájának fontos része, hogy értelmezni tudja az utasítássorozatokat (kód), és elképzelése legyen a végrehajtás eredményéről.

Annak ellenére, hogy egyetlen parancs hatása világos és egyértelmű, sokszor nagyon bonyolult az egyszerű parancsokból alkotott programrészek (szekvenciák) megértése.

KULCSSZAVAK

Szekvencia, kódolás

WEBOLDALAK

https://hu.wikipedia.org/wiki/Struktur%C3%A1lt_programoz%C3%A1s



FŰZFÁK ÉS NYÁRFÁK (2019-BE-03)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Berti faültetésbe kezd, hogy a meleg nyári napokra árnyékról gondoskodjon. A fűzfa és a nyárfa ültetésén gondolkodik, de tudni szeretné, melyik nő gyorsabban. Ahhoz, hogy több információhoz jusson a tabletje kereső alkalmazását használja. Az alkalmazás kibővített keresője két speciális szimbólummal dolgozik.

... # ...

A # jel két oldalán lévő feltételeket vizsgálja. Csak azokat a találatokat adja eredményül, amelyekben **MINDKÉT** feltétel teljesül.

[...]

A **zárójel** közötti kifejezést invertálja. Csak azokat a találatokat adja eredményül, melyekben **NEM** teljesül a keresési feltétel.

Például:

keresőfeltétel	eredmény
fűzfa	Fűzfákról szóló oldalak.
[fűzfa]	NEM fűzfákról szóló oldalak.
fűzfa # nyárfa	Olyan oldalak, melyek a fűzfáról ÉS a nyárfáról egyidejűleg tartalmaznak információkat.
[[nyárfa]]	Nyárfákról szóló oldalak. (= NEM a NEM a nyárfákról szóló oldalak).
[fűzfa # nyárfa]	Oldalak melyek NEM szólnak egyidejűleg MINDKÉT fáról.
fűzfa # [nyárfa]	Oldalak melyek a fűzfákról szólnak, DE a nyárfáról NEM.

Az alábbi keresőfeltételek közül, melyikkel találjuk meg azokat és csak azokat az oldalakat, melyek (csak) a fűzfáról, vagy (csak) a nyárfáról vagy a nyárfáról és a fűzfáról egyidejűleg tartalmaznak információkat?

- A) [fűzfa] # [nyárfa]
- B) [[fűzfa] # nyárfa] # [fűzfa # [nyárfa]]
- C) [[fűzfa] # [nyárfa]]
- D) [[fűzfa] # [nyárfa]] # [fűzfa # nyárfa]



„C” válasz a helyes

Ahhoz, hogy megértsük, miért ez a helyes válasz, tekintsük a valamivel egyszerűbb 'A' választ: [fűzfa] # [nyárfa]. Ez a keresési feltétel az összes olyan oldalt megtalálja, mely eleget tesz annak a feltételnek, hogy [fűzfa] ÉS [nyárfa], vagyis NEM tartalmaznak információt a fűzfáról ÉS ugyanakkor NEM tartalmaznak információt a nyárfáról sem, vagyis azok az oldalak, melyek sem a fűzfáról, sem a nyárfáról nem szólnak. Ez számunkra nem lenne megfelelő válasz, tulajdonképpen, pont az ellentéte annak, amit Berti keres. Az "ellentéte" vagyis a "[...]"-ja. Tehát, az A keresési feltétel szögletes zárójelek közé helyezésével pontosan a keresett feltételt adhatjuk meg, ami történetesen a 'C' válasz.

MIÉRT INFORMATIKA?

A képzeletbeli keresőnkhez hasonlóan, a modern keresőmotorok is alkalmasak a keresőfeltételek különböző összekapcsolására. Ehhez a NOT, AND vagy OR kulcsszavakat használhatjuk a NEM, ÉS, VAGY kapcsolatok kifejezéséhez. Berti tulajdonképpen a fűzfa VAGY nyárfa kifejezést akarja használni, de a keresője a VAGY kulcsszót nem ismeri. Berti kéréséhez a VAGY kapcsolót a NEM és az ÉS kapcsoló segítségével fejezzük ki.

Általánosságban: **a VAGY b = NEM (NEM(a) ÉS NEM(b))**

A matematikai logika tudományában ezekre De Morgan azonosságokként hivatkozunk.

Programozás közben számos logikai kifejezést alkalmazunk. A De Morgan azonosságok ismeretével egyszerűbbé válik a komplex algoritmusok elkészítése és megértése.

A VAGY logikai kapcsolót körültekintően kell kezelnünk, mert két különböző jelentés társítható hozzá. Az egyik helytelen válaszunk - a 'D' - a kizáró VAGY (angolul XOR) eredményét adja. Fűzfáról vagy nyárfáról szóló oldalakat, de kizárja azt, hogy mindkettőről egyszerre szóljanak. A mindennapi életben inkább ezzel a jelentéssel használjuk a VAGY kifejezést. Ha valaki megkérdezi, hogy pizzát, vagy sültet ennél ebédre, általában nem azt válaszoljuk, hogy mindkettőt.

A 'C' válasz a megengedő VAGY kapcsolót mutatja be: oldalakat eredményez a fűzfáról, a nyárfáról, vagy mindkettőről egyszerre.

Talán furcsának találod, hogy a feladatbeli keresőnk a # szimbólumot használta az ÉS kifejezésére – talán a '&' jel használata érthetőbb lett volna – és a [] zárójelet a NEM kifejezésére. A számítógépek számára nincs furcsaság, azt teszik, amire programozták őket. A programozók általában olyan egyezményes jelöléseket használnak, melyeket – a félreértések elkerülésének érdekében – az emberek könnyen megértenek, de a számítógépek ez nem igazán izgatja: a számítógépünk számára a # jel is éppen olyan jól megfelel, mint bármely másik.

KULCSSZAVAK

De-Morgan azonosságok, logikai műveletek

WEBOLDALAK

https://hu.wikipedia.org/wiki/De_Morgan-azonoss%C3%A1gok



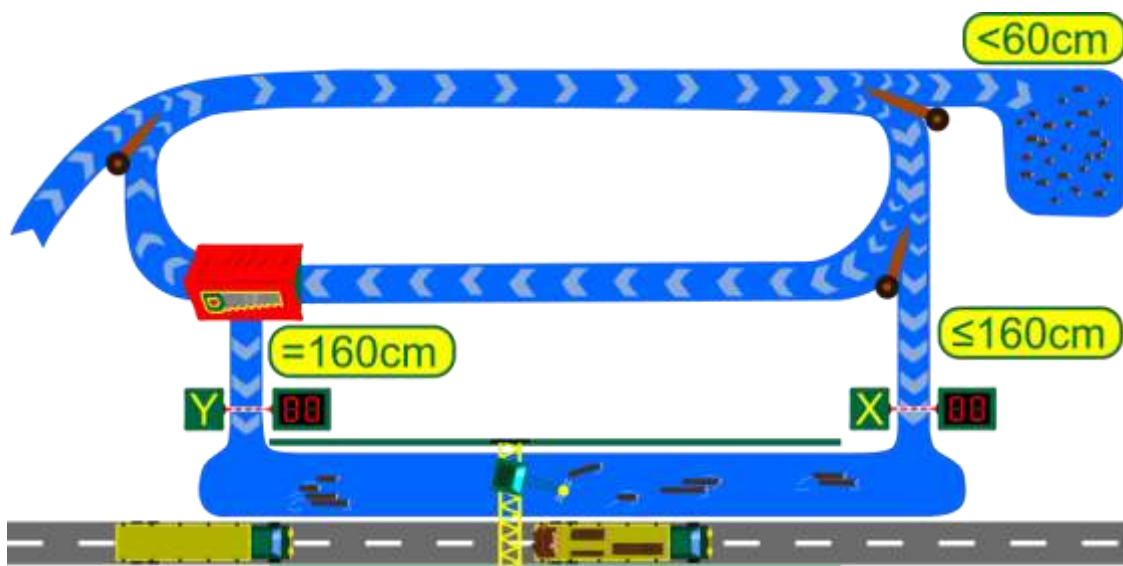
FŰRÉSZTELEP (2019-BE-06)

SENIOR – KÖZEPES

Egy fűrésztelepen a fatörzseket 60 cm és 160 cm közötti hosszúságúra vágják, azután kamionra rakják. A fűrésztelepen belül a fatörzseket csatornában szállítják.

A következő a feldolgozási folyamat:

- A **bal felső** csatornán érkeznek a fatörzsek.
- A **jobb felső** csatornában vezetik ki a 60 cm-nél rövidebb fatörzseket.
- **Lent jobbra** az összes 160 cm-es vagy ennél rövidebb fatörzset kamionra rakják. Ezeket számolja az X érzékelő.
- **Lent balra** minden fatörzsről levágnak egy 160 cm-es darabot. A levágott darabot a kamionra rakják és ezt az Y érzékelő számolja. A fatörzs maradéka pedig ismét bekerül a körforgásba.



Egy 60 cm-es, egy 140 cm-es és egy 360 cm-es fatörzset szállítanak be és dolgoznak fel a fűrésztelepen.

Hány fatörzset számol meg az X és hányat az Y érzékelő?

- Az X érzékelő egyet sem, az Y érzékelő 4 fatörzset.
- Az X érzékelő 1, az Y érzékelő 3 fatörzset.
- Az X érzékelő 2, az Y érzékelő 2 fatörzset.
- Az X érzékelő 3, az Y érzékelő 1 fatörzset.



„C” válasz a helyes

A 60 cm-es fatörzs a jobb felső csatornánál nem kerül kiválasztásra, mivel nem rövidebb, mint 60 cm. De a jobb alsó kiválasztásnál felkerül a kamionra, mivel 160 cm-nél rövidebb. Így az X érzékelő ezt a fatörzset megszámlolja. A 140 cm hosszú fatörzs ugyanúgy nem kerül kiválasztásra a jobb felső leágazásnál, hiszen 60 cm-nél hosszabb, de a jobb alsó kiválasztásnál szintén felkerül a kamionra, mivel 160 cm-nél rövidebb. Így az X érzékelő még egy fatörzset számol.

A 360 cm hosszú fatörzs ugyanúgy nem kerül kiválasztásra a jobb felső leágazásnál, hiszen 60 cm-nél hosszabb, és a jobb alsó kiválasztásnál nem kerül fel a kamionra, mivel 160 cm-nél hosszabb. A bal alsó üzetrészben levágnak 160 cm-es darabot, melyet az Y érzékelő megszámlol. A 200 cm-es darab ismét bekerül a körforgásba. Nem kerül kiválasztásra a jobb felső leágazásnál, hiszen hosszabb, mint 60 cm. A jobb alsó leágazásnál is tovább halad, mivel hosszabb, mint 160 cm. A bal alsó üzetrészben levágnak belőle egy 160 cm-es darabot, melyet az Y érzékelő megszámlol és felkerül a kamionra. A maradék 40 cm-es darab ismét bekerül a körforgásba. De ezt a darabot a jobb felső leágazásnál kisselektálják, hiszen 60 cm-nél rövidebb.

MIÉRT INFORMATIKA?

A körforgásban csak a fatörzsek hossza fontos. A fűrésztelepet programként is tekinthetjük, melynél egész számokat adunk meg és meghatározott méréseket végzünk. Így a fűrésztelep mint reaktív program működhet: mialatt a program megváltoztatja a számot, idővel megváltozik a mérés eredménye. Reaktív programozásra példát elsősorban a táblázatkezelőknél láthatunk. A táblázatban képletekkel kiszámolt értékek reagálnak más cellák értékeinek megváltozására.

A feladatban több reaktív műveletet használunk: fent balra két adatfolyamot egyesítünk (merge), fent jobbra és lent jobbra kiszűrjük a bizonyos tulajdonságúakat (filter), lent balra módosítunk (transform) az adatainkon. A két érzékelő végzi a méréseket (scan).

A dinamikus folyamatok elemzése – mint ebben a feladatban is – az informatika egyik alapvető (fő) tevékenysége.

KULCSSZAVAK

Dinamikus folyamatok, folyamatelemzés

WEBOLDALAK

https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0052_37_modern_programozasi_mintak_webprogramozasban/lecke9_lap1.html



KÜLÖNLEGES TORNYOK (2019-CA-01)

BENJAMIN – NEHÉZ

KADÉT – KÖNNYŰ

Egy torony akkor különleges, ha minden tőle balra álló torony alacsonyabb és minden tőle jobbra álló magasabb nála.



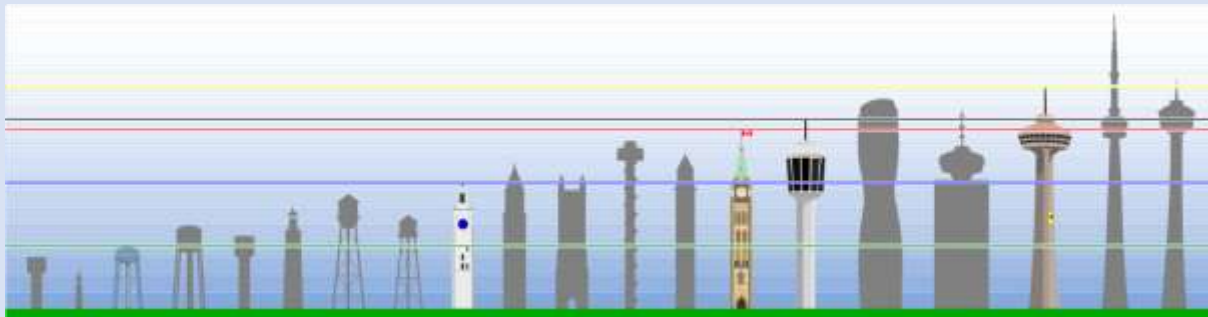
A rajzon hány torony különleges?

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



„B” válasz a helyes

A képen látható vonalak segítenek meglátni, hogy a következő öt torony különleges, tehát a B a helyes válasz.



MIÉRT INFORMATIKA?

Ebben a feladatban a tornyokat a magasságuk szerint hasonlítjuk össze. Ilyen hasonlításokat a keresési és a rendezési feladatoknál találhatunk, melyek az informatika részterületei.

Többféle rendező algoritmus létezik, melyek különböző alkalmazásokhoz megfelelőek. Az egyik ismert és gyors rendezési algoritmus például a Quick-sort, melynek egy lényeges pontja az, hogy egy adott elemtől jobbra elhelyezkedők mind kisebbek, a balra lévők pedig mind nagyobbak. Ez az elem a rendezendő elemeket két alcsoportra osztja, így a rendezési problémát két kisebb rendezési problémára bontja. Ezt a köztes elemet hívják pivot, vagy fő- illetve vezérelemnek. Ha a jobbra állók nem mind kisebbek és a balra állók nem mind nagyobbak, akkor cserékkel kell megoldanunk a helyes sorrendet. Ezt az eljárást kell megismételni minden részterületre egészen addig, amíg csak egy elem marad... ami pedig rendezett. Ezt a rekurzív eljárást, amikor a nagy problémát kisebb problémára vezetjük vissza, „oszd meg és uralkodj” elvnek hívjuk. Igen elterjedt nehéz problémák megoldására.

A quick-sort (gyorsrendező) algoritmus sok más rendező-algoritmushoz képest gyorsabb, ahogy a neve is mutatja. Ez amiatt van, hogy normál esetben a pivot (vezér)elem kiválasztásával a rendezendő részsorozataink feleződnek.

KULCSSZAVAK

Gyorsrendezés, algoritmus

WEBOLDALAK

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



NYELVHATÁROZÓ (2019-CA-03)

JUNIOR – KÖNNYŰ

Felfedezők egy csoportja az alábbi ősi szavakba botlott egy barlang falán:

paqrooob puue t'se grub meoub lai' lai qy

A felfedezők egy rendszer segítségével próbálták meghatározni, hogy az egyes szavak mely nyelvhez tartoznak:

- minden szót egy kezdeti 10-es értékkel láttak el
- ezt az értéket az alábbi szabályok módosítják:

p-vel kezdődik	-2
b-re végződik	-2
6 karakternél hosszabb	+3
a q-t közvetlenül r vagy y követi	-4
ha három magánhangzó (a,e,i,o,u) szerepel egymás mellett	+5
tartalmazza a (') jelet	+1

- **Ha a végső érték 10, vagy 10-nél több,** akkor a rendszer a szót a **Hódoló** nyelvbe sorolja
- **Különben** a **Hódító** nyelvbe tartozik.

Például: a palliob szó értéke az alábbiak szerint alakul $10 - 2 - 2 + 3 = 9$, vagyis a palliob szó a Hódító nyelvbe tartozik.

A megadott rendszert használva, hány barlangbeli szó sorolható a Hódító nyelv szavai közé?

- A) Egyik sem.
- B) 1
- C) 2
- D) 3



„B” válasz a helyes

Az alábbi leírás a szavakhoz rendelt értékeket és a szó adott nyelvbe való besorolását összegzi:

paqrooob:	$10-2-2+3-4+5+0=10$	→	Hódoló
puue:	$10-2-0+0-0+5+0=13$	→	Hódoló
t'seqrub:	$10-0-2+3-4+0+1=8$	→	Hódító
meoub:	$10-0-2+0-0+5-0=13$	→	Hódoló
lai'laigy:	$10-0-0+3-0+0+1=14$	→	Hódoló

Ebből megbizonyosodhatunk arról, hogy egyetlen egy szó tartozik a Hódító nyelvbe.

MIÉRT INFORMATIKA?

A természetes (emberi) nyelvekkel kapcsolatos kutatások izgalmas és aktuális területei az informatikának. Ide tartozik a beszédhang írottá konvertálása (alakítása), és a gépelt szöveg hanggá alakítása is. Minden bizonnyal többen használunk otthonunkban, vagy zsebeinkben olyan eszközöket, melyek ezt a technológiát használják. Egy másik gyakori példa a szövegek egyik nyelvről egy másik nyelvre való fordítása. Ha az eredeti nyelvet nem ismerjük, az első lépés az ismeretlen nyelv meghatározása. Ez a fenti feladat lényege is.

Általánosságban elmondhatjuk, hogy a nyelvfelismerés problémája a természetes nyelvfeldolgozás területén nagyon nehéz, és a tökéletes megoldást (még) nem ismert. A feladat megoldása során a heurisztikát (vö. ökölszabály) alkalmaztuk. Ez azt jelenti, hogy olyan szabályokat használunk, melyektől azt reméltük, az esetek nagy részében a helyes következtetéshez vezetnek minket. A heurisztika a korábban megszerzett tapasztalatokon alapuló megoldáskeresési módszer, mely ámulatba ejtően jó megoldásokkal szolgál. Pszichológusok dolgoznak olyan alkalmazásokon, melyek megállapítják egy író (közlő) életkorát vagy nemét, vagy éppenséggel megállapítják azt, ha valaki hazudik.

KULCSSZAVAK

Heurisztika, ökölszabály, nyelv

WEBOLDALAK

<https://hu.wiktionary.org/wiki/%C3%B6k%C3%B6lszab%C3%A1ly>



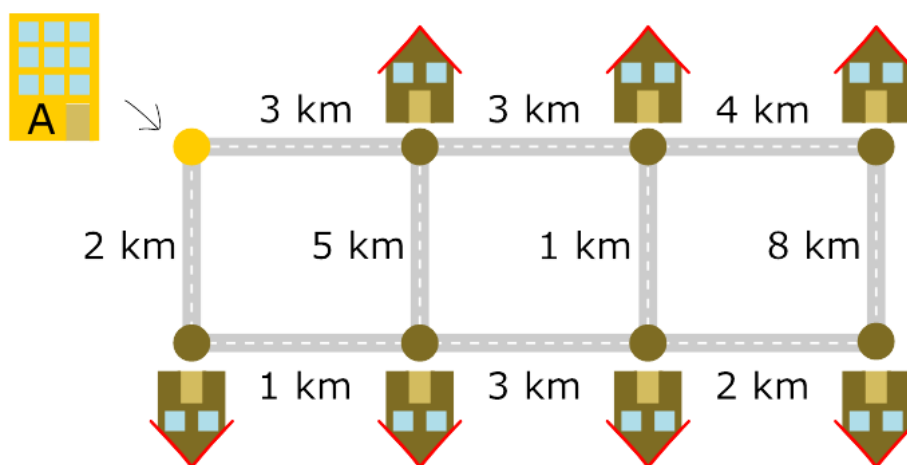
NYÁRI MUNKA (2019-CA-04)

SENIOR – NEHÉZ

Nyári munkaként csomagokat szállítasz ki. Az „A” helyen kezdted és a térképen látható **mind a hét helyre** egy-egy csomagot viszel. Az utolsó helyen fejezed be az utadat.

A munkaadód a csomagokkal megtett összes utat fizeti ki neked.

Az egyes utak hossza a térképen látható. A munkaadód szabad választást enged számodra addig, amíg **nem mész kétszer ugyanarra a helyre**.



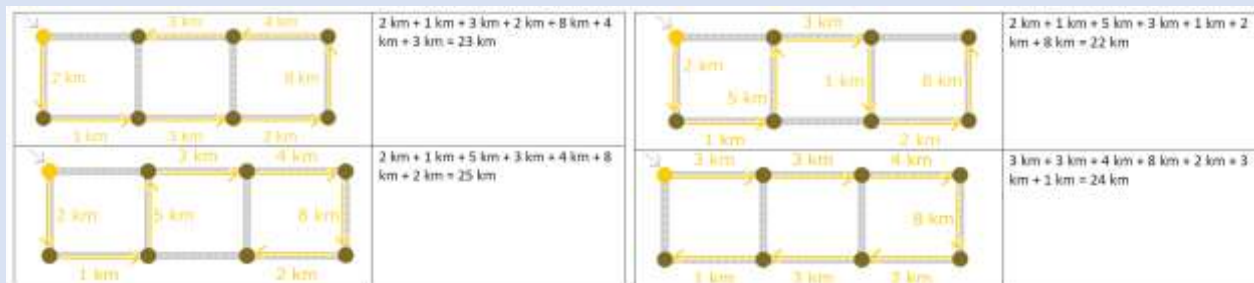
Mekkora a leghosszabb összes út, melyet a munkaadód kifizet?

- A) 22
- B) 23
- C) 24
- D) 25



„D” válasz a helyes:

Összesen csak négy olyan lehetséges út létezik, amikor nem megyünk kétszer ugyanarra a helyre. A hosszuk:



MIÉRT INFORMATIKA?

Ebben a feladatban egy utat kell találnunk, amellyel pontosan egyszer minden helyet meglátogatunk. Az ilyen utat Hamilton-útnak nevezzük. Mivel a térképet, mint gráfot, a helyeket, mint csomópontokat és az utakat, mint éleket (kapcsolatokat) jeleníthetjük meg, a Hamilton-út megtalálása egy gráfelméleti probléma. Egy tetszőleges gráfban egy Hamilton-út megtalálása, illetve annak vizsgálata, hogy létezik-e ilyen út, egy híres NP-teljes probléma, melyet még a számítógépek sem tudnak hatékonyan megoldani.

Ebben a feladatban nem csak egy utat keresünk, hanem az összes lehetséges közül a legnagyobb értékűt, melyet a térképen jelölt értékek összegeként kaphatunk meg. Ez még érdekesebbé teszi a problémát.

Ebben az esetben olyan lehetőségek is kiesnek, melyeket más esetben részben optimális megoldásként elfogadnánk. Egy klasszikus megközelítés lenne a greedy-algoritmus (más néven mohó algoritmus), melyben a lehető legnagyobb értékeket gyűjtjük: tehát elsőként a 3 km-es utat választjuk. Ezzel az eljárással csak a 24 km-es utat találánánk meg. Mivel ha „mohón” másodikként az 5 km-es utat választanánk, kizárnánk a lehetőségét annak, hogy minden helyre eljussunk.

KULCSSZAVAK

Gráfok, Hamilton-út, mohó algoritmus

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Hamilton-%C3%BA>

https://hu.wikipedia.org/wiki/Moh%C3%B3_algorithmus



HÓDÉRMÉK (2019-CH-03B)

KISHÓD – KÖZEPES

BENJAMIN – KÖNNYŰ

Hódországban használt pénzt hódérmének hívják. Az érmék értékei a következők:

				
16	8	4	2	1

A hódok nem szeretnek sok érmét magukkal vinni, ezért mindig a lehető legkevesebb érmével fizetnek.

Legkevesebb hány darab érmével tudnak 13 hódpenzt kifizetni?

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



„C” válasz a helyes

A legjobb és ezért helyes megoldás az (1x8, 1x4 és az 1x1 hódérme). Két lehetséges stratégiával is megtalálhatjuk a helyes megoldást:

- 1) 13 db 1-es érmével kezdünk és ameddig csak lehetséges, egy pár kisebb érmét egy következő nagyságúra cserélünk.

1,1,1,1,1,1,1,1,1,1,1,1,1

2,2,2,2,2,2,1

4,4,4,1

8,4,1

- 2) A másik oldalról közelítjük meg a megoldást és kiválasztjuk a lehető legnagyobb érmét, amelyik egyszer „belefér” a fizetendő összegünkbe. Az ezzel csökkentett összegre megint megkeressük a legnagyobb lehetséges érmét. Mindezt addig folytatjuk, amíg nullát nem kapunk végül.

8 a legnagyobb, 13-nál kisebb érménk. A maradék 5.

4 a legnagyobb, 5-nél kisebb érménk. A maradék 1.

1 éppen 1.

A második megoldás több esetben gyorsabb, így sokszor azt célravezetőbb használni.

MIÉRT INFORMATIKA?

Az informatikusok szakemberek az információk szimbólumok sorozataként való ábrázolásában, például a számok esetében is. Ebben a feladatban egy számot bármilyen érmegyűjtemény segítségével ábrázolhatunk, azzal a tulajdonsággal, hogy az érmék értékeinek összege megegyezik a kijelölt számmal. Ez a megfeleltetés nem minden esetben egyértelmű. Lehetséges, hogy az érmék különböző kombinációja ugyanazt az összeget fogja kiadni.

Ebben a feladatban ugyanazzal az értékkel rendelkező két érme a következő nagyobb érmére cserélhető.

Ez a kettes számrendszert határozza meg az 1, 2, 4, 8, 16-os helyiértékekkel. A kettes számrendszerben egy tetszőleges szám ábrázolása mindig egyértelmű: egy helyiértéket használunk vagy sem.

A híres Abakus, a már évszázadok óta használt számológép, ugyanezen az elven működik.

KULCSSZAVAK

Bináris számok, Abakus, számábrázolás

WEBOLDALAK

https://hu.wikipedia.org/wiki/Kettes_sz%C3%A1mrendszer



AZ ŐSHÓDOK ÜZENETE (2019-CH-10)

BENJAMIN – NEHÉZ

KADÉT – KÖZEPES

Dara hód a gát mélyén egy ősi fát talált. Közelebbről nézve felfedezi a fába faragott misztikus jeleket.

Dara úgy gondolja, hogy ez a kódtábla abból az időből van, amikor az őshódok még itt a gátban éltek.

I	II	III	III	○	○	○	○	○
A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	

A tábla tanulmányozása során Dara rájön, hogyan működik: az ismeretlen jelek azoknak a szimbólumoknak a kombinációi, melyeket az oszlopok és a sorok adnak meg.

Például: A <H> betű kódolása a következő:

I	II	III	III	○	○	○	○	○
A	B	C	D	E	F	G	H	I
J	K	L	M	N	O	P	Q	R
S	T	U	V	W	X	Y	Z	

Dara emlékszik, hogy ilyen jeleket már látott a gát másik oldalán is. Odamegy, és ezt találja:



Melyik üzenetet írták az őshódok?

- A) FIGYUZZ
- B) FIGYELJ
- C) VIGYAZZ
- D) VERSELJ



„B” válasz a helyes



Az első jel a és szimbólumokból áll, ezért az első sor 6. oszlopában található betűt, azaz az F-et jelöli. Ezzel a C és a D válaszlehetőségeket kiejtettük.

A második, harmadik és negyedik betű ugyanaz az A és a B válaszlehetőségnél, ezért érdemes például az utolsót összehasonlítani.

Mivel ez a jel a második sor első oszlopának a kombinációját jelenti, azaz a J-t, így a B) a helyes megoldás.

MIÉRT INFORMATIKA?

A mai társadalomban fontos kérdés az adatbiztonság. Az adatok jogosulatlan olvasás elleni védelmének egyik módja a titkosítás.

Az információ titkosításának (kriptográfia) tudománya legalább 3500 éves. A titkosítás egyik legrégebben ismert módszere a betűk más betűkkel vagy karakterekkel való cseréje. Titkosításkor a sima szöveget egy rejtjelezett kulcs segítségével kódolják. A sima szöveg rejtjelekből történő rekonstruálását dekódolásnak nevezzük. A kriptanalízis tudománya az, ha valaki a kulcs ismerete nélkül meg tudja fejteni az eredeti szöveget.

Ennek a feladatnak a titkosítási módszere egy úgynevezett monoalfabetikus titkosítás. Ezek az eljárások pontosan egy, és mindig ugyanazt az egy új karaktert választják az egyes betűkhöz. Gyakran használnak olyan rendszereket, amelyek könnyen megjegyezhetők. Ennek a feladatnak a rendszere hasonló a szabadkőműves ábécéhez. Az ilyen szövegek megfejtésével foglalkozó kriptanalitikusok speciális technikákat, például gyakoriságvizsgálatot vagy n-grammot használnak ahhoz, hogy a titkosított szöveg karaktereit hozzárendeljék az eredeti betűkhöz. Edgar Allen Poe 1843-ban közzétett „Az aranybogár” (The Gold Bug) című novellájában megmutatta, hogy ez általában lehetséges a monoalfabetikus titkosításokban.

Mit tehetne Dara, ha nem találta volna meg a táblát, de rendelkezésére állt volna a négy lehetséges jelentés?

Kizárhatta volna az első (A) és az utolsó előtti (C) jelentéseket, mert azoknál a két utolsó betű egyforma, míg a kódolt üzenetben nem. Ugyancsak kizárhatta volna az utolsó (D) megoldást, mivel abban szerepel 2 azonos betű (az e), ami a kódolt üzenetben nem.

KULCSSZAVAK

Kriptográfia, adatvédelem,

WEBOLDALAK

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

https://hu.wikipedia.org/wiki/A_kriptogr%C3%A1fia_t%C3%B6rt%C3%A9nete

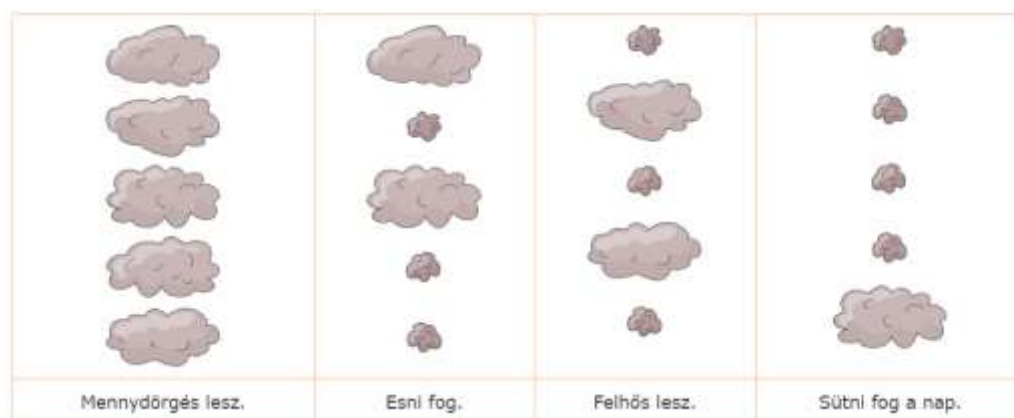


FÜSTJELEK (2019-CH-11C)

KISHÓD – KÖZEPES

BENJAMIN – KÖNNYŰ

A hegy tetejéről az időjárásért felelős hód füstjeleket küld a völgyben lévő hódoknak. Nagy és kis füstfelhőket tud létrehozni. A hódok a következő füstjelek jelentésében állapodtak meg.



Egy nagyon szeles napon a völgyben lévő hódok nem látják jól a füstfelhőket. Ami biztos, hogy a második, és a negyedik füstfelhő nagy, a többit kérdőjellel helyettesítettük.

**Mit jelenthet a füstjel?**

- A) „Mennydörgés lesz” vagy „Esni fog”.
- B) „Esni fog” vagy „Felhős lesz”
- C) „Esni fog” vagy „Sütni fog a nap”.
- D) „Mennydörgés lesz” vagy „Felhős lesz”.



„D” a helyes válasz

A völgyben lévő hódok felismertek két nagy füstfelhőt: a másodikat és a negyediket. A „Mennydörgés lesz” és a „Felhős lesz” füstjelzéseknél a második, és a negyedik helyen van két nagy füstfelhő. Az „Esni fog” és a „Sütni fog a nap” ugyanezen helyein kis füstfelhő van. Így ez a két füstjel nem egyezik meg a völgyben megfigyelt hódokéval. Így a helyes válasz a D) „Mennydörgés lesz” vagy „Felhős lesz”.

MIÉRT INFORMATIKA?

Ha valaki üzenetet küld valakinek, akkor fontos, hogy az helyesen érkezzon meg a vevőhöz. A feladatban lévő üzenetet nagy és kis füstfelhők segítségével továbbítják. Általános esetekben szimbólumokról beszélünk. Ezért van értelme szimbólumok sorozatát választani, hogy a továbbítandó üzenet még akkor is megérthető legyen, ha az közben megsérült. Ezt úgy lehet elérni, ha a szükségesnél több információt közlünk. Ezt a többlet információt nevezzük redundanciának.

Ha a sérült üzenetet legfeljebb n hibával lehet rekonstruálni, akkor az n önjavító kódolásról beszélünk. Az informatikusok tipikus feladata, hogy a szimbólumok sorozataként küldött információkat, amik útközben sérültek rekonstruálják, illetve olyan algoritmusokat, eljárásokat találjanak ki, melyek minél több hibát tudnak nem csak észrevenni, de javítani is.

Egyébként két füstfelhő elegendő lett volna az üzenet továbbításához:

			
Mennydörgés lesz.	Esni fog.	Felhős lesz.	Sütni fog a nap.

A hódok azonban öt füstfelhőt használtak. Azokban az esetekben, amikor kettő vagy akár három füstfelhő is olvashatatlan, ez még lehetővé teszi az üzenet megfelelő megértését. A feladatban a hódok úgy állapodtak meg az üzenetekkel, hogy két üzenet mindegyike különbözzön legalább három helyen.

KULCSSZAVAK

Hibajavító eljárás

WEBOLDALAK

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

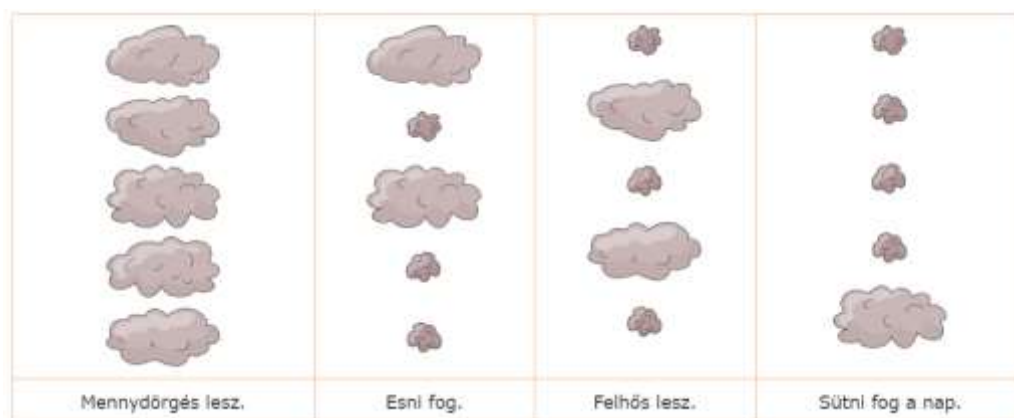


FÜSTJELEK (2019-CH-11D)

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

A hegy tetejéről az időjárásért felelős hód füstjeleket küld a völgyben lévő hódoknak. Nagy és kis füstfelhőket tud létrehozni. A hódok a következő füstjelek jelentésében állapodtak meg.



Egy nap a völgyben lévő hódok ezt látják:

Mivel ez nem tartozik a megállapodott üzenetek közé, arra gondolnak, hogy az időjárásért felelős hód hibát vétett, és az öt füstfelhő közül egyet rosszul jelzett, azaz az egyik nagy felhő helyet kicsit vagy fordítva.

Mi lehetett az eredeti üzenet, amit küldeni akart, ha pontosan egy felhőt jelzett rosszul?

- A) Mennydörgés lesz.
- B) Esni fog.
- C) Felhős lesz.
- D) Sütni fog a nap.



„C” a helyes válasz

Ha megváltoztatjuk a harmadik füstfelhőt nagyról kicsire, megkapjuk a felhős kódot. Ezért a C) a helyes válasz.

Az A) válasz helytelen, mert az első és az utolsó füstfelhőt kell megváltoztatni. a B) válasz szintén helytelen, mivel az első, a második és a negyedik füstfelhőt is meg kell változtatni. A D) válasz szintén helytelen mivel az első füstfelhő kivételével mindet meg kellene változtatni.

Ha pontosan egy füstfelhő hibás, akkor öt különböző füstjel állítható elő a lehetséges javításokkal. Az első, a második, a negyedik vagy az ötödik füstfelhő eltérő értelmezése azonban nem vezet a négy elfogadott füstjel egyikéhez sem.

Ha azonban a harmadik füstfelhőt kis füstfelhőként értelmezzük, akkor a füstjelnek a helyes választ adja a C) "felhős".

Össze is hasonlíthatjuk az értelmezett füstjelet a négy elfogadott füstjelzéssel, és megnézhetjük, hogy hány füstfelhő különbözik egymástól. Ez a "viharos" füstjel esetében két füstfelhő (felső és alsó), a "esős" füstjel esetében három füstfelhő (a felső két és a második legalacsonyabb), a " felhős" füstjel esetében egy füstfelhő (a középső, tehát ez a megfelelő megoldás, amint azt fentebb írtuk) és a "napos" füstjel esetében négy füstfelhő (a felső egy kivételével).

MIÉRT INFORMATIKA?

Ha egy szimbólumsorozatot használunk a kommunikációhoz (emberek vagy számítógépek között), akkor jobb ha úgy választjuk meg a sorozatot, hogy az átadott információ javítható, módosítható legyen, még akkor is, ha a sorozat egyes részei hiányoznak vagy sérültek. Ez úgy érhető el, hogy a szükségesnél több információt küldünk el. Az információban ezáltal lesznek redundáns részek. Az informatikában ezt a módszert nap mint nap használják. Például, ha valaki zenét küld el. Ily módon a zene lejátszható, még akkor is, ha az átvitt adatok részben sérültek.

Ha a sérült üzenetet legfeljebb n hibával lehet rekonstruálni, akkor az n önjavító kódolásról beszélünk. Az informatikusok tipikus feladata, hogy a szimbólumok sorozataként küldött információkat, amik útközben sérültek rekonstruálják, illetve olyan algoritmusokat, eljárásokat találjanak ki, melyek minél több hibát tudnak nem csak észrevenni, de javítani is.

Egyébként két füstfelhő elegendő lett volna az üzenet továbbításához:

			
Mennydörgés lesz.	Esni fog.	Felhős lesz.	Sütni fog a nap.

A hódok azonban öt füstfelhőt használtak. Azokban az esetekben, amikor kettő vagy akár három füstfelhő is olvashatatlan, ez még lehetővé teszi az üzenet megfelelő megértését. A feladatban a hódok úgy állapodtak meg az üzenetekkel, hogy két üzenet mindegyike különbözzön legalább három helyen.

KULCSSZAVAK

Hibajavító eljárás

WEBOLDALAK

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



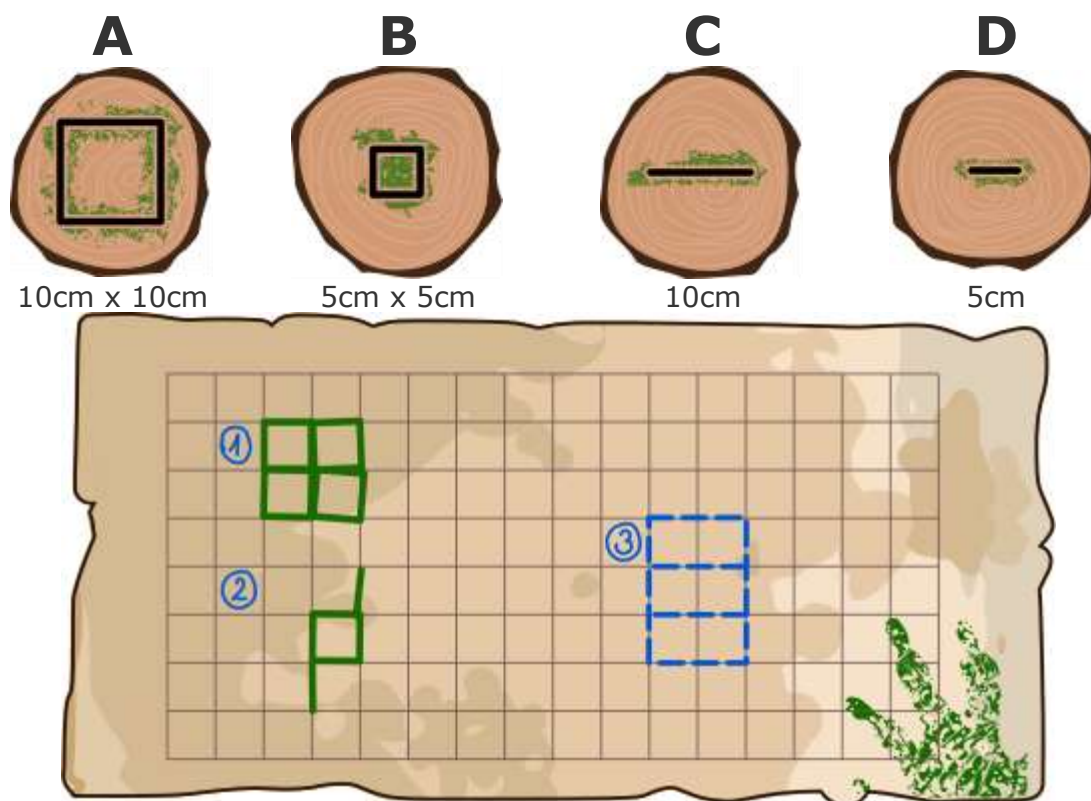
NYOMDA (2019-CH-13D)

KISHÓD – KÖNNYŰ

Hód Péternek négy nyomdája van: A, B, C és D. Ezekkel nyomdázta az ①-es és a ②-es ábrákat.

- Az ①-es ábrához csak a B nyomdát használta (négyszer).
- A ②-es ábrához a B nyomdát (egyszer) és a D nyomdát (kétszer) használta.

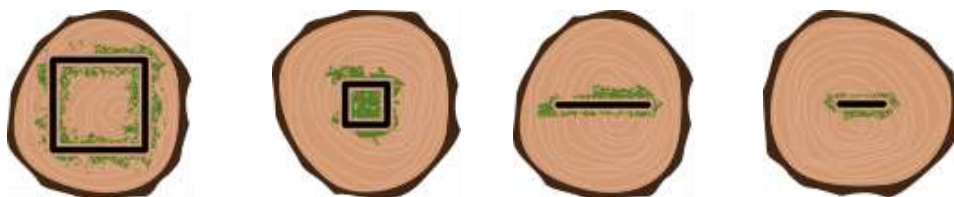
Most a ③-as ábrát szeretné nyomdázni és barátjánja, Mari felajánlja a segítségét.



Mari azt állítja, hogy a ③-as ábrához csak egy nyomdát kell kétszer használniuk.

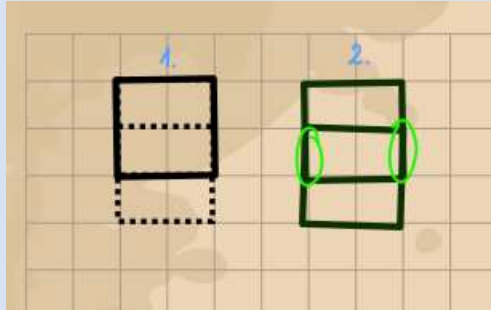
Melyik nyomdára gondolt?

- A) Nagy négyzet B) Kis négyzet C) Hosszú vonal D) Rövid vonal



„A” válasz a helyes

Mari a nagy négyzetet (A,) fogja használni. Az első lépésben a felső részét nyomdázza, majd az alsót. A nyomatok két helyen fedni fogják egymást. Ezt mutatja a zöld jelölés a képen.



MIÉRT INFORMATIKA?

A megoldást ebben a feladatban más nyomdával is el tudjuk érni: például a rövid vonalat ábrázolóval. Azzal többször kell egymás mellé/alá,... nyomdoznunk.

Sok feladathoz létezik több helyes megoldás. Van, amelyiket ezek közül könnyebb megtalálni, mint a többit. De gyakran ezek a megoldások nem ugyanannyira értelmesek. Egyes megoldások több lépést (számolási ráfordítást) igényelnek, másoknál több ideig tart megtalálni a megoldást (hardver követelmény) vagy több tároló kapacitásra van szükségük. Ennél a feladatnál is több különböző megoldást találhatunk: különböző nyomdákat használhatunk, különböző gyakorisággal. Megadott feltételekkel a legjobb megoldást megtalálni fontos feladat az informatikában.

KULCSSZAVAK

Programoptimalizálás, számítástudomány

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Programoptimaliz%C3%A1l%C3%A1s>

<https://hu.wikipedia.org/wiki/Sz%C3%A1m%C3%ADt%C3%A1studom%C3%A1ny>



STRAND (2019-CH-18)

KISHÓD – KÖNNYŰ

BENJAMIN – KÖNNYŰ

Nyár van, és a tizenkét éves Anita szeretne elmenni a strandra. A hatéves öccsét, Jancsit is magával viszi.

A strand bejáratánál az alábbi szabály olvasható:

- Alsó korhatár 8 év;
- 8 év alatti gyerekek csak 10 évnél idősebb kísérővel léphetnek a strand területére.

Melyikük mehet be a strandra?

- A) Anita és Jancsi.
- B) Anita igen, de Jancsi nem.
- C) Anita nem, de Jancsi igen.
- D) Sem Anita, sem Jancsi



„A” válasz a helyes

A szabálynak két jelentése van:

- 1) Mindenki, aki 8 évnél idősebb, bemehet a strandra. Mivel Anita 8 évnél idősebb, bemehet.
- 2) Azok, akik 8 évnél fiatalabbak, csak akkor mehetnek be, ha 10 évnél idősebb személy kíséri őket. Mivel Jancsi 8 évnél fiatalabb, de Anita – aki 10 évnél idősebb – kíséri, ezért ő is bemehet.

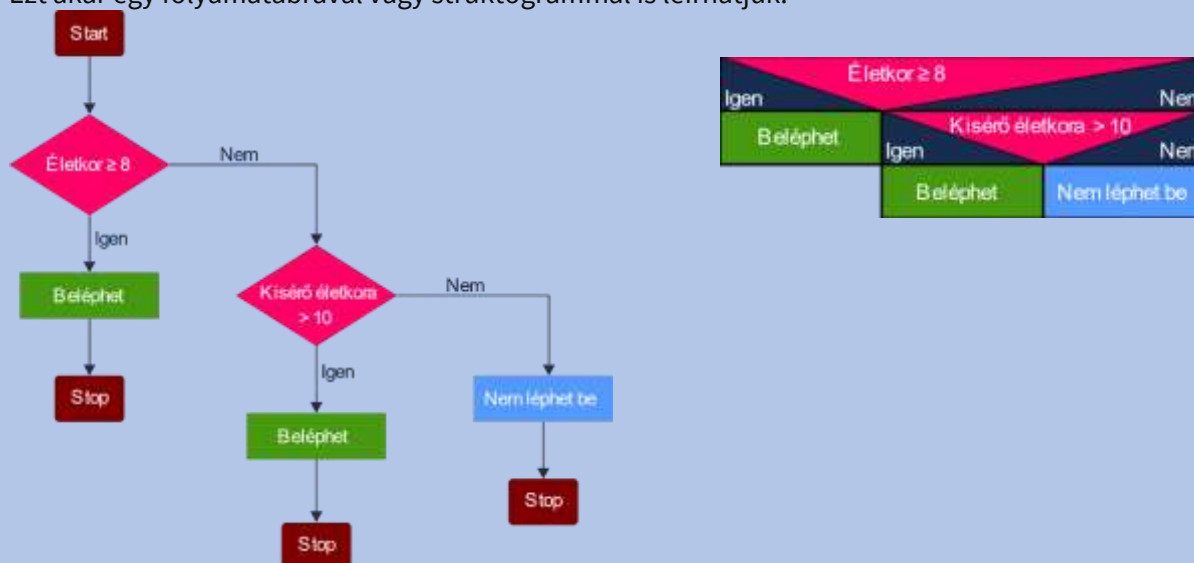
Tehát az A) a helyes válasz.

MIÉRT INFORMATIKA?

A strandon található szabály feltételeket tartalmaz, melyek közül van, amelyik megenged és van, amelyik tilt valamit. Ebben az esetben két feltételt állítottak: hogy egy személy bemehet-e vagy sem. Ha a feltétel teljesül, akkor bemehet a strandra. A szabályt így is megfogalmazhattuk volna:

*Ha a személy 8 évesnél idősebb:
bemehet
különben, ha a személyt egy 10 évnél idősebb kíséri
bemehet
különben, nem mehet be.*

Ezt akár egy folyamatábrával vagy struktogrammal is leírhatjuk:



Az ilyen eldöntési eljárásokat nevezik az informatikában elágazásoknak, melyek egy programozási feladatnál gyakran előfordulnak.

KULCSSZAVAK

Feltételek, elágazások, struktogram

WEBOLDALAK

https://hu.wikipedia.org/wiki/Felt%C3%A9teles_utas%C3%ADt%C3%A1s
<https://hu.wikipedia.org/wiki/Vez%C3%A9rl%C3%A9sfolyamgr%C3%A1f>



SZÍNES KÍNAI KARAKTEREK (2019-CN-03A)

BENJAMIN – NEHÉZ

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

A kínai karakterek szerkezete idegennek tűnhet számunkra. Hogy könnyebben megértsük a felépítésüket, az alábbi elgondolás segíthet: öt részt különböztetünk meg: a **felső** ▲, az **alsó** ■, a **bal** ■, a **jobb** ■ és a **középső** ★ részt.



Ezekből a részekből az alábbi 3 struktúrát lehet felépíteni:

Struktúra	Bal-jobb-közép struktúra	Bal-jobb struktúra	Felső-közép-alsó struktúra	Felső-alsó struktúra
Példa rajz	川	儿	三	吕
Példa színezés				

A 劳, 二 és 八 kínai karakterek ábrázolásának melyike felel meg az elgondolásban leírtaknak?

A)

B)

C)

D)



„B” a helyes válasz

Az első karakter egy felső-középső-alsó struktúrának felel meg, tehát a felső sor világoskék, a középső vonal sárga, és az alsó rózsaszín.

A második karakter megfelel a felső-alsó struktúrának, tehát a felső sor világoskék, az alsó pedig rózsaszínű.

A harmadik karakter a bal-jobb struktúrának felel meg, tehát a bal oldali vonal sötétkék, a jobboldali zöld, ezért a helyes válasz a B).

Az A) válaszban a második karakter helyesen van elemezve, de a másik két írásjelhez a helytelen színeket rendeltük hozzá: az első karakter a felső színe rossz, a harmadik karakter két színe pedig fordított.

A C) válaszban az összes karakter hibásan van elemezve. Az első karakternél a középső és az alsó szín tévesen van kiválasztva, a második karakternél a felső szín nem volt megfelelő, a harmadik karakternél pedig mindkét szín rossz.

A D) válaszban a harmadik karakter helyesen van értelmezve, de az elsőnél az alsó és a felső szín hibás, és a másodiknál mindkét szín helytelen.

MIÉRT INFORMATIKA?

A kínai betűk összetett rajzokból állnak. Még az egyszerűbb változatokban is, több mint 200 különböző alapelem (radikális) létezik, amelyekből a karaktereket összeállítják. Ezek egymás mellé vagy egymás alá vannak írva, így valóban egyfajta szerkezeteket alkotnak, amint azt a feladat is írta. Így tehát több ezer karakter kombinálható. Hogy meg tudja az ember tanulni ezeket a karaktereket, meg kell érteni azok szerkezetét. Ehhez gyakran színeket használnak segítségül.

Az általunk használt latin írásrendszer másképp működik: egy betű egy hangjelzést jelent olyan kivételekkel, mint pl. a dupla karakterekből álló betűink (cs, dz, ny, ...).

Mégis, akkor mi köze van ennek az informatikához? Egyrészt az ilyen karaktereket meg kell tudnunk jeleníteni a számítógépen. Erre különböző megközelítések léteznek, melyek egyike a feladatban leírt sémát használja. Másrészt meg kell tudnunk keresni szavakat például szótárakban vagy lexikonokban. A manapság használt jelek (radikálisok) egy nagy része olyan szótárból származik, melyet 1710 és 1716 között Kangxi császár uralkodása alatt készítettek el. Az egyes radikálisokat a vonalaik száma szerint rendezzük.

KULCSSZAVAK

Kínai írásjelek

WEBOLDALAK

https://hu.wikipedia.org/wiki/K%C3%ADnai_%C3%ADr%C3%A1s#Az_%C3%ADr%C3%A1sjegyek_szerkezete



PARKOLÁS (2019-DE-02)

KISHÓD – NEHÉZ

BENJAMIN – KÖZEPES

A tübingeni „eltolósos” parkolóban az autók a normál helyek előtt keresztben is leparkolhatnak. Egy keresztben parkoló autó, amelyik egy másik autó útját elzárja, óvatosan előre vagy hátra eltologatható. Így az elzárt autóval is ki lehet állni.

A képen egy példa látható: Az A jelű autó nincs elzárva, így ki tud állni. Az L jelű autó elzárta, ha az M jelű autót arrébb toljuk, az L jelű ki tud állni.

A példánkban egy autó kétszeresen zárolt: két másik autót kell eltolni ahhoz, hogy ki tudjon állni.



Melyik ez az autó?

- A) G jelű autó.
- B) H jelű autó.
- C) I jelű autó.
- D) K jelű auto.



„C” válasz a helyes

Az I jelű autó kétszeresen elzárt.

Az N jelű autót kellene eltolni, hogy ki tudjon állni, de ehhez nincs elég hely. Csak ha előbb az O jelű autót balra vagy az M jelűt jobbra toljuk, tudjuk az N jelűt eltolni úgy, hogy az I jelű kiállhasson.

A többi autó egyike sincs kétszeresen elzárva. Az A, D, E, J és Q jelűek szabadon kiállhatnak. A B, C, F, G, H, K és L jelűeknél elég egy autót eltolni: A B és C jelűeknél a P jelűt, az F és G jelűeknél az O, a H jelűnél az N, míg a K és az L jelűnél az M jelűt.

MIÉRT INFORMATIKA?

Az autóknek helyre van szükségük és a hely költséges. Jó ötletnek hangzik, hogy a parkolóhelyeket amennyire csak lehet kihasználják – ahogy ez Tübingenben az „eltolósos” parkolóban (Schebeparkplatz) történik.



Az informatikában egy ideje dolgoznak azon, hogy az autók autonóm módon, azaz vezető nélkül is tudjanak közlekedni, parkolni. Illetve vannak olyan törekvések is, melyek a parkolók helykihasználását igyekeznek úgy megoldani, hogy ne az autósoknak kelljen beállni az egyes helyekre – így megspórolhatóak a gyalogos közlekedés, a rámpa, lift, ... által használt területek.

KULCSSZAVAK

Parkolás, automatizáció, önvezető autók

WEBOLDALAK

<http://epiteszforum.hu/robot-technologiaval-mukodo-parkolohaz-pesten-europaban-az-elso>
http://www.hogyparkoljak.hu/index.php?option=com_content&view=article&id=214:aprklas-tortenete&catid=54&Itemid=428&showall=&limitstart=2&lang=en



MONITOROK (2019-DE-03)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

A pályaudvaron egy biztonsági kamera van felszerelve, amely 10 másodpercenként készít egy fényképet. Egy számítógép elemzi a fényképeket, és létrehoz egy úgynevezett „különbségképet”. Ezen a „különbségképen” minden olyan képpont ki van emelve, amely különbözik a két fényképen.

Példa: A két baloldali fénykép a pályaudvart, a jobb szélén lévő kép az első két fénykép különbségét mutatja.



Az alábbi fénykép és az öt „különbségkép” között öt esemény következett be.

Milyen sorrendben következhetek be ezek az események?



A)	B)
Két ember találkozik. A ház ajtaja kinyílik. Két ember egymás mellett jobbra megy. Fúj a szél. A ház ajtaja becsukódik.	A ház ajtaja becsukódik. Két ember találkozik. Két ember egymás mellett jobbra megy. A ház ajtaja kinyílik. Fúj a szél.
C)	D)
A ház ajtaja kinyílik. Két ember egymás mellett jobbra megy. Két ember találkozik. Fúj a szél. A ház ajtaja becsukódik.	Fúj a szél. A ház ajtaja kinyílik. Két ember találkozik. Két ember egymás mellett jobbra megy.



„B” a helyes válasz

Az első különbségkép az ajtó környékén mutat változást. Lehet, hogy kinyitották, vagy becsukták az ajtót. Elméletileg az is lehetséges, hogy a ház ajtajában megjelenik egy vagy több ember („Két ember találkozik”).

A második különbségkép a fától balra, a kép közepén mutat két változást, ami a két ember találkozására utal, de még nem egymás mellett mennek.

Mivel a ház ajtajánál már nincs jelzett változás, ezért az előző esetben az ajtó nyitása vagy zárása lehetett csak az esemény. Ha két ember állt volna az ajtóban, akkor a „változatlan” azt jelentené, hogy továbbra is ott állnak.

A harmadik különbségkép három változást mutat: a fától balra közepén (ahol az előző különbségkép is mutatta) és a fa alatt. A fától balra, közepén található két változás arra utal, hogy az ott találkozó két ember elhagyta azt a területet (vagy együtt, vagy külön), vagy mozogtak. A fa alatti szélesebb blokk szerint pedig együtt „egymás mellett jobbra” mentek el.

A negyedik különbségkép ismét az ajtó környékén és az egymás melletti embereknel mutat változást. Az első azt jelenti, hogy a ház ajtaja kinyílt vagy becsukódott. Nem jelentheti azt, hogy az ajtóban találkozott két ember, vagy átsétáltak oda, mivel az az esemény már megtörtént.

Ahol eddig a két ember egymás mellett volt, ismét történt változás, vagy mozogtak, vagy elmentek.

Az ötödik különbségkép látható változásokat mutat a fánál. Ez csak a szél lehet.

Tehát a fotók valahogy így nézhettek ki.

MIÉRT INFORMATIKA?

Napjainkban már számos közteret és magánterületet megfigyelnek kamerák segítségével. Mivel túlságosan drága lenne minden kamerát állandóan figyelni, számítógépek automatikusan elemzik, melyiken észlelnek változást, és szükség esetén értesítik a tulajdonost vagy a biztonsági szolgálatot.

Természetesen ezek a számítógépes elemzések jobbak, mint az ebben a feladatban leírtak: általában a kis változásokat (például amikor egy madár messze repül a képtől) vagy a lassú folyamatos változásokat (például amikor lemegy a nap) figyelmen kívül hagyják, és csak a nagyobb vagy gyorsabb változásokról küldenek értesítést. Ha az aktuális képet és a különbség képet elküldik, akkor az ember gyorsan eldöntheti, hogy kell-e reagálnia.

A közterek és magánterületek ellenőrzése ellentmondásos. Egyrészt, a fent leírtak alapján, sok hely figyelhető meg egyszerre, és ha problémák merülnek fel, gyorsan és időben beavatkozhatnak.

Ha a felvételeket rögzítik is, egy büntett esetén később bizonyítékokkal rendelkezhetünk az elkövetők megtalálásához és elítéléséhez. Másrészt az ilyen megfigyelő kamerákkal könnyen vissza is lehet élni. 2013-ban Londonban nyilvános szemeteseknél helyeztek ki kamerákat, amelyek nemcsak képernyőként működtek reklámok megjelenítéséhez, de az okostelefonok WLAN moduljainak felhasználásával rögzítették a járókelők mozgásprofiljait is ... a járókelők ismerete vagy beleegyezése nélkül. A megfigyelő kamerák arcfelismerés segítségével vagy a profiljuk „felismerésével” azonosíthatják ezeket a járókelőket, vagy legalább azonosíthatják a potenciális ügyfelek konkrét csoportjait, és célzott hirdetéseket mutathatnak a kukákon.

Tehát az épületek biztonsági megfigyelése fontos és elfogadható lehet a szupermarketekben is, de társadalmi szempontból meg kell találni a megfelelő megoldást a biztonsági igények, a kereskedelmi érdekek és az embereknek magánélethez való joga között.

KULCSSZAVAK

Digitális képfeldolgozás

WEBOLDALAK

https://hu.wikipedia.org/wiki/Digit%C3%A1lis_k%C3%A9pfeldolgoz%C3%A1s

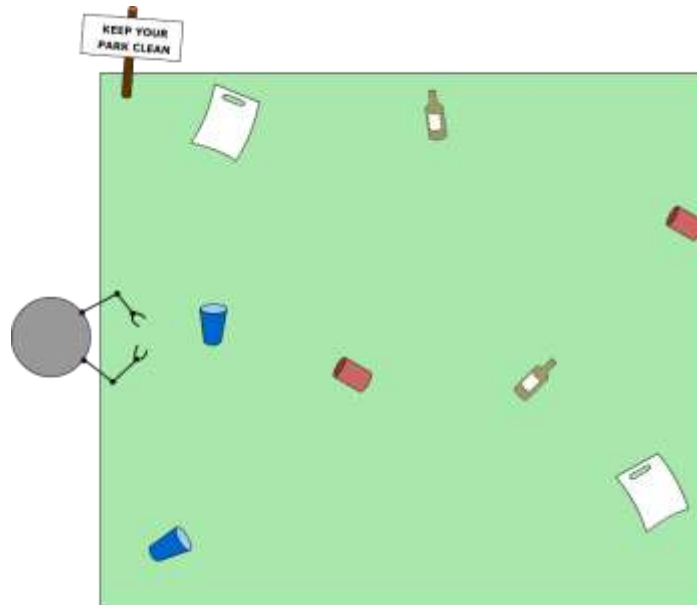


TAKARÍTÁS (2019-DE-04)

KISHÓD – KÖZEPES

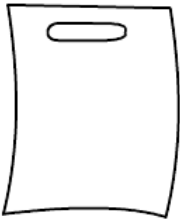
BENJAMIN – KÖNNYŰ

A parkban zajló koncert után a robot összegyűjti a szemetet, amit a közönség a gyepen hagyott.



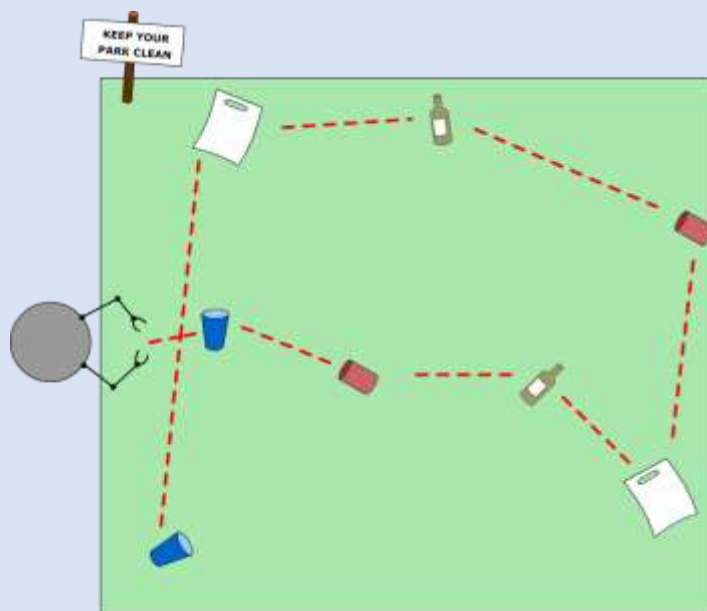
A robot mindig a gyepen található legközelebbi tárgyhoz megy, amit azután felvesz. Ezután az ahhoz legközelebbi tárgyhoz megy. A robot addig folytatja ezt a tevékenységet, amíg az összes szemetet be nem gyűjti.

Melyik tárgy lesz az, amelyiket a robot utoljára gyűjt be?

A)	B)	C)	D)
			



„B” válasz a helyes



Ez a kép mutatja a robot útját. A végén egy kék papírpoharat gyűjt be.

MIÉRT INFORMATIKA?

A robot működését egy számítógépes program vezérli. Ez a robotba beépített számítógépen fut. A program olyan utasításokból áll, amely a számítógép által értett nyelven írtak. A robot rendelkezik kamerákkal, érzékelőkkel, melyekkel érzékeli a környezetében lévő tárgyakat, és kiszámítja ezek távolságát. Majd a program feldolgozza az ezekből származó adatokat, és lehetővé teszi a robot számára az önálló mozgást, így nem kell külön távirányítással működtetnie egy embernek.

A szemétszedő robot munkája meglehetősen veszélytelen. Viszont egy robot az emberek számára veszélyes munkákat is el tud végezni. A robotok a katasztrófák után túlélőket kereshetnek, vagy mérgező, illetve radioaktív anyagokat is be tudnak gyűjteni, bombákat tudnak hatástalanítani.

KULCSSZAVAK

Program, robot, szenzor

WEBOLDALAK

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



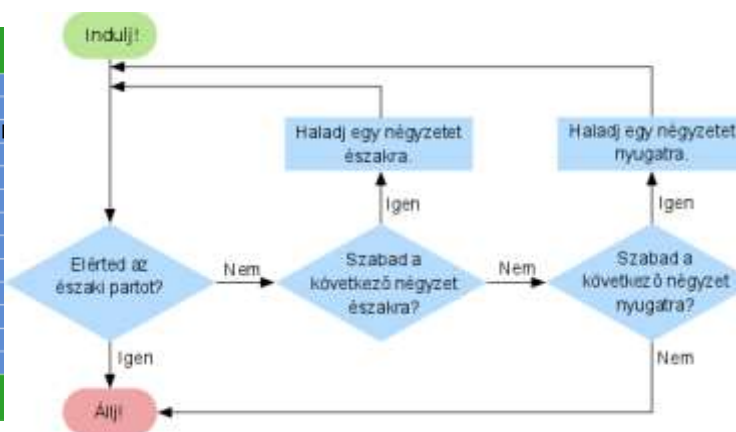
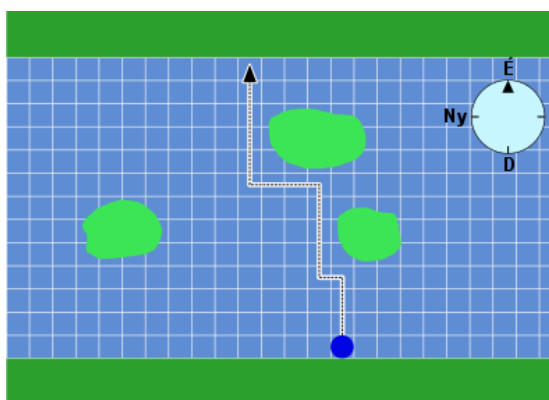
A CSAPDA (2019-DE-05)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Egy önvezető hajó a folyó déli partjától halad az északi felé. Közben ki kell kerülnie a szigeteket. A hajó a digitális térképén mindig egyik négyzetről a másik négyzetre mozog. A következő négyzetre csak akkor léphet, ha az teljesen szabad, és nincs benne szigetrész.



Az önjáró hajó a **jobb oldalon lévő kép** utasításainak megfelelően mozog. A kalózok mesterséges szigeteket hoztak létre, amelyek csapdát jelentenek a hajó számára. Hogyha a hajó túl közel kerül a szigethez, előfordulhat, hogy meg kell állnia, és nem folytathatja tovább az útját. Akkor elfogták.

A hajó számára melyik sziget jelenthet csapdát?

A)



B)



C)



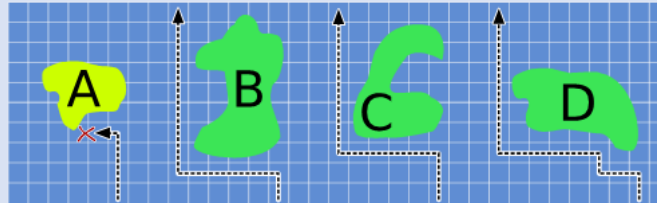
D)



„A” válasz a helyes

Az A betűjelű sziget csapda balra kanyarodásnál.

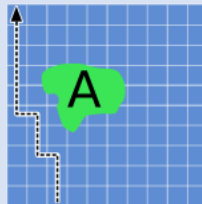
A képen a nyilak mutatják a hajó útját az egyes szigetek esetében:



Amikor az önjáró hajó jobb oldalról éri el a szigetet, ahogy a képen látható, először nyugatra megy majd megáll anélkül, hogy elérné a folyó északi partját. Ekkor csapdába esett. Az ok: nem tud sem nyugatra, sem északra továbblépni, és más mozgási irányokhoz pedig nincs utasítása.

Ha más a kiindulási pontja – mint a jobb oldali képen látható – akár ki is kerülheti az A szigetet, azaz ez a sziget sem fogja el.

A többi sziget nem jelent veszélyt a hajóra. Ezeknek a szigeteknek a déli partján nincs olyan szabad négyzet, amelyeknek a nyugati és az északi szomszédja is foglalt lenne. Ha egy hajó a szigetek egyikét megközelíti délről, akkor mindig nyugatra térhet ki, és meg tudja kerülni a szigetet (lásd kép).



MIÉRT INFORMATIKA?

Az önjáró hajó irányítását egy folyamatábra írja le. A folyamatábra (vagy programterv, PAP) meghatározza az algoritmust vagy a munkafolyamatot. Minden téglalapja utasításokat tartalmaz, minden rombusz pedig egy eldöntendő kérdést, amire igen vagy nem a válasz. A nyilak jelzik az utasítások végrehajtásának sorrendjét. A lekerekített téglalap jelöli a folyamat kezdetét és a végét.

A folyamatábrán jelölt utasítások egészen jól működtek, a folyó természetes kör alakú szigeteivel. Azonban nem vették figyelembe azt, hogy a szigetek eltérő alakúak is lehetnek, és bizonyos feltételek mellett megállíthatják a hajót egy szigetenél az északi part helyett. Ez a folyamatábra gyengesége, amit a kalózok kihasználnak, mint biztonsági rést.

Az informatikai szakemberek olyan szoftvereket próbálnak kifejleszteni, amelyek minden elképzelhető körülmény között megfelelően működnek. Például az interaktív weboldalak, amelyek bemeneti mezőket tartalmaznak, alaposan átvizsgálják a bemeneti szövegeket és elutasítják azokat a szövegeket, amelyek veszélyesek lehetnek a számítógépre nézve (pl. káros szkriptek).

KULCSSZAVAK

Folyamatábrák, navigáció, tájékozódás

WEBOLDALAK

<https://wiki.prog.hu/wiki/Algoritmus#Folyamat.C3.A1br.C3.A1k>

CELEB-HELYZET (2019-DE-08)

BENJAMIN – NEHÉZ

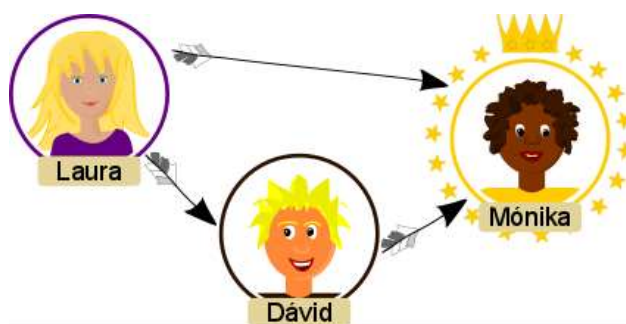
KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

A TiniGram szociális hálón a tagok más tagokat követhetnek. Ezen kívül a TiniGramon a tagok csoportokat alkotnak. Egy csoporton belül egy tag akkor celeb, ha:

- minden más csoporttag követi, és
- ő maga nem követ senkit a csoporton belül.

A következő csoportban Laura követi Mónikát és Dávidot, Dávid követi Mónikát, de Mónika nem követ senkit. Ezért Mónika egy celeb.



Egy másik csoport hat tagból áll: Andrea, Dani, Fanni, Gergő, Robi és Sanyi. A következőképpen követik egymást:

- Andrea követi Danit, Fannit és Gergőt.
- Dani követi Fannit, Gergőt és Robit.
- Fanni követi Gergőt.
- Robi követi Danit, Fannit és Gergőt.
- Sanyi követi Andreát, Danit, Fannit, Gergőt és Robit.

Van celeb a csoportban?

- A) Nincs
- B) Igen, Sanyi a celeb.
- C) Igen, Gergő a celeb.
- D) Igen, Fanni és Gergő mindketten celebek.



„C” válasz a helyes

Gergő a celeb ebben a csoportban.

Mindkét feltétel teljesül:

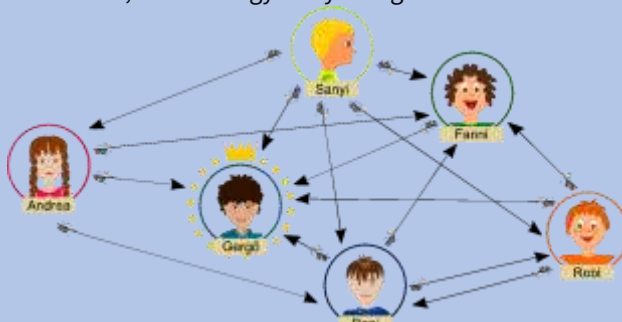
- 1) A csoport minden tagja követi Gergőt.
- 2) Gergő senkit sem követ a csoportból.

A többi válaszlehetőség hamis. Fanni nem lehet celeb, mivel ő követi Gergőt. Sanyi sem lehet celeb, mivel követi Andreát, Danit, Fannit, Gergőt és Robit.

MIÉRT INFORMATIKA?

A szociális hálók, mint a kitalált TiniGram azért működnek, mert a tagok egymás között kapcsolatokat (relációkat) alakítanak ki. Gyakran olyan egy szociális háló, hogy kapcsolatok csak egy irányba mutatnak (Andrea követi Danit). De természetesen az is gyakran megtörténik, hogy két tag kölcsönösen követi egymást (Dani követi Robit és Robi követi Danit).

Ezeket a kapcsolatokat, mint a feladat példájában is, egy gráf segítségével rajzolhatjuk fel. Nyilakat használunk arra, hogy megmutassuk, ki kit követ. Egy gráfban a tagokat csomópontoknak, a nyilakat éleknek nevezzük. Ha az élék egy meghatározott irányba mutatnak, akkor az egy irányított gráf. A feladathoz tartozó gráf így néz ki:



A sok taggal rendelkező szociális hálók gyakran nagyon nagy gráfot alkotnak. A cégek, melyek szociális hálókat üzemeltetnek, érdeklődnek az iránt, hogy különlegességeket találjanak ezekben a gráfokban.

Egy celeb nem feltétlenül valaki, akit mindenki követ, de olyan, akit sokan követnek. Ha például ez a celeb a csoportjában valami meghatározott terméket reklámoz, sokkal több tagot ér el a reklám, mintha ezt egy kevésbé követett tag tette volna.

A celebek arra is törekednek, hogy kimondottan sok követőjük (follower) legyen, és néha akár megkérdőjelezhető eszközökkel is kapkodnak utána, hogy növeljék a követőik számát: minél több követő, annál magasabb profit, melyet a reklámokkal és a termékeik terítésével elérhetnek. Ők válnak influenszerekké, akik más embereket befolyásolnak.

Hogy ezt a piacot üzleti alapokra helyezték, a szociális hálók ma már sok eszközt felhasználnak, hogy a követési kapcsolatok mennyiségét növeljék. Az is elég, hogy egy böngészőben egy adott névre rákeressünk, vagy az okostelefonunk a helyzetünket egy adott helyhez közelinek érzékelje, és a szociális háló felajánlja, hogy az adott személyt vagy egy üzletet kövessünk.

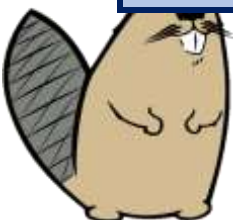
KULCSSZAVAK

Kapcsolatháló-elemzés, internetes ismeretségi hálók, gráfok

WEBOLDALAK

https://hu.wikipedia.org/wiki/Ismerets%C3%A9gi_h%C3%A1l%C3%B3zat#Internetes_ismerets%C3%A9gi_h%C3%A1l%C3%B3k

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



KÖNYVESBOLT (2019-HU-01)

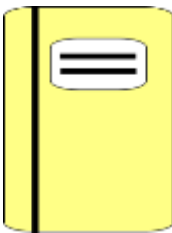
KISHÓD – NEHÉZ

BENJAMIN – KÖNNYŰ

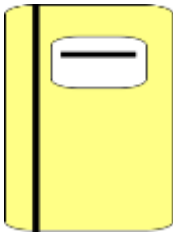
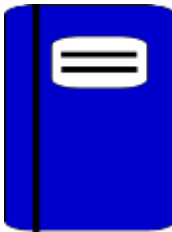
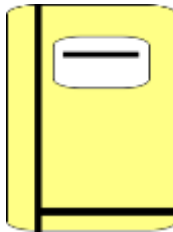
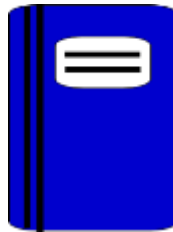
Hugo és Barbara Hódvár első könyvesboltját készülnek megnyitni, és hozzálátnak a kirakatrendezéshez.

A könyveket a borítók alapján akarják rendezni, a következő szabály szerint: Minden könyv **csak egy dologban különbözhet** a mellette álló könyvtől (a különbség vonatkozhat például a színére: sárga vagy kék; a borítón található sávokra vagy a címkén látható sávokra).

De mindenekelőtt a legnépszerűbb könyvet szeretnék a középpontba helyezni. A kirakatban az alábbiak szerint még négy könyvnek jut hely.

1.	2.	3.	4.	5.
				

Hugo a következő könyveket szeretné a kirakatba tenni:

			
1.	2.	3.	4.

Melyik könyv állhat a 2. számmal jelzett helyen?

- A) Az 1. VAGY a 2. könyv.
- B) A 2. VAGY a 4. könyv.
- C) A 3. VAGY az 1. könyv.
- D) A 4. VAGY az 1. könyv.



„A” a helyes válasz

Az első és a második könyv egyaránt állhat a 2. helyen.

A helyes sorrend:

4., 2. (középső könyv) 1., 3. **VAGY**

3., 1. (középső könyv) 2., 4.

Mivel a középső könyvet nem mozdíthatjuk el, ennek kell a kiindulópontnak lennie. Az 1. könyv színe nem változik a középső színéhez viszonyítva, ezért közvetlenül előtte vagy utána helyezhetjük el.

Ha ezt (1.) a könyvet a második helyre tesszük, akkor a válaszunk egy része már adott. (Vagyis kiszűrhetjük azokat a válaszokat, melyekben nem szerepel az 1.) Emellett, a 3. könyv állhat az első üres helyen. A negyedik helyen a 2. könyv állhat, és így a 4. könyv kerül majd az ötödik helyre.

Ha azt választjuk, hogy az 1. könyvet a középső után helyezzük el, a negyedik helyre, az előbbi sorrend megfordul, és a második helyre kerül a 2. könyv.

MIÉRT INFORMATIKA?

Az elemekhez (objektumokhoz) tartozó tulajdonságok meghatározása és kezelése nem csupán programozási probléma az informatikában. Hasonló tulajdonságok alapján az elemek kigyűjthetőek, listába rendezhetőek, vagy megváltoztathatóak. Vagy a tulajdonságaik megváltoztathatóak.

A programozásban az elemek tulajdonságait úgynevezett változókból tároljuk. A változók értéke a program futása során akár meg is változtatható.

KULCSSZAVAK

Tulajdonság, változó, oop, adatbázis

WEBOLDALAK

<https://hu.wikipedia.org/wiki/V%C3%A1ltoz%C3%B3>

https://hu.wikipedia.org/wiki/Objektumorient%C3%A1lt_programoz%C3%A1s



BOMBONOS ZACSKÓ (2019-HU-02)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Petrának van egy üres tálja és egy nem átlátszó zsákja 4 piros, 4 zöld és 4 sárga cukorral.

Petra és Móric a következő játsszák: Móric három körön keresztül húz cukrokat a zsákból. Minden húzott cukorra a következő szabályok vonatkoznak.

- Amíg a húzott cukor zöld, azt a tálba teszi és újat húz helyette. A kör folytatódik.
- Ha a húzott cukor piros, beleteszi a tálba és befejezi a kört.
- Ha a húzott cukor sárga, Móric megeszi anélkül, hogy a tálba tenné és befejezi a kört.



Maximum hány cukor lesz a játék (a harmadik kör) végén a tálban?

- A) 6
- B) 7
- C) 8
- D) 9



„B” a helyes válasz

7 cukor lesz a tálban.

A legkedvezőbb esetben Móric elsőre kihúzza mind a négy zöld cukrot. Ez azt jelenti, hogy mind a négy zöld cukor a tálba kerül, és Móric még egy ötödik cukrot húzhat az első körben. Kedvező esetben ez a piros, amit szintén a tálba kerül.

A további két körben a legjobb esetben 1-1 piros cukrot húz, így azok is a tálba kerülnek. Azaz összesen 7 cukor kerül a tálba.

A kihúzott cukorkák sorrendje mindegy addig, amíg az utolsó piros lesz, mivel akkor a zöld bonbonok biztosítják az új húzásokat.

MIÉRT INFORMATIKA?

A feladatban felállított szabályok közül kettő eldöntést tartalmaz: ha egy megadott feltétel teljesül, egy meghatározott utasítást (akciót) kell végrehajtani. Ilyen eldöntések a programozás során gyakran előfordulnak. Elágazásoknak vagy feltételes utasításoknak hívjuk ezeket.

Egy szabály úgy került megfogalmazásra, hogy valamit egészen addig ismételünk, amíg egy megadott feltétel teljesül. Azaz abbahagyjuk, amikor már nem teljesül. Az ilyen programozási elemeket nevezzük ciklusnak. Amennyiben pontosan tudjuk, hányszor kell megismételnünk valamit, akkor számlálós ciklust használunk. A feladatban szereplő játékot tehát megfogalmazhatjuk így is:

```
körök_száma legyen 3
Amíg legalább egy a körök_száma
csökkentsd a körök_száma-t eggyel
  húzz egy cukrot
  amíg a cukor zöld
    tedd a tálba
  húzz újabb cukrot
ha a cukor piros, akkor tedd a tálba
ha a cukor sárga, edd meg
```

A feladat megoldásához a programunkat elemeznünk kell. Egy ilyen egyszerű esetben, mint ez, természetesen a cukrok összes lehetséges húzási sorrendjét kipróbálhatjuk. Ezt akár egy számítógéppel automatizálhatjuk is.

A megoldásban felvezetett magyarázat ezzel szemben azon alapszik, hogy megértsük az összefüggéseket, és anélkül tudjuk megadni, hogy egy megoldás megfelelő-e, hogy lefuttatnánk a programot. Az ilyen elemzéseket egy számítógép nem mindig tudja levezetni, ahogy ezt a kiszámíthatóság-elmélet tudományága meg is mutatja.

Donald Knuth, a XX. század egyik legnagyobb informatikusa ezt alappontként meg is fogalmazta: „Óvatosan a fenti kóddal! Én csak a helyességét bizonyítottam, de nem próbáltam ki”.

KULCSSZAVAK

Ciklus, feltételes utasítás, iteráció, algoritmus

WEBOLDALAK

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

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

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



KÖNYVTÁRI BETÖRÉS (2019-HU-03)

KADÉT – KÖNNYŰ

Az iskolai könyvtárból egy (vagy több) tolvaj elemelte az informatikai szakkönyveket. A tettes(ek) a könyvekkel megpakolt autóval hajtottak el.

A rendőrség három jól ismert számítógépgurut azonosított gyanúsítottként: Ádámot, Bélát és Cecilt. Őket behívták a rendőrségre, ahol megtörtént a kihallgatásuk. A kihallgatás alatt az alábbi információk láttak napvilágot.

1. **Ádám, Béla** és **Cecil** kivételével senki sem lehet az elkövető.
2. **Cecil** semmit nem követ el Ádám nélkül.
3. **Béla** nem tud vezetni.

Bűnös-e Ádám?

- A) Igen.
- B) Nem.
- C) Az információk alapján nem tudjuk egyértelműen megmondani.



„A” válasz a helyes

Ádám vagy Cecil közül legalább az egyik bűnös.

- Ha Béla ártatlan, akkor Ádám vagy Cecil, vagy mindketten bűnösök. (ld. 1. pont)
- Ha Béla bűnös, akkor legalább Ádám vagy Cecil bűnös, hiszen Béla nem tudta volna vezetni az autót (ld. 3. pont). Vagyis az autót Ádám vagy Cecil vezette.
- Ha Cecil ártatlan, akkor Ádám bűnös Béla esetében levezetett okfejtés miatt.
- Ha Cecil bűnös, akkor Ádámnak is bűnösnek kell lennie, mivel a kihallgatás 2. információja alapján Cecil mindig Ádámmal együtt tevékenykedik.

Vagyis Ádám mindenképpen bűnös.

MIÉRT INFORMATIKA?

Számos informatikai probléma megoldásához a logikus levezetés segítségével jutunk el. Ha a feladatot sok feltétel befolyásolja, akkor gyakran segít az, ha a feladatot kisebb részfeladatokra bontjuk, és megpróbáljuk ezeknek a feltételeit külön teljesíteni. A fenti feladatban különböző feltevéseket próbáltunk meg teljesíteni. Ezek közül elvetettük azokat, melyek ellentmondáshoz vezettek. A megmaradt esetek mindegyikében arra a következtetésre jutottunk, hogy Ádám, bűnös.

A feladatot megoldhatjuk a matematikai logika segítségével is.

Legyen A, az alábbi állítás: „Ádám bűnös.”

Legyen B, az alábbi állítás: „Béla bűnös.”

Legyen C, az alábbi állítás: „Cecil bűnös.”

A rendőrségi kihallgatás alapján az alábbi állítások teljesülnek.

$$1. A \vee B \vee C$$

$$2. C \Rightarrow A$$

$$3. B \Rightarrow (A \vee C)$$

Ahol:

$\vee$ a logikai VAGY jele, ($X \vee Y$ igaz akkor és csak akkor, ha legalább X vagy Y igaz),

$\Rightarrow$ a logikai következtetés (implikáció) jele ($X \Rightarrow Y$ igaz akkor és csak akkor, ha X hamis vagy X és Y is igaz).

A fenti premisszákat (állításokat) felhasználva készíthetünk egy táblázatot, mely az A,B,C hármas minden lehetséges értékpárját tartalmazza. Az A állítás értéke, minden olyan esetben IGAZ, ahol a megadott feltételeink mindegyike teljesül. Vagyis levonhatjuk a helyes következtetést: Ádám bűnös.

A	B	C	$A \vee B \vee C$	$C \Rightarrow A$	$B \Rightarrow (A \vee C)$
Hamis	Hamis	Hamis	Hamis	Igaz	Igaz
Hamis	Hamis	Igaz	Igaz	Hamis	Igaz
Hamis	Igaz	Hamis	Igaz	Igaz	Hamis
Hamis	Igaz	Igaz	Igaz	Hamis	Igaz
Igaz	Hamis	Hamis	Igaz	Igaz	Igaz
Igaz	Hamis	Igaz	Igaz	Igaz	Igaz
Igaz	Igaz	Hamis	Igaz	Igaz	Igaz
Igaz	Igaz	Igaz	Igaz	Igaz	Igaz

Az ehhez hasonló táblákat igazságtáblának nevezzük. Az igazságtábla elkészítésével bonyolult feltételrendszerek teljesülésének feltételeit vizsgálhatjuk..

KULCSSZAVAK

Logika, premisszák, következtetés

WEBOLDALAK

<https://hu.wikipedia.org/wiki/%C3%8Dt%C3%A9letlogika>



ALKIMISTÁK (2019-HU-04)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

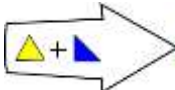
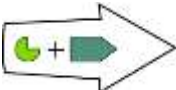
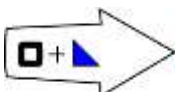
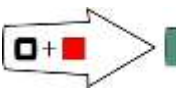
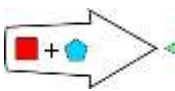
Három tudós a következő négy elemmel dolgozik:    

Egy új elemet próbálnak létrehozni: 

Minden tudós két-két napot dolgozik, de naponta egy tudós egy műveletet tud végrehajtani, és egy nap csak egy tudós dolgozik.

Egy művelettel két elem vegyíthető, így egy újabb elem jön létre. Ezt az új elemet akárhányszor használhatjuk a következő műveletek során.

Minden tudós, csak kétfajta műveletet tud elvégezni.

Kémikus Karola (K)				
Fizikus Fanni (F)				
Biológus Bendegúz (B)				

Az alábbi sorozatok írják le, hogy milyen sorrendben dolgoztak a tudósok. Melyik sorozat NEM vezet a keresett  elem létrejöttéhez?

- A) KFBFKB
- B) FFKKBB
- C) KFBKFB
- D) FBFKKB



„C” a helyes válasz

Mivel B (Biológus Bendegúz) második művelete hozza létre a kívánt elemet, ezért a sorozat utolsó betűje a B. Ahhoz, hogy az utolsó műveletet elvégezhesse, szüksége van erre a két elemre:



A fekete kör előállításához, K (Kémikus Karola)-nak szüksége lesz a (szürke nyíl) elemre, ezért a laborban F (Fizikus Fanni) után kell következnie. K-nak az első műveletet is végre kell hajtania a második művelete előtt. Ahhoz, hogy B (Biológus Bendegúz) elvégezhesse az első műveletét, szüksége van F (Fizikus Fanni) első műveletének eredményére.

A C válaszban a sorozat első F művelete csak az egyik elemet tudja létrehozni, de az őt követő B és K műveletekhez mindkettőre szükség lenne. A megfelelő B vagy K művelet egyike nem lesz elvégezhető, vagyis a sorozat nem vezethet a kívánt elem előállításához.

MIÉRT INFORMATIKA?

Az informatikában és a programozásban vannak folyamatok (processzek, függvények), melyek bemeneti értékek alapján képeznek kimeneti értéket. Fontos látni azt, hogy egy folyamat nem hajtható végre, ha hiányzik a megfelelő bemeneti érték, amit egy másik folyamat még nem hozott létre. A programozónak a részfeladatokat úgy kell megszervezni, hogy a szükséges bemeneti értékek rendelkezésre álljanak akkor, amikor a folyamat használja azokat.

Egy jó programozónak tudnia kell, hogy milyen sorrendben érdemes végrehajtani a különböző folyamatokat.

KULCSSZAVAK

Eljárás, programozás

WEBOLDALAK

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



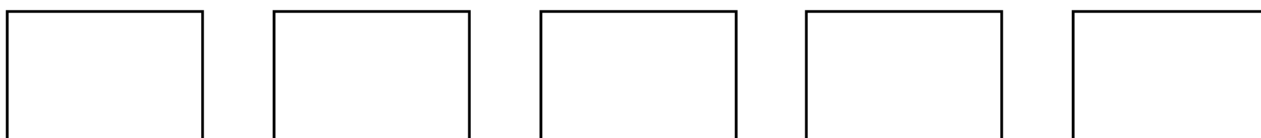
DÍSZÍTSÜNK(2019-HU-05)

KISHÓD – KÖZEPES

BENJAMIN – KÖNNYŰ

Diána a szülei házassági évfordulójára szeretné feldíszíteni az asztalokat.

Egy sorban öt asztal található. Világoskék, vagy sötétvörös terítőket, és három különböző virágcsokrot használhat a díszítéshez.



Asztalterítők:  vagy 

Virágok:  vagy  vagy 

Segíts teljesíteni az elképzeléseit, melyek:

- vagy mindegyik virágfajta legalább egy asztalt díszítsen vagy minden asztalon ugyanolyan virág legyen;
- két egyforma színű terítő ne kerüljön egymás mellé.

Melyik összeállítást válassza Diána?

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



„B” válasz a helyes

A B összeállításban az asztalterítők színe váltakozik, és minden fajta virág szerepel benne.

Az A és D összeállításokban nem teljesül az asztalterítők színére vonatkozó kikötés. Az A összeállításban egymás mellett szerepel két világoskék terítő, míg a D-ben három sötétvörös terítő van egymás mellett.

A C összeállításban a virágokra vonatkozó kikötés nem teljesül: vagy minden fajta virág szerepeljen, vagy csak az egyik fajta. De nekünk négy piros és egy kék virágunk van, a sárga virág pedig hiányzik.

MIÉRT INFORMATIKA?

Az informatikai problémákat gyakran kikötések listájával fogalmazhatjuk meg. A fenti feladatban olyan megoldást kellett találnunk, mely minden kikötésnek eleget tesz.

A bonyolultabb feladatoknál logikai műveleteket használunk a megoldás során. Az ÉS és a VAGY logikai operátorok (műveletek). Az ÉS (ang. AND) vagy konjunkció művelet értéke igaz, ha az összes kitétel teljesül. A VAGY (ang. OR) vagy diszjunkció művelet értéke igaz, ha legalább az egyik kikötés teljesül.

Ebben a HÓD feladatban ezt a két logikai műveletet kell felhasználni.

KULCSSZAVAK

Konjunkció, diszjunkció, logikai műveletek

WEBOLDALAK

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

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

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



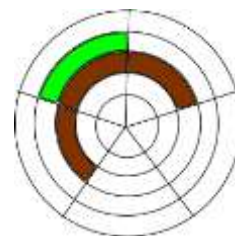
HÓDANYÓ ÉS TURING (2019-HU-06)

JUNIOR – NEHÉZ

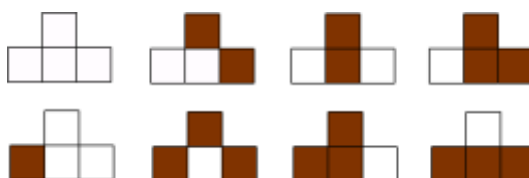
SENIOR – KÖZEPES

Hódanyó kertje körív alakú parcellákból áll. Egy parcella lehet üres, vagy beültetett (színezett).

Hódanyó körgyűrűnként belülről kifelé akarja beültetni a parcellákat. Egy parcellának három “belső” szomszédja van (lásd az ábrán).

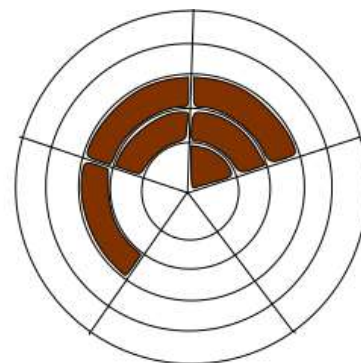
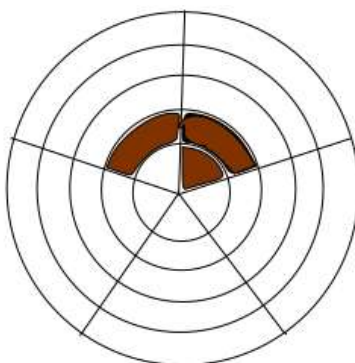
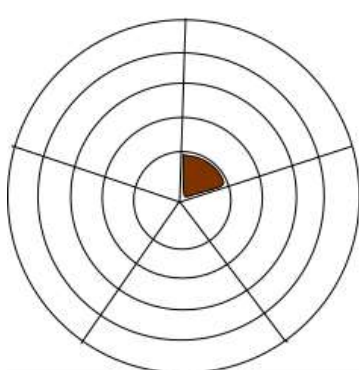


Hódanyó a belső szomszédos parcellák alapján az alábbi ültetési szabályokat alkalmazza:



Az első szabály szerint, ha a három belső “szomszéd” üres, akkor az új parcella sem ültethető be. A második szabály szerint, ha csak a jobb oldali van beültetve, akkor az új parcellába kell ültetni. A többi ültetési szabály az ábra alapján hasonlóan értelmezendő.

A legbelső körben Hódanyó csak egy parcellát ültetett be. És a szabályok szerint ültette be a következő két körgyűrű parcelláit:



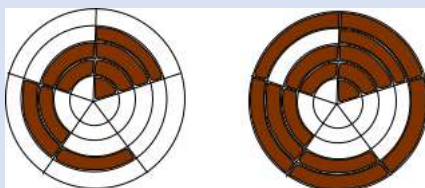
Hány parcella lesz beültetve az utolsó (ötödik) körgyűrűben?

- A. 0
- B. 2
- C. 4
- D. 5



„D” a helyes válasz

Az alábbi két ábrán látható, hogyan kerül beültetésre a két utolsó körgyűrű.



MIÉRT INFORMATIKA?

Ez a feladat valójában sokkal több, mint aminek elsőre látszik.

Bebizonyítható ugyanis, hogy kellően nagy méret esetén (sok-sok parcellával egy körön) csupán a virágok ültetésével Hódanyó bármilyen számítást elvégezhet, amit egy hagyományos értelemben vett számítógép el tud végezni. Azaz, megfelelő számú parcellát és körgyűrűt használva bármely számítás és alkalmazás modellezhető ezzel a (sejt)automatával.

Ezt szokás úgy mondani, hogy a kert és a rá vonatkozó ültetési szabályok egy univerzális Turing-gépet testesítenek meg.

A feladatban használt szabályrendszer alkalmazható a Turing gép működésének szimulálására. Fontos, hogy nem minden szabályrendszer képes erre, és csupán párat ismerünk közülük.

A feladatban használt szabályrendszer a „110-es szabály” néven ismert, és bizonyos értelemben ez a legegyszerűbb rendszer, amivel univerzális számítás végezhető. A „110-es szabály” nevének eredete abból a tényből fakad, hogy a szabályrendszer a 01101110 bináris sorozattal összegezhető, ami a decimális 110-nek felel meg.

Léteznek más szabályok is, némelyek teljesen érdektelenek, másokat bonyolult számelméleti/titkosítási problémákban használnak, netán fizikai rendszerekre emlékeztetnek, megint másokról még keveset tudunk.

KULCSSZAVAK

Turing-gép, 110-es szabály, Turing-teljesség,

WEBOLDALAK

https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_formalis_nyelvek_es_automata/ch09.html

[https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-](https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0063_13_parhuzamos_algoritmusmodellek/ch03s04.html)

[0063_13_parhuzamos_algoritmusmodellek/ch03s04.html](https://www.tankonyvtar.hu/hu/tartalom/tamop412A/2011-0063_13_parhuzamos_algoritmusmodellek/ch03s04.html)

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

https://en.wikipedia.org/wiki/Rule_110

<https://hu.wikipedia.org/wiki/Turing-g%C3%A9p>



RANGOLI (2019-IN-09)

BENJAMIN – KÖZEPES

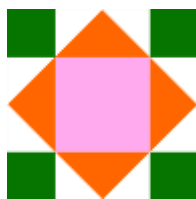
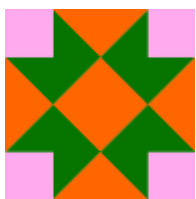
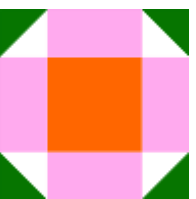
KADÉT – KÖNNYŰ

Rangoli egy művészeti forma Indiában, melyben mintákat raknak ki a padlóra. Ezek a minták általában szimmetrikusak.

Priának három különböző fajta Rangoli köve van: nyolc zöld háromszög, négy rózsaszínű négyzet és hat narancssárga háromszög. Az azonos színű kövek ugyanakkorák.

Nyolc zöld háromszög	Négy rózsaszínű négyzet	Hat narancssárga háromszög
		

Egy weboldalon a következő Rangoli mintákat találta (a fehér területek üresen maradnak):

1. minta	2. minta	3. minta
		

Mely mintákat tudja Pria kirakni a meglévő köveiből?

- A) Csak az 1. mintát.
- B) Csak az 2. mintát.
- C) Csak az 3. mintát.
- D) Mindhárom mintát.



„A” a helyes válasz

Pria csak az 1. mintát tudja a meglévő köveiből kirakni.

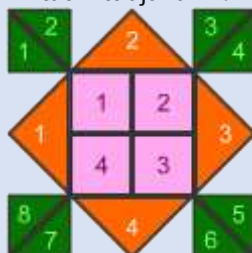
A következő ábra megszámlolja az 1. mintához szükséges köveket. Mivel minden típusú kőből legalább annyira van szüksége, amennyi rendelkezésre áll, így az 1. mintát ki tudja rakni.

A 2. mintához összesen 12 zöld háromszögre lenne szüksége, mivel a négy zöld csoport mindegyike három zöld háromszögből áll. De Priának csak nyolc zöld háromszöge van, így ezt a 2. mintát nem tudja kirakni.

A harmadik mintához összesen nyolc rózsaszínű négyzetre lenne szüksége, mivel ebben a mintában a négy rózsaszínű rész mindegyike két rózsaszínű négyzetet tartalmaz. Ezért a 3. mintát sem tudja kirakni.

Mivel sem a 2. sem a 3. mintát nem tudja kirakni, ezért a D) válasz sem helyes. Pria csak az 1. mintát tudja a meglévő köveiből kirakni. Ezért a helyes válasz az A.

A következő ábra megszámlolja az 1. mintához szükséges köveket. Mivel minden típusú kőből legalább annyira van szüksége, amennyi rendelkezésre áll, így az 1. mintát ki tudja rakni.



A 2. mintához összesen 12 zöld háromszögre lenne szüksége, mivel a négy zöld csoport mindegyike három zöld háromszögből áll. De Priának csak nyolc zöld háromszöge van, így ezt a 2. mintát nem tudja kirakni.

A harmadik mintához összesen nyolc rózsaszínű négyzetre lenne szüksége, mivel ebben a mintában a négy rózsaszínű rész mindegyike két rózsaszínű négyzetet tartalmaz. Ezért a 3. mintát sem tudja kirakni.

Mivel sem a 2. sem a 3. mintát nem tudja kirakni, ezért a D) válasz sem helyes.

MIÉRT INFORMATIKA?

Rangoli egy művészeti forma, melyhez Indiában hagyományosan színezett rizst és lisztet, vagy akár színes homokot vagy virágokat használnak. Elsősorban dekoratív célja van, de vallási vagy családi hagyományok ápolásában, a jó szerencse elnyerésében és megtartásában is szerepet kap.

Ebben a feladatban egy összetett formát kellett kisebb formákra szétbontani, hogy a rendelkezésre álló alapformákkal összehasonlíthassuk. Ezt az eljárást dekompozíciónak nevezzük és az informatikában gyakran előfordul.

A szétbontott formák alapformákkal való összehasonlításának folyamatát mintaillesztésnek vagy mintafelismerésnek is nevezzük. Ennek az informatikában nagy jelentősége van, hiszen nem csak grafikus mintákat kereshetünk, de kereshetünk szavakat szövegekben vagy adatokat nagyobb adatbázisokban, vagy akár bűncselekmények esetében viselkedési mintákat is.

KULCSSZAVAK

Dekompozíció, Rangoli, India

WEBOLDALAK

https://www.facebook.com/pg/TradionalisIndia/photos/?tab=album&album_id=216062608468432
<https://hu.wikipedia.org/wiki/Probl%C3%A9ma#Dekompoz%C3%ADci%C3%B3>

SZÉP, HOGY VANNAK FÁK (2019-IT-01)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

Sergio írt egy dalt, melyben azt írta le, hogy egy fából hogyan lesznek/lehetnek különböző dolgok. A vers így hangzik:

*Szép, hogy vannak fák.
Egy fán levelek nőnek,
Egy fán virágok nyílnak,
A virágokból gyümölcs fejlődik,
Levelekből és virágokból koszorút fonhatok magamnak.*

Sergio-nak fontos volt, hogy az első verssor után csak olyan dolgokat írjon le, amiket korábban már megemlített a versben.

Sergio melyik verse **NEM** ilyen?

- A) Szép, hogy vannak fák.
Egy fán virágok nyílnak,
Egy fán levelek nőnek,
Levelekből és virágokból koszorút fonhatok magamnak,
A virágokból gyümölcs fejlődik.
- B) Szép, hogy vannak fák.
Egy fán virágok nyílnak,
Egy fán levelek nőnek,
A virágokból gyümölcs fejlődik,
Levelekből és virágokból koszorút fonhatok magamnak.
- C) Szép, hogy vannak fák.
Egy fán levelek nőnek,
A virágokból gyümölcs fejlődik,
Egy fán virágok nyílnak,
Levelekből és virágokból koszorút fonhatok magamnak.
- D) Szép, hogy vannak fák.
Egy fán virágok nyílnak,
A virágokból gyümölcs fejlődik,
Egy fán levelek nőnek,
Levelekből és virágokból koszorút fonhatok magamnak.



„C” válasz a helyes

A „fa”, „levelek”, „virágok”, „koszorú” és „gyümölcs” dolgok egymáshoz való viszonyát (függőségét) egy gráf segítségével lerajzolhatjuk. A nyilak azt jelentik, hogy az egyikhez szükség van a másakra.



Ez alapján a gyümölcs előtt meg kell említeni a virágokat és a koszorú előtt a leveleket és a virágokat is. Az egyes válaszoknál a megemlített dolgok sorrendje a következő:

- A) fa, virágok, levelek, koszorú, gyümölcsök
- B) fa, virágok, levelek, gyümölcsök, koszorú
- C) fa, levelek, gyümölcsök, virágok, koszorú
- D) fa, virágok, gyümölcsök, levelek, koszorú

A C válasznál előbb szerepel a „gyümölcsök”, mint a „virágok”, de ez ellentmond Sergio céljának, hiszen a gyümölcshöz virág kell. Az összes többi versben betartotta ezeket a szabályokat.)

MIÉRT INFORMATIKA?

1974-ben az olasz zeneszerző, Sergio Endrigo (1933-2005) megzenésítette Gianni Rodari (1920-1980) gyermekversét, a „Ci vuole un fiore”-t. Ebben a dalban éneklük meg, hogy ha asztalra van szükségünk, ahhoz deszka kell, a deszkához fa, a fához mag, a maghoz gyümölcs és a gyümölcshöz egy virág. Itt vezeti le egy ágon, egy fán, egy erdőn, egy hegyen és a földön keresztül azt is, hogy a virághoz ugyancsak virágra van szükségünk. Azzal zárja a dalt, hogy végül mindenhez virágra van szükségünk.

Hasonló levezetések találhatók a magyar láncmesékben (ld. pl. Benedek Elek: A kakas és pipe) vagy láncversekben, mondókákban.

Egy dolog (tárgy) másiktól való függőségét egy irányított gráf segítségével írhatjuk le. Egy ilyen gráfot készítettünk a válaszhoz is. Ez egy irányított körmentes gráf, mely több megengedő sorrendet ír le. Ha egy csomópontot (dolgát) szeretnénk, minden csomópontra szükségünk van, amiből arra mutat nyíl. Ezekre a csomópontokra ugyanez vonatkozik, így rekurzív módon egészen addig a csomópontig el kell jutnunk, amelyekre már nem mutat nyíl. Ezt használhatjuk kiinduló csomópontnak.

Sergio Endrigo eredeti dalára nem tudunk egy körmentes gráfot készíteni. Mivel a korábban is leírt második versszakban egy virághoz végül egy virágra van szükség. Ez ellentmondás arra, hogy a gráf körmentes legyen, azaz irányított körutat ne tartalmazzon.

KULCSSZAVAK

Gráf, irányított körmentes gráf

WEBOLDALAK

<https://lyricstranslate.com/hu/ci-vuole-un-fiore-it-takes-flower.html>

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

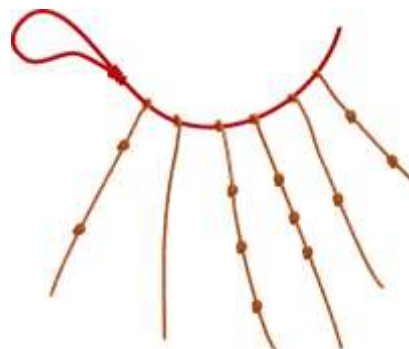


KIPU (2019-JP-03)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

Az inkák korábban az üzenetek továbbítására csomóírást használtak. Egy fő madzagon további madzagok lógtak, amelyeken lévő csomókat használták az üzenetekhez. Az ilyen, úgynevezett kipuk nagyok voltak és körülményes volt elkészíteni őket.



Képzeld el, hogy 30 féle üzenet egyikét kell eljuttatni valahova. Ehhez a kipu egyszerűsített változatát fejleszthetjük ki a következő megkötésekkel:

- A fő madzagon a lehető legkevesebb madzagnak kell lógnia.
- A madzagok csak a rajtuk lévő csomók számában különbözhetnek.
- Egy madzagra 0, 1, 2 vagy 3 csomó köthető.

Hány madzagra van szükségünk (a fő madzagon kívül) ennek a kipunak a megvalósításához?

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



„B” válasz a helyes

3 madzag kell a megvalósításhoz.

Minden fő madzagon lógó madzag 4 különböző értéket tárolhat (0, 1, 2 vagy 3). Két madzaggal már $4 \cdot 4 = 16$ lehetséges kombinációt, hárommal pedig $4 \cdot 4 \cdot 4 = 64$ -et valósíthatunk meg. Tehát három madzag elég. Ha többet használnánk, ellentmondanánk az első megkötésünknek, hogy „a fő madzagon a lehető legkevesebb madzagnak kell lógnia”.

Mivel a madzagok sorrendje a fő madzagon lévő csomóval meghatározott, nem kell azzal törődnünk, hogy valaki a másik irányból is olvashatná az üzenetet.

MIÉRT INFORMATIKA?

A kiput valóban használták az inkák Dél Amerikában: könyveléshez és illetékek, adók kivetéséhez. Feltételezzük, hogy színezett fonalak segítségével 95 különböző hangot tudtak kódolni és így levelezést folytatni. Az egyszerű példánkkal ellentétesen nem csak egyféle csomót használtak, és bizonyos esetekben további almadzagokat toldottak be, melyek a lógó madzagokról lógtak le.

A példában egy igen leegyszerűsített változat van. Mivel a fő madzag csomója meghatározta a lógó madzagok sorrendjét, az egyes értékek (0, 1, 2 vagy 3) egy számrendszert határoznak meg, ebben az esetben a 4-es számrendszert. A különböző számrendszerek elterjedtek: általában a 10-es számrendszert használjuk, de a számítógép a 2-es (binárisnak is nevezett) számrendszeren alapul. A kezdetekben volt rá kísérlet, hogy a 3-as számrendszeren (-1, 0, 1 értékeket használva) alapuló számítógépet építsenek. Egy b alapú számrendszerben n helyen pontosan b^n különböző értéket tudunk tárolni. A számítógép esetében egy byte (8 bit, mindegyik 0 vagy 1 értékkel) tehát $2^8 = 256$, míg a példában használt kipu $4^3 = 64$ különböző értéket tud tárolni.

Az inkáknak 1-től 30-ig való tárolásra egy lógó madzag is elég lett volna. Ők 10-es számrendszert használtak, mint most mi és az egyes számjegyek „leírására” különböző csomókat használtak egy madzagon.

KULCSSZAVAK

Számrendszerek, kipu

WEBOLDALAK

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

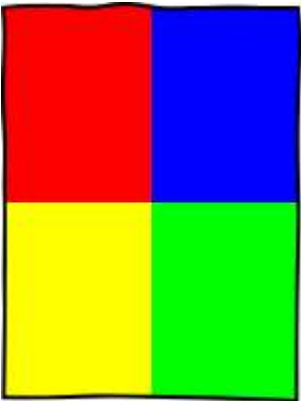
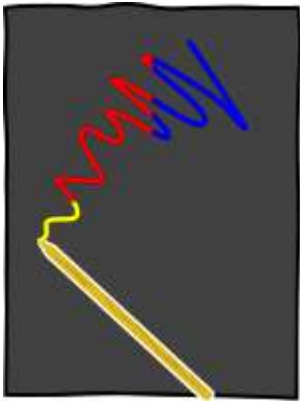
<https://hu.wikipedia.org/wiki/Sz%C3%A1msz%C3%A9r>



KARCPAPÍR (2019-KR-01)

KISHÓD – KÖNNYŰ

A karcpapírral egyszerűen készíthetsz színes képeket. Egy pálcával eltávolítod a felső réteget és a színes háttér válik láthatóvá.

		
Kezdetben a karcpapír mindenütt feketével fedett.	A felső réteg alatt ez a négyszínű papír van elrejtve.	Ahol a pálcával a fekete fedőréteget megkarcoljuk, előtűnnek az elrejtett színek.

Az alábbi négy kép esetében melyiken tűnik elő pontosan három szín?

A)



B)



C)

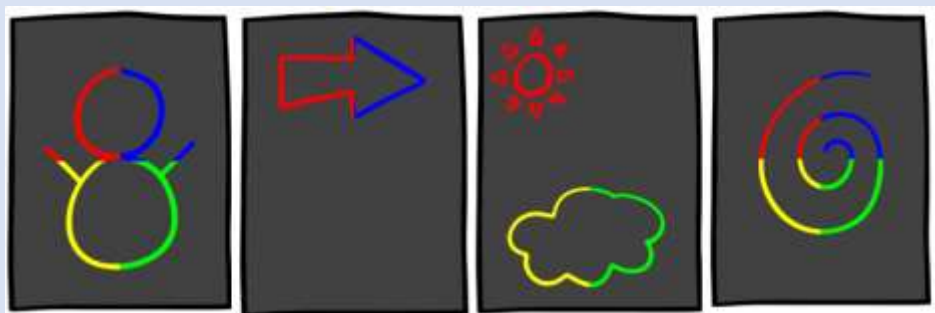


D)



„C” válasz a helyes

A négy kép esetében a következő színek bukkannak elő:



A helyes válasz tehát a C), hiszen a piros, a sárga és a zöld színek bukkannak elő. A negyedik szín, a kék nem, mivel a jobb felső negyedébe a képnek nem karcoltunk semmit.

Az A) és a D) képek esetében mind a négy szín előbukkan, míg a B) esetében csak a felső két szín, a piros és a kék.

MIÉRT INFORMATIKA?

karcpapír felső rétegére karcolva a réteget azokon a helyeken átlátszóvá tesszük, így az alatta lévő réteg láthatóvá válik.

Sok képszerkesztő program rétegekkel (angolul: layer) dolgozik, melyek egyes helyeken akár átlátszóak is lehetnek. Időnként például egy adott képre szöveget helyeznek. A szöveget tartalmazó réteg pedig a szöveget kivéve mindenhol átlátszó. Természetesen a szöveget egyből ráírhatnánk a képre, de ha rétegeket használunk, könnyebb eltávolítani vagy megváltoztatni az egyes összetevőket anélkül, hogy a többi „károsodás” érné.

Ez a feladat talán könnyebben megoldható, ha a nagyobb képet kisebb részekre (itt negyedekre) osztjuk és a negyedeknél vizsgáljuk az átlátszóságot. Ezt az eljárást dekompozíciónak, azaz részekre osztásnak nevezzük, és az informatikában gyakran használatos.

KULCSSZAVAK

Képszerkesztő, layerek, dekompozíció

WEBOLDALAK

<https://halado.fotokonyv.hu/photoshop-a-legelejetol-layers-panel-avagy-retegek-panel/>
<https://hu.wikipedia.org/wiki/Probl%C3%A9ma#Dekompoz%C3%ADci%C3%B3>



HAMBURGER HOZZÁVALÓK (2019-KR-07)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

A hódburgerben hat összetevő található (A, B, C, D, E és F). Az alábbi táblázat négy hódburger összetevőit mutatja úgy, hogy a betűjelek nem feltétlenül jelentik az összetevők sorrendjét.

Hódburger				
Összetevők	C, F	A, B, E	B, E, F	B, C, D

Melyik hódburger hozzávalói az A, E és F betűjelűek?

A)



B)



C)

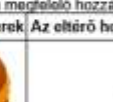


D)



„A” válasz a helyes

Ahhoz, hogy megtudjuk melyik hozzávalóhoz, melyik betűjel tartozik, mindig két hódburgert kell összehasonlítani:

Burgerek összehasonlítása	Közös betűjel	Közös hozzávaló
 	F	
 	C	
 	B	
 	B (már meghatároztuk)	
 	E	

Két hozzávaló csak egy hamburgerben található meg. Mivel már ismerjük az összes többi betűjel jelentését, azonosíthatjuk a megfelelő hozzávalókat.

Egyedi burgerek	Az eltérő hozzávaló betűjele	Az eltérő hozzávaló
	A	
	D	

Ezért a keresett hódburger hozzávalói az A (paradicsom), E (rántott sajt) és az F (saláta) hozzávaló, és ez csak az A betűjelű hódburgerre igaz.



MIÉRT INFORMATIKA?

A logikai érvelés fontos szempont az informatika tudományában is. A feladat megoldásához a következőket kell alkalmazni: Ha ugyanolyan hozzávalójú hódburgereket hasonlítunk össze, akkor meghatározhatjuk, hogy az egyes betűjelhez milyen hozzávaló tartozik.

Ebben a feladatban a közös hozzávalók halmazában a két hódburger közös összetevői lesznek. Csak azokat a hozzávalókat tartalmazza, amelyek mindkét hódburgerben megtalálhatók. Az első összehasonlításhoz ezt a kifejezést írhatjuk $\{C, F\} \cap \{B, E, F\} = \{F\}$. A metszethalmaz párja a $\{C, F\} \cup \{B, E, F\} = \{B, C, E, F\}$, amely legalább az egyik hódburger összes alkotóját tartalmazza.

A különbség-halmaz meghatározásához meg kell keresni azokat a hozzávalókat melyek csak egy hódburgerben szerepelnek. Csak az első halmaz azon összetevőit tartalmazza, amelyeket a második halmaz nem. Például az első hódburger összetevőiből a következő kifejezést lehet leírni: $\{A, B, E\} \setminus (\{C, F\} \cup \{B, E, F\} \cup \{B, C, D\}) = \{A, B, E\} \setminus \{B, C, D, E, F\} = \{A\}$.

A halmazelméletet mindenki ismeri a matematika órákról. Az informatikában többek között az adatbázisoknál fordul elő. A halmazelmélet alapja a logika, amit Boole algebrának nevezünk.

KULCSSZAVAK

Logikai következtetés, Halmazelmélet, Logika

WEBOLDALAK

[https://hu.wikipedia.org/wiki/Boole-algebra_\(informatika\)](https://hu.wikipedia.org/wiki/Boole-algebra_(informatika))

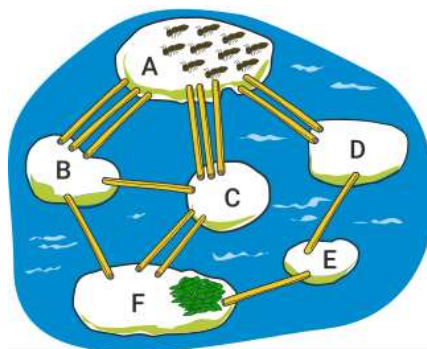
HANGYÁK AZ INGOVÁNYBAN (2019-LT-06)

BENJAMIN – NEHÉZ

KADÉT – KÖZEPES

JUNIOR – KÖNNYŰ

Tíz hangya áll egy kövön, amit a képen A-val jelöltünk. Az F-fel jelölt kövön lévő ételhez szeretnének eljutni. A kövek szalmaszálakkal vannak összekötve. Az egyes szalmaszálakon egyszerre csak egy hangya kelhet át és egy percbe telik, mire egy hangya átkel egy szalmaszálon.



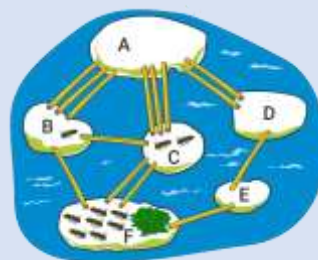
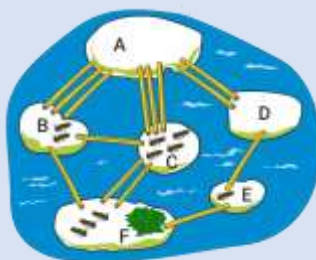
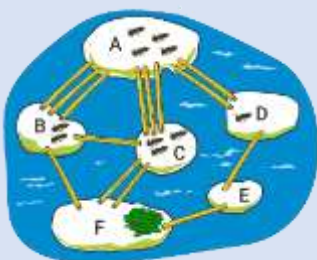
Három perc múlva maximum hány hangya érheti el az F-fel jelölt, ételt tartalmazó követ?

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



„C” válasz a helyes

Egy lehetséges átkelést mutatnak a képek percenként:



Az első másodperc után.

A második perc után.

A harmadik perc után.

Természetesen más megoldások is léteznek, de az adott időn belül a legtöbb hangya átkeléséhez be kell látnunk, hogy a D köré az első lépésben nem érdemes egynél több hangyát küldeni, mivel 3 perc alatt csak egy hangya érhet el F köré (E-n keresztül).

A második perc elején induló hangyák B-ről és C-ről maximum 3-an lehetnek és ugyanennyien tudnak indulni a harmadik perc elején is, mivel az A köről a 3-3 szalmaszál kellő utánpótlást biztosít.

Ennél több hangya viszont nem érhet el F köré, mivel egyszerre csak egy hangya lehet egy szalmaszálon.

MIÉRT INFORMATIKA?

Ennél a feladatnál az a célunk, hogy a hangyák szalmaszálakon való mozgását optimalizáljuk oly módon, hogy adott idő alatt (3 perc) a lehető legtöbben jussanak el az egyik pontból (kőről) a másikra.

Az ilyen, illetve ehhez hasonló feladatokat optimalizációs problémáknak is nevezzük. A hangyák nem ismerik előre az úthálózatot, így a legjobb megoldás megkeresését sem célozhatják meg. Egy külső megfigyelő viszont, aki az útvonalakat látja, találhat optimális stratégiát.

A gráfok olyan elméleti adatszerkezetek, melyet arra is használnak, hogy hálózatokat modellezenek. Számos algoritmus létezik meghatározott keretek közötti folyamatok optimalizálására. A megfelelő kiválasztására a megszorításokat kell alaposan és pontosan megadnunk (specifikálnunk).

KULCSSZAVAK

Optimalizáció, hálózat, gráf

WEBOLDALAK

[https://hu.wikipedia.org/wiki/%C3%81ram_\(gr%C3%A1fm%C3%A9let\)](https://hu.wikipedia.org/wiki/%C3%81ram_(gr%C3%A1fm%C3%A9let))

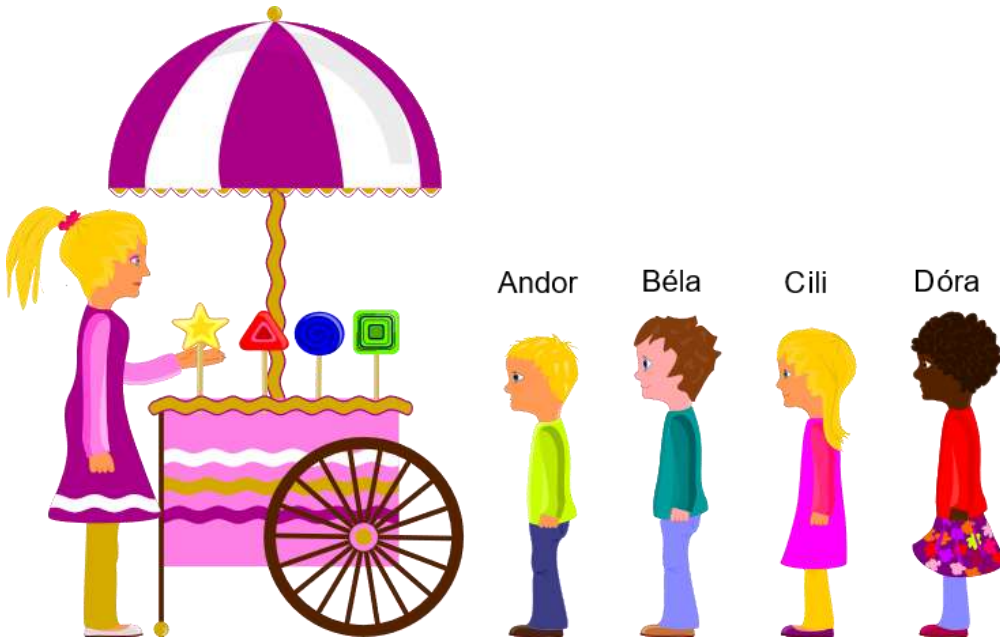
<https://hu.wikipedia.org/wiki/Gr%C3%A1f>



NYALÓKA(2019-PK-01)

KISHÓD – KÖNNYŰ

Andor, Béla, Cili és Dóra sorban állnak a nyalókaárusnál. Mindegyikük azt a nyalókat kapja meg, amelyik legközelebb van hozzá akkor, amikor sorra kerül. Andor tehát elsőként a  zöld négyzet alakú nyalókat kapja meg.



Kié lesz a  piros háromszögletű nyalóka?

- A) Andor
- B) Béla
- C) Cili
- D) Dóra



„C” válasz a helyes

Andor az első, így övé a zöld, négyzet alakú nyalóka.

Miután a zöld, négyzet alakú nyalókát eladták, a kék kör alakú lesz legközelebb a következő vevőhöz, aki Béla.

A kék kör alakú után a piros háromszögletű nyalóka kerül sorra, legközelebb a következő vevőhöz, aki Cili.

Dórának marad a sárga csillag alakú nyalóka.

MIÉRT INFORMATIKA?

Egy sorbanállás hasonló az egyik informatikában gyakran használ adatszerkezethez, melyben adatokat tárolnak. Ez a sor. Elemeket mindig „hátról” tudunk hozzáadni és „előről” kivenni. Tehát az alábbi szabály alapján működnek: „amelyiket előbb raktuk hozzá, azt vesszük el előbb”, angolul „first-in-first-out” (első be, első ki), röviden FIFO.

KULCSSZAVAK

Adatszerkezet, Sor, FIFO

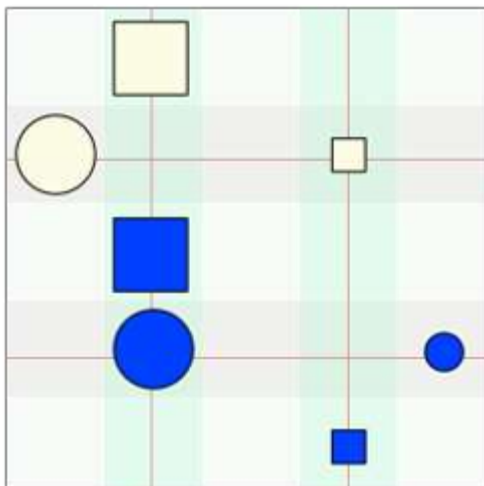
WEBOLDALAK

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



IGAZ VAGY HAMIS (2014-RU-02)**SENIOR – NEHÉZ**

Aliz és Tomi „igaz vagy hamis” játékot játszanak az osztályterem mágnes táblájánál. Aliz hét különböző mágnes teszt a táblára.



Ezután állításokat mond a mágnesek alakjáról, színéről, nagyságáról és elhelyezkedéséről.

Egy állítás igaz, a többi hamis. Tamásnak ki kell találnia, melyik állítás igaz.

Melyik állítás igaz?

- A) Van két mágnes X és Y úgy, hogy X sötétkék és Y világossárga és X Y felett van.
- B) Minden tetszőleges két X és Y mágnesre igaz, hogy ha X egy négyzet és Y egy kör, akkor X Y felett van.
- C) Minden tetszőleges két X és Y mágnesre igaz, hogy ha X kicsi és Y nagy, akkor X jobbra van Y-től.
- D) Minden tetszőleges két X és Y mágnesre igaz, hogy ha X világossárga és Y sötétkék, akkor X Y alatt van.



„C” válasz a helyes

A kis mágnesek mind jobbra vannak a nagy mágnesektől.

Az A válasz hamis, mert nincs olyan sötétkék mágnes, ami világossárga mágnes felett lenne.

B válasz hamis, mivel nem minden négyzet alakú mágnes található kör alakú mágnesek felett.

D válasz hamis, mivel nem minden világossárga mágnes található sötétkék mágnes alatt.

MIÉRT INFORMATIKA?

Ebben a HÓD feladatban arról van szó, hogy megállapítsuk, egy állítás igaz vagy hamis.

Egy mágnes tulajdonságait „négyzet alakú(X)”, „kör alakú(X)”, „nagy(X)”, „kicsi(X)”, „sötétkék(X)”, „világossárga(X)” predikátumokkal írhatjuk le. Két mágnes közötti kapcsolatot pedig a „felette(X,Y)”, „alatta(X,Y)” és „jobbra(X,Y)” predikátumokkal írhatjuk le.

A predikátum (elsőrendű) logika formális nyelvén a kijelentések így néznek ki:

A) létezik X, Y: sötétkék(X) és világossárga(Y) és felette(X,Y)

B) minden X, Y-ra: (négyzet alakú(X) és kör alakú(Y))-ből következik, hogy felette(X,Y)

C) minden X, Y-ra: (kicsi(X) és nagy(Y))-ből következik, hogy jobbra(X,Y)

D) minden X, Y-ra: (világossárga(X) és sötétkék(Y))-ből következik, hogy alatta(X,Y)

Az informatikában vannak olyan programnyelvek, melyekben az elsőrendű (predikátum) logikai állításokkal programozunk. Például a Prolog egy ilyen logika-vezérelt programnyelv.

KULCSSZAVAK

Predikátum, logikai állítások, Prolog

WEBOLDALAK

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

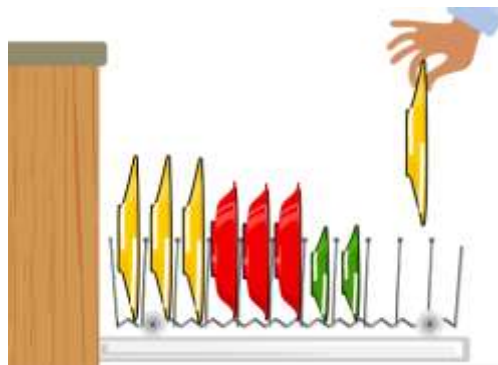


TÁNYÉROK (2019-RU-01)

BENJAMIN – NEHÉZ

KADÉT – KÖNNYŰ

Egy rendszerető hód mindig úgy rakja sorrendbe a tányérjait, mint ahogy az a képen látható. Először a nagy tányérok, utána a közepes, és végül a kis tányérok. A tányérok között nincsenek üres helyek. A vacsora után szeretne még egy további nagy tányért behelyezni a mosogatógépbe.



Legkevesebb hány tányért (beleértve az új tányért is) kell elmozdítania, hogy újra jó legyen a sorrend?

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



„A” a helyes válasz

A rendszerető hód úgy tudja a feladatot megoldani, hogy csak 3 tányért mozdít el. A képek mutatják, hogy csinálta.



A kérdés az, vajon megoldható-e kevesebb áthelyezéssel a feladat. Gondoljuk át. Szükségünk van egy mozdulatra az új tányér mosogatóba való elhelyezésére. De sajnos nincsen hely a mosogatóban a nagy tányérok között. Biztosan szükségünk van még egy mozdulatra. Ha egy nagy tányért helyeznénk át, akkor nem lennének a tányérok a továbbiakban szépen sorba rendezve. Tehát biztosan kell több mint egy mozdulat az áthelyezéshez és a helyes sorrend megtartásához. Ezzel együtt szükség van több mint 2 lépésre, ami azt jelenti, hogy minimum 3 lépésből tudjuk megoldani.

MIÉRT INFORMATIKA?

A rendszerető hód nagyon okosan oldotta meg a problémát. A megfelelő tányérokat helyezte át, így a lehető legkevesebb lépésből megoldotta a feladatot. De ez egy egyedi helyzet volt. Ha a mosogatógépben más a tányérok száma, a hódnak újra kell gondolnia, hogyan tudja áthelyezni a tányérokat, hogy a megfelelő sorrendben legyenek. Hogy ne kelljen minden esetben átgondolni a feladat megoldását, létezik egy általános eljárás rá. Ezt algoritmusnak nevezzük. Egy egyszerű algoritmus a feladat megoldásához a következő:

- 1) Mozdassa át az össze kisebb tányért a mosogatógépben egy hellyel jobbra
- 2) Állítsa be az új tányért az üres helyre.

Ebben az esetben a 6 kisebb tányért kell odébb tenni egyet, utána beállítani az új tányért. Ez összesen hét mozdulat. Sokkal rosszabb, mint a mi megoldásunk, viszont ez az algoritmus minden esetben működik. Az egyetlen kitétel ami maradt az az, hogy a mosogatógép elég nagy legyen, és az algoritmust gondolkodás nélkül teljesen automatikusan végrehajtható.

A rend megtartása és sorba rendezés nagyon fontos feladat az informatikában. Például ha az adatok sorba vannak rendezve, könnyebben megtalálhatjuk a keresett adatot. Emiatt lettek az algoritmusok a számítógépes rendszerekbe is bevezetve. A legfőbb különbség, hogy a számítógépek sokkal több objektummal dolgoznak. Amikor milliárdos nagyságrendű objektummal dolgozunk, a mi egyszerű algoritmusunk nagyon lassan oldaná meg a problémát. Az informatikusok ezért fejlesztik folyamatosan az algoritmusokat, amikkel gyorsabban megy.

KULCSSZAVAK

Algoritmus, rendezés

WEBOLDALAK

https://hu.wikipedia.org/wiki/Kateg%C3%B3ria:Rendez%C3%A9si_algoritmusok



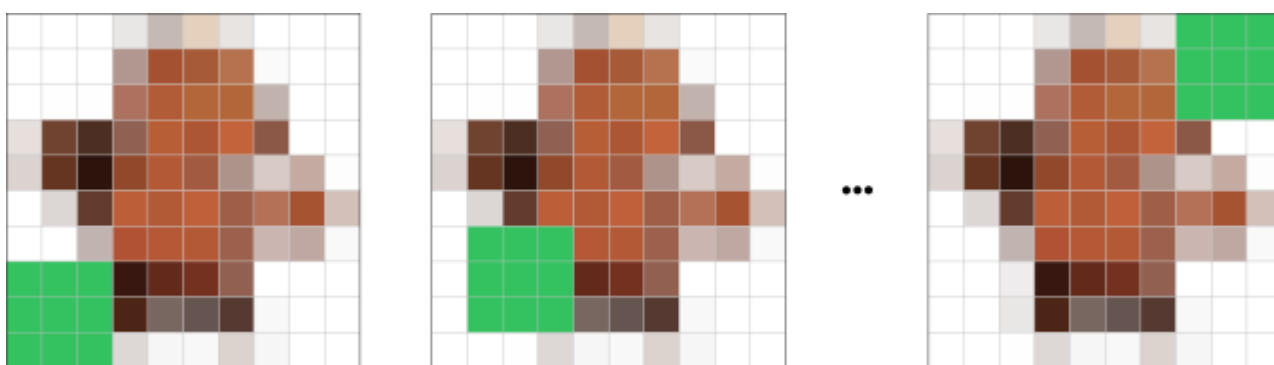
VIDEÓ TÖMÖRÍTÉS (2019-RU-02)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

Videóknak sok tárhelyre van szükségük, pedig gyakran tartalmaznak nagyon hasonló, két egymást követő pillanatképet.

A következő videó nagysága 10x10 képpont. A baloldali alsó sarokban található zöld négyzet nagysága 3x3 képpont. Ez mozog pillanatképről pillanatképig, mindig egy képpontot jobbra és felfele. A legvégén a jobb oldali felső sarokba érkezik meg.



Azért, hogy tárhelyet spóroljunk, a második pillanatképtől csupán azokat a képpontokat mentjük, amik megváltoztak.

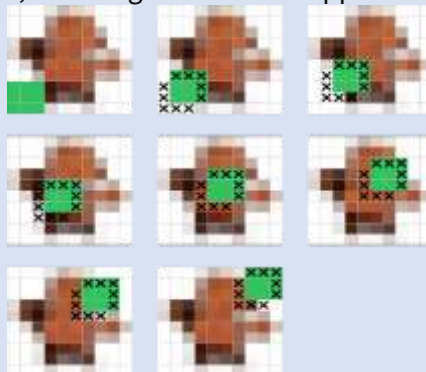
Hány képpontot kell menteni az összes videóhoz?

- A) 100
- B) 135
- C) 140
- D) 170



„D” a helyes válasz

Az egyes pillanatképek így néznek ki, ha a megváltoztatott képpontokat feljegyezzük:



Az első pillanatkép $10 \times 10 = 100$ képpontot tartalmaz. Minden további pillanatképért az eddig megváltoztatott képpontokat kell menteni. Ez öt képpont a négyzet bal alsó feléről, ami a háttér képkockáin keresztül kerül pótlásra, és öt képkocka felül jobbra a négyzetben, ami újonnan ábrázolja a négyzetet. Tehát minden pillanatképnél 10 képpont változik. A négyzetnek szüksége van 7 további pillanatképre ahhoz, hogy bal oldalon alulról, jobbra felfele tudjon mozogni, tehát kell $10 \times 7 = 70$ képpont a megváltoztatott képpontnak az eredeti 100 képponthoz hozzáadva. Így a helyes válasz D) 170.

MIÉRT INFORMATIKA?

Mint a feladatban is le van írva, a videótömörítés fontos szerepet játszik a mai világban. Ehhez az előzőekben leírt eljárás csak egy a sok közül, amit videók tömörítésére használnak.

Egy másik módszer, hogy elhagyunk meghatározott információkat, amik az emberek által nem észlelhetők. A JPEG formátum ezt használja ki. A különösen erősen tömörített képeknél fel lehet ismerni a tömbösítést, mert egy ilyen tömb a színek hasonlóságát nem tudja helyesen megjeleníteni/értelmezni (a színátmenetek nem jelennek meg jól).

Egy másik megoldás a színskála csökkentése. Erre az ötletre alapoz az MPEG-Standard. Különbséget tesz a különböző pillanatképtípusok között, hasonlóan, mint a feladatban. Egyik fajtája a pillanatképeknek (az úgynevezett Intra-képek) egy teljes pillanatképet ábrázol (hasonlóan a mi pillanatképünkhöz). Egy másik fajtája a megelőző pillanatkép („P-Bilder”, hasonlóan a többi pillanatképünkhöz) vagy a követő pillanatkép („D-Bilder”, ami ebben a feladatban nem fordul elő). Ahhoz, hogy a puffermemóriát kis méreten tudjuk tartani, és az átvitelihibáknál újra növelni tudjuk, a közökhöz Intra-képek vannak hozzáadva. Az erősen tömörített videóknál fel lehet ismerni P-képeket és D-képeket is, amikor egy fényszegény háttér hirtelen „beugrik”, habár az adott rész csak lassan mozgott.

A mentésszükséglet nem olyan egyszerű, mint ahogy a feladatban le van írva. A színértékekhez el kell menteni az adott helynek a megváltoztatott pixeleit is. Ez megváltozott pixelenként egy második faktort jelent. De még akkor is, ha a 800 helyett 240 memóriaegységet használunk, lenyűgöző helymegtakarításnak számít.

KULCSSZAVAK

Tömörítés, JPEG, MPEG

WEBOLDALAK

<https://hu.wikipedia.org/wiki/JPEG> ,
<http://oktel.hu/szolgalattas/kamerarendszer/keprogzitok/a-keptomorites/>

ERDEI FAFÉLÉK (2019-SI-02)

KISHÓD – NEHÉZ

BENJAMIN – KÖZEPES

KADÉT – KÖNNYŰ

Bence rendez egy partit hat hód barátjának, rengeteg ízletes finom fával. Sajnos azonban mind a hét hódnak megvan a maga ízlése:

A hódok ...	... ezek azok a fafajták, amiket szeretnek:
Anna	juhar, tölgy, kőris, fűz
Bence	tölgy, hárs, fűz
Csaba	tölgy
Dóra	nyír, kőris
Emese	juhar, nyír, fűz
Frédi	tölgy, kőris
Gyuri	juhar, hárs



Bence azonban szeretne spórolni a beszerzésen és lehetőleg minél kevesebb fafajtát vásárolna.

De azt is szeretné, hogy minden hód számára legalább egy ízletes fafajta jusson a kedvenceiből.

Legalább hány fafajtát kell beszereznie Bencének?

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5
- F) 6



„C” a helyes válasz

Bencének mindenképpen kell szerezni egy tölgyet Csabának. Ezzel elégedett lesz Anna, maga Bence, és Frédi is.

Dórának, Emesének és Gyurinak nincs közös kedvenc fatípusa. De két másik fajtaival eleget tud tenni az igényeiknek, mint például nyír és juhar. Összeségében Bencének három fafajtát kell beszereznie, hogy minden hód számára jusson ízletes fatípus.

23 óránál kevesebb idő alatt nem tudnak végezni, hiszen így mindketten folyamatosan, szünet nélkül dolgoznak. Ugyan össze tudnánk csoportosítani a feladatokat két 22 órás részre is $(B(3)+E(10)+F(9))$ és $(A(2)+C(5)+D(7)+G(4)+H(6))$, de ekkor az egymásra való várakozásnál keletkeznie kellene plusz időnek (hiszen a 2. hód még nem készül el az F feladathoz szükséges feladatokkal).

MIÉRT INFORMATIKA?

A feladatban a hódok ízletes fafajtája egy táblázatban van megadva. A kérdés megválaszolásához hasznos lett volna egy olyan táblázat mely a fatípusokhoz teszi az azt kedvelők nevét.

A fafajták ...	... az alábbi hódok szeretik:
juhar	Anna, Emese, Gyuri
nyírfa	Dóra, Emese
tölgy	Anna, Bence, Csaba, Frédi
kőris	Anna, Dóra, Frédi
hárs	Bence, Gyuri
fűz	Anna, Bence, Emese

A hét hódot hozzá lehet kötni a fafajták összmenységének egy részéhez. Ezután a vendéglátó Ben megkeresi a legkevesebb választási lehetőséget tartalmazókat ezekből a részcsoportokból, így mind a hét hód szerepel a kiválasztásban. Például a juhar, nyír és a tölgyfa választás minden hódot tartalmaz.

Az informatikában és a matematikában jól ismert az a probléma, hogy a részcsoportok közül a lehető legkisebbet válasszák ki annak érdekében, hogy ezeknek a részhalmazoknak az egyesítése adja az összmenységet. Ezt halmazfedési problémának vagy angolul set cover problem-nek is nevezik.

Habár ezt a hód feladatot nem volt nehéz megoldani, általában a set cover feladatok a legnehezebben megoldható, úgynevezett NP-teljes problémák közé tartoznak. Az informatika még nem ismer olyan módszert mellyel ezeket hatékonyan, vagyis az ember számára belátható időn belül optimálisan meg lehetne oldani. Azonban a legtöbb esetben ismertek olyan eljárások, amelyek ugyan nem mindig a legjobb, de aránylag jó megoldásokat találnak

KULCSSZAVAK

Halmazfedési probléma, NP-teljes probléma,

WEBOLDALAK

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

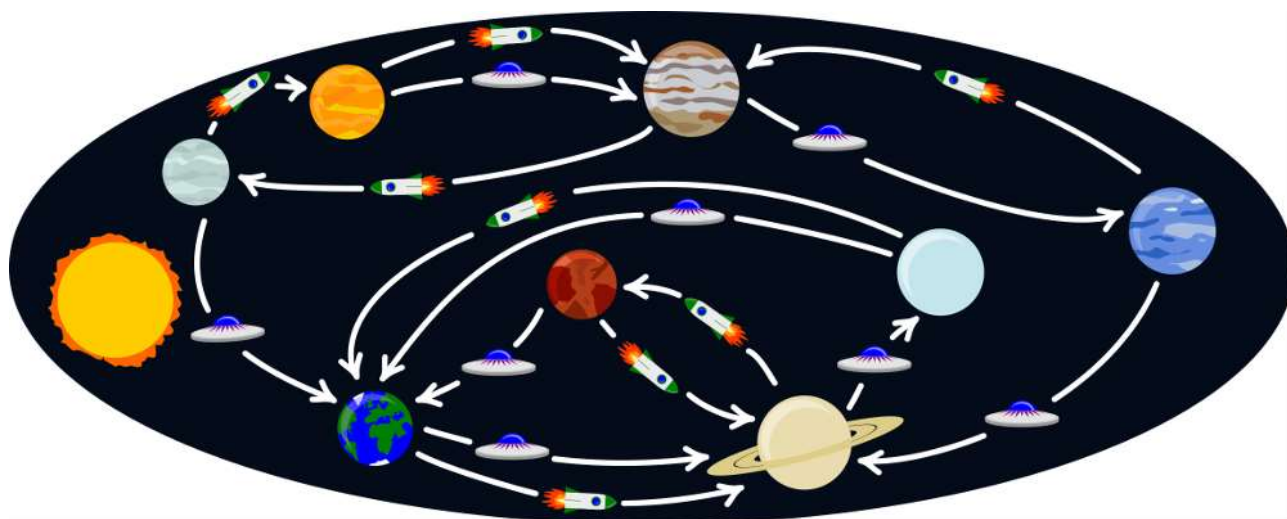


UTAZÁS A VILÁGYEGYETEMEN KERESZTÜL (2019-SI-03)

KISHÓD – NEHÉZ

BENJAMIN – NEHÉZ

A naprendszerünkben az űrhajósok rakétákon (🚀) és csészealjakon (🛸) tudnak a bolygók között repülni. A következő ábra mutatja a lehetséges repülési útvonalakat:



Egy űrhajós, aki a Vénuszról (🌸) a Szaturnuszra (♄) szeretne repülni, utazhat rakétán (🚀) vagy csészealjon (🛸) a Jupiterre (♃), onnan csészealjon (🛸) a Neptunuszra (♆), végül csészealjon (🛸) a célbolygóra, a Szaturnuszra (♄). Ha az űrhajós először rakétán (🚀) megy, a következő ábra írja le az utazását:



Helga az űrhajós jelenleg a Neptunuszon (♆) tartózkodik. Szeretne hazajutni a Földre (🌍). Az űrutazási iroda négy lehetőséget küld neki.

Melyik ajánlattal NEM jut vissza Helga a Földre (🌍).

- A) 🛸 🛸 🚀
- B) 🚀 🛸 🚀 🛸
- C) 🚀 🛸 🛸 🛸 🚀
- D) 🚀 🚀 🚀 🛸



„B” válasz a helyes

A B) válasz az egyedüli rossz útvonal.

Ha Helga ezt az útvonaltervet választja, végül újra a Neptunuszra landol. Először ugyanis egy rakétán utazik a Jupiterre, onnan csészealjon vissza a Neptunuszra, majd újra rakétán Jupiterre és vissza csészealjon a Neptunuszra.

A másik három ajánlattal el tud jutni a Földre. A megállók:

A Neptunuszról csészealjon a Szaturnuszra, onnan csészealjon az Uránuszra és végül rakétán a Földre.

A Neptunuszról rakétán a Jupiterre, Jupiterről vissza csészealjon a Neptunuszra, innen csészealjon a Szaturnuszra, csészealjon az Uránuszra és legvégül egy rakétán a Földre.

A Neptunuszról rakétán a Jupiterre, innen ismételtén egy rakétán a Merkúrra, és végezetül egy csészealjon a Földre.

MIÉRT INFORMATIKA?

A térképnek a bolygóközi utakról van egy különleges tulajdonsága: Minden bolygóról pontosan két út vezet el, egy rakétán és egy csészealjon. Emiatt mindig egyértelmű, hogy melyik bolygóra repülünk legközelebb.

Egy ilyen térkép egy determinisztikus véges automatát ír le. Egy ilyen automata áll valamennyi állapotból (ebben az esetben ezek a bolygók) valamennyi átmeneti állapotból (ebben az esetben a nyilak, amiknek a segítségével az űrhajósok egyik bolygóról a másikra tudnak repülni), illetve valamennyi bemenetből (a mi esetünkben ezek a rakéták, illetve az csészealjak). Ezen kívül van egy kezdő (Neptunusz) és több elfogadó állapot (a mi esetünkben Föld). A térképet ezáltal nevezhetjük állapotdiagramnak vagy állapotátmeneti táblának.

A determinisztikus véges automatákat több esetben is használják, mivel könnyű programozni. Tipikus példái: kávégépek, mosogatógépek, vagy akár italautomaták. De ezt használjuk akár a helyes szó/szövegfelismerésben is (például, hogy egy szöveg tartalmaz-e e-mail címet). A véges automatákat a nyelvtan és a mesterséges nyelvek különböző osztályaiban is tudjuk használni kapcsolatok felépítésére és az egyik világról a másikra ugrásban. Ez sok probléma megoldásában tud segítséget nyújtani.

Az űrutazási irodának egy másik feladata is volt. Keresniük kellett az állapotdiagrammon egy lehetséges utat, az egyik állapotból egy másikba. Ebben olyan az állapotdiagram, mintha egy irányított gráfról kellene leolvasni a lehetséges útvonalat, hogy az egyik csúcsból a másikba hogyan lehet eljutni irányított élek mentén. Ezekhez vannak szabványos algoritmusok, hogy az űrutazási irodának ne kelljen mindig előlről elkezdenie a keresést.

KULCSSZAVAK

Determinisztikus véges automaták, algoritmus, állapotdiagram

WEBOLDALAK

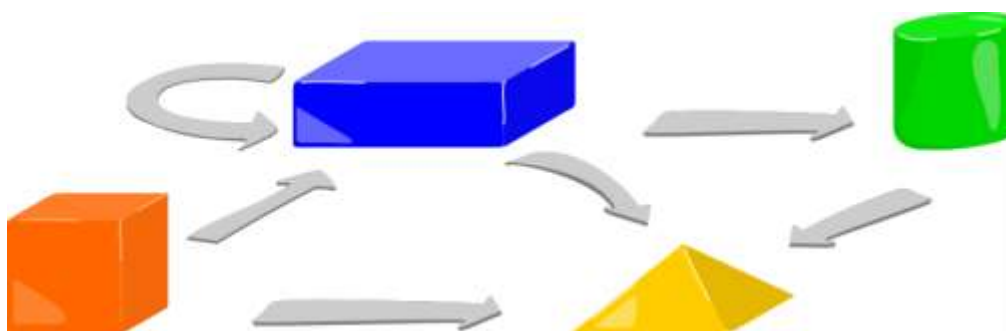
https://hu.wikipedia.org/wiki/Determinisztikus_v%C3%A9ges_%C3%A1llapot%C3%BA_g%C3%A9p



TORONY (2019-SK-03)**BENJAMIN – KÖZEPES****KADÉT – KÖNNYŰ**

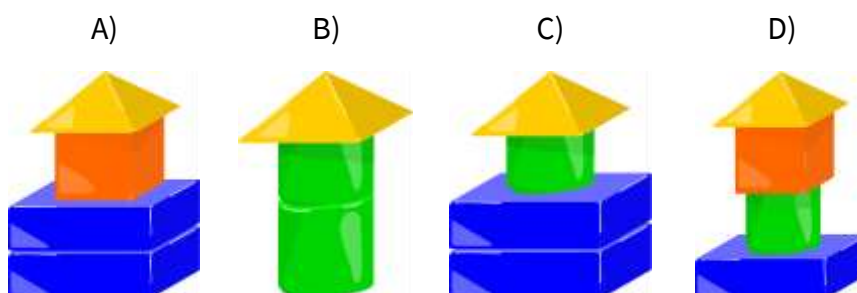
Laci kistestvére szabályokat hozott arra, hogyan rakja össze az építőkockáit. Ezt le is rajzolta.

- Laci egy szabadon választott építőelemmel kezdhet.
- Mindig követnie kell a nyilakat. Ha egy építőelemből több nyíl indul ki, kiválaszthatja, melyik legyen a következő. Ha egy nyíl visszavezet ugyanahhoz az építőelemhez, még egy ugyanolyat választhat, amit rátesz a toronyra.
- Ha az utoljára használt építőelemtől nincs több nyíl, Laci nem tehet több építőkockát a toronyra.



Laci négy különböző tornyot épített a kistestvérének.

Melyik tornyot építette Laci, a kistestvérének szabályai alapján?



„C” válasz a helyes

Az A) válasz tornya helyesen indul két kék téglatesttel. A második kék téglatestet egy narancssárga kocka követi. Mivel nincs nyíl a kék téglatesttől a narancssárga kockába, az A) válasz nem helyes.

A B) válasz tornya helyesen indul egy zöld hengerrel. A zöld hengert egy másik zöld henger követi, mivel nincs visszavezető nyíl a zöld hengerbe, ezért a B) válasz is helytelen.

A C) válasz tornya helyesen, két kék téglatesttel kezdődik. A második kék téglatestet egy helyesen lehelyezett zöld henger követi, és ezután egy sárga piramis. A sárga piramisból nem vezet már nyíl, ami helyes, mivel a sárga piramis a torony legfelső építőeleme. Tehát a C) válasz a helyes.

A D) válasz tornya is helyesen indul, egy kék téglatesttel. Ezt – helyesen – egy zöld henger követi. Ezután viszont egy sárga kocka következik, ami helytelen, mivel a zöld hengerből nem megy nyíl a sárga kockába. Tehát a D) válasz is helytelen.

MIÉRT INFORMATIKA?

A toronyra vonatkozó szabályok azon a tényen alapulnak, hogy a torony aktuálisan legfelső építőeleme határozza meg, mely építőelemek megengedettek következőnek. Így a torony aktuálisan legfelső építőeleme a torony jelenlegi állapota. A szabályok meghatározzák, hogy melyik következő állapotba kerüljön a torony. A nyilakkal ábrázolt grafikon egy állapotdiagram vagy állapotátmeneti diagram. Mivel az összes építőelem használható alsó építőelemként, minden építőelem lehetséges kezdőállapot. A sárga piramis az egyetlen építőkö a torony befejezéséhez (ha korábban még nem esett le).

A toronyépítés ezen szempontjai egy úgynevezett nemdeterminisztikus véges automatát írnak le. Azért hívják nemdeterminisztikusnak, mert vannak olyan állapotok, amelyekben az útvonalak közül választani lehet: egy kék kocka után egy másik kék kocka, egy zöld henger vagy egy sárga piramis következhet. Azért hívják végesnek, mert a lehetséges állapotoknak véges száma van: a négy építőelem egyike. Ugyanakkor végtelen magas tornyot építhetnénk elméletileg ... de végtelen számú kék kockára lenne szükségünk, és a magas tornyok néha ledőlnek (gyakran az építők nagy öröme).

A nemdeterminisztikus véges automaták modelljét gyakran használják az informatikában. Kiválóan alkalmasak nagyon különböző dolgok leírására: a szoftvermodulok vagy akár a teljes programok viselkedése, az egyszerű nyelvi struktúrák, a hardverkomponensek kölcsönhatása és még sok más. Egy ilyen formális leírás segítségével kipróbálható, hogy a szoftver a kívánt módon viselkedik ... vagy a torony helyesen van-e felépítve.

KULCSSZAVAK

Nemdeterminisztikus véges automaták fa, állapotdiagram

WEBOLDALAK

https://hu.wikipedia.org/wiki/Nemdeterminisztikus_v%C3%A9ges_%C3%A1llapot%C3%BA_g%C3%A9p



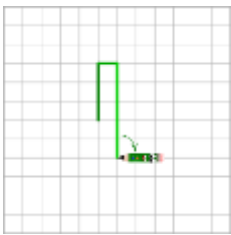
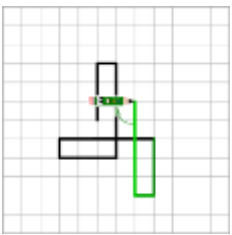
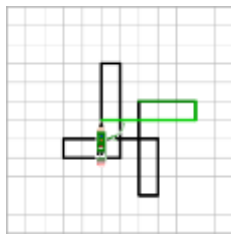
ROBOT (2019-SK-04)

JUNIOR – NEHÉZ

SENIOR – KÖZEPES

Egy robot egy négyzetrácsos papíron mozog és egyeneseket rajzol. Három szám segítségével irányítható.

Ha a 3  1  5  számokat adjuk neki, ezt a mintát rajzolja ki:

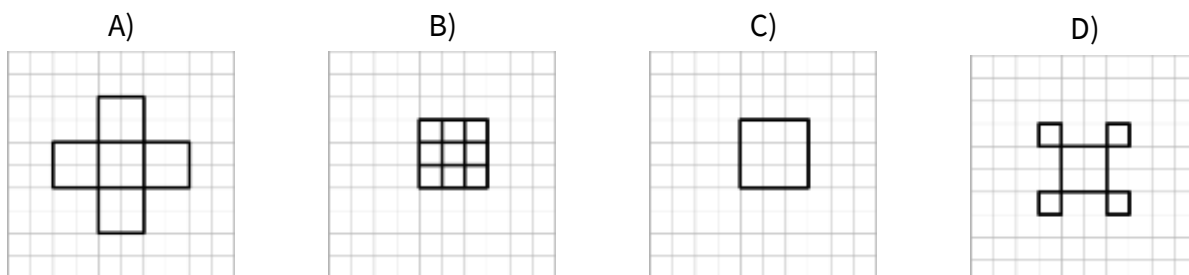
Első ismételés	Második ismételés	Harmadik ismételés	Negyedik ismételés
			

Ehhez összesen négyszer ismétli meg a következő lépéseket:

- Menj előre annyi négyzetnyit, ami az első szám helyén meg van adva
- Fordulj egy negyed fordulatot jobbra
- Menj előre annyi négyzetnyit, ami a második szám helyén meg van adva
- Fordulj egy negyed fordulatot jobbra
- Menj előre annyi négyzetnyit, ami a harmadik szám helyén meg van adva
- Fordulj egy negyed fordulatot jobbra

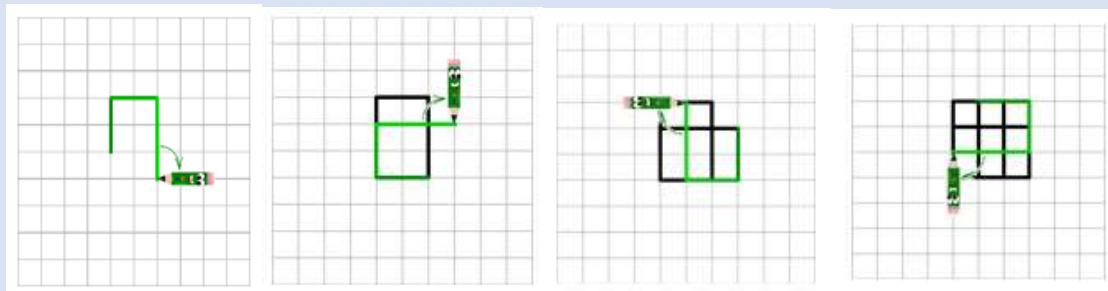
A robotnak ezeket a parancsokat adjuk meg: 2  2  3 

Hogy néz ki a megrajzolt minta?



„B” válasz a helyes

A **2**  **2**  **3**  parancssor azt jelenti, hogy a robot előre megy kettőt, majd elfordul egy negyed fordulatot jobbra, megint két rácsot megy előre, utána egy negyed fordulat jobbra, és végül három rács előre, és egy negyed fordulattal zár jobbra. Ezt a rajzot kaptuk ezután:



Ez pontosan megegyezik a B) válasszal.

A másik három ábra is megrajzolható a robottal, csak más számokat kell neki megadni.

A)	C)	D)
4  2  4 	3  3  3 	1  4  1 

MIÉRT INFORMATIKA?

A rajzolórobot ebben a feladatban csak nagyon egyszerű programokat tud végrehajtani. A programozási nyelv, amit a robot megért, összesen három szám. Minden programnak pontosan három számból kell állnia, amit fordulat szimbólum követ. Ezen kívül be van építve a programba, hogy a robot pontosan négyszer ismételve meg a parancssort, akár akarjuk, akár nem.

A legtöbb robot és számítógép megért nagyon sok összetett és komplex programozási nyelvet is. A legtöbb programozási nyelv hasonló alaptulajdonságokkal rendelkezik:

- 1.) A programok tetszőleges mennyiségű utasításból állhatnak, amik egymást követik. (Szekvenciák)
- 2.) Ismétlés parancsoknak különböző fajtái használhatók, de nem kötelezően – egy adott feltételtől függ a végrehajtásuk. (Ciklusok)
- 3.) Ezekon kívül léteznek meghatározott utasítások (Elágazások), amelyek a programállapottól függenek és különböző programkimeneteket tudnak eredményezni.

Egy programozási nyelvvel, ami ciklusokat és elágazásokat tartalmaz, minden kiszámítható és leírható. Az informatikában ezeket a programozási nyelveket univerzálisnak vagy Turing-teljesnek is nevezik.

A feladat robotja egy klasszikus példája annak a környezetnek, amiben megtanulhatunk programozni. A robot helyére egy teknőcöt állítunk, ami egyeneseket tud rajzolni. Ilyen teknőcgrafika például a Comenius Logo vagy az Imagine Logo.

KULCSSZAVAK

Strukturált programozás, Turing-gép. LOGO

WEBOLDALAK

https://hu.wikipedia.org/wiki/Struktur%C3%A1lt_programoz%C3%A1s

<https://hu.wikipedia.org/wiki/Turing-g%C3%A9p>

[https://hu.wikipedia.org/wiki/Logo_\(programoz%C3%A1si_nyelv\)](https://hu.wikipedia.org/wiki/Logo_(programoz%C3%A1si_nyelv))

HÓDHÁLÓ (2019-TH-08)

KADÉT – NEHÉZ

JUNIOR – KÖZEPES

SENIOR – KÖNNYŰ

Három világosbarna és három sötétbarna hód úszik át egy csatornarendszeren alulról felfelé. Minden csatornakereszteződésnél találkozik két hód. Ha a hódoknak különböző a színük, akkor a világosbarna hód úszik balra és a sötétbarna jobbra. Ha egyforma színűek, az egyik úszik jobbra másik balra (ahogy a képek is mutatják).



A végén a következő sorrendben kell megérkezniük a hódoknak: sötétbarna, világosbarna, sötétbarna, világosbarna, sötétbarna, világosbarna.

Hogyan kell a három világosbarna és három sötétbarna hódnak elindulnia, hogy a helyes sorrendben érkezzenek be?



- A) sötét barna, sötét barna, világos barna, sötét barna, világos barna, világos barna
- B) sötét barna, sötét barna, világos barna, világos barna, világos barna, sötét barna
- C) világos barna, világos barna, világos barna, sötét barna, sötét barna, sötét barna
- D) sötét barna, világos barna, sötét barna, világos barna, sötét barna, világos barna



„A” válasz a helyes

Két helyes válasz van:



Mind a kettő helyes megoldás, de mi csak az egyiket adtuk meg A válaszként.

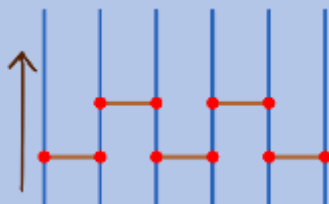
Annak érdekében, hogy egy sötétbarna hód érkezzen a bal szélső helyre, egyetlen világosbarna hódnak sem szabad átúsznia balról az első kereszteződésen, mivel balra kellene akkor úsznia. Így a két bal kiindulási helyet két sötétbarna hódnak kell elfoglalnia.

Ugyanez érvényes a jobb oldali pozícióra is. A jobb szélére egy világosbarna hódnak kell érkezni, ami csak úgy lehetséges, ha a két jobb oldali pozícióból világosbarna hódok indulnak.

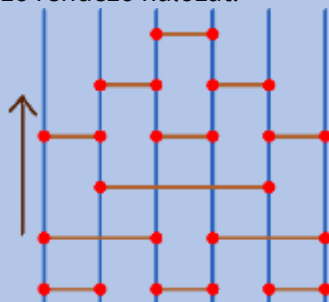
A két középső helyen, teljesen mindegy, hogy milyen sorrendben állnak a világosbarna és sötétbarna hódok. Akármilyen a sorrend, az első kereszteződésnél a világosbarna hód balra a sötétbarna hód jobbra fog úszni.

MIÉRT INFORMATIKA?

A hódok csatornarendszere, illetve a szabály, hogy ki az, aki jobbra és ki az, aki balra úszik, egy rendező hálózattal ábrázolható. Egy rendező hálózatban az adatok egy egyenes mentén (ebben a feladatban a csatornák) haladnak, és minden kapcsolatnál (a kereszteződések) ellenőrzik, hogy kicserélhetőek-e vagy sem. Egy sötétbarna hód feleljen meg egy 0-nak, egy világosbarna hód pedig 1-esnek. Így néz ki egy rendező hálózatban:



Mellesleg, egy teljes és minimális rendező hálózat ehhez a feladathoz hasonlóan nézne ki, jól látható, hogyan integrálódik a feladatból származó rendező hálózat:



A rendező hálózatok különösen akkor hatékonyak, ha az összehasonlítások párhuzamosan futtathatóak. Nagyobb adatmennyiségek esetén, nehéz megtalálni az optimális rendező hálózatot.

Általánosítva a hódok csatornarendszere felfogható úgy, mint egy kábelrendszer a számítógépekben vagy mint az internethálózat. Itt a csatornák a kábelek a routerek között.

KULCSSZAVAK

Rendező hálózat, hálózat, hálózati eszközök

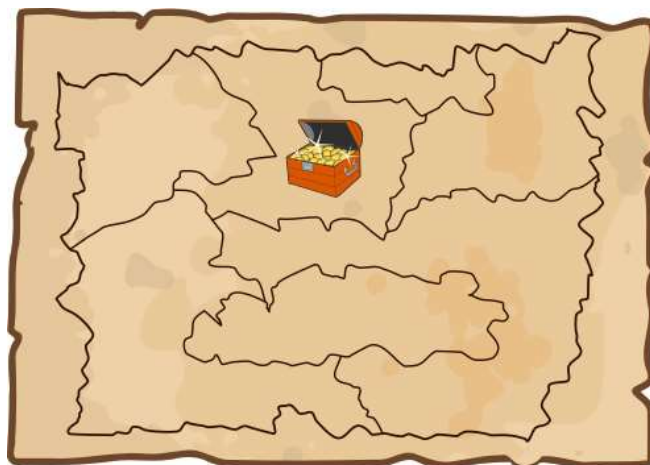
WEBOLDALAK

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

KINCSES TÉRKÉP (2019-VN-04)

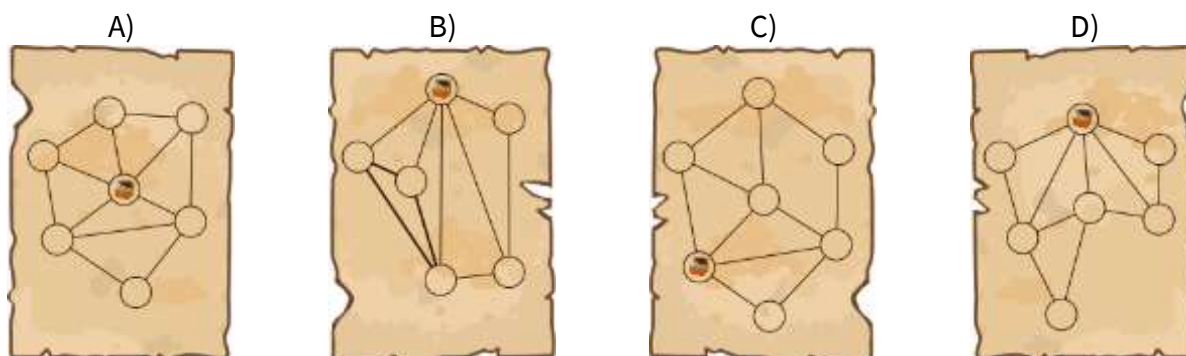
SENIOR – NEHÉZ

A hódok királya hét tartományon uralkodik, melyek határai a térképen láthatóak. Az egyik tartományban elrejtett egy kincset:



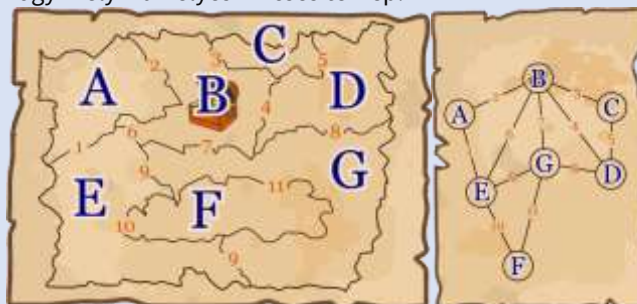
A király készített egy kincses térképet, ahol az egyes tartományokat körökkel ábrázolta, és megjelölte a kincset tartalmazó tartományt. Két kört akkor kötött össze, ha az egyes tartományoknak van közös határa. Hogy a rablókat megakadályozza abban, hogy megtalálják a kincset, még három hamis kincses térképet is készített.

Melyik a valódi kincses térkép?



„D” válasz a helyes

A következő nézetben az egyes területeket az A, B, C, D, E, F és G betűkkel jelöltük meg. Az egyes határterületeket pedig 1-től 11-ig beszámoltuk. Mivel az E és a G terület esetében két külön határ is van, de a meghatározás arra vonatkozott, hogy van közös határuk, ezért ugyanúgy 9-cel jelöltük mindkettőt. A jelölések a segítségével gyorsan megbizonyosodhatunk róla, hogy melyik a helyes kincses térkép.



Az A) válasz hamis: az A, C és F három tartomány mindegyikének csak két szomszédja van. Ezért a kincses térképen három olyan körnek kellene szerepelnie, melyekből csak két vonal indul ki (két másik tartománnyal van összekötve). De itt csak egy olyan kör szerepel.

A B) válasz sem lehet helyes, mivel csak hat kör szerepel, de hét tartományunk van.

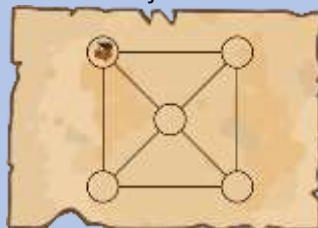
A C) válasz sem helyes: a kincset tartalmazó tartománynak öt határos szomszédja van (A, C, D, E és G), tehát öt vonalnak kell kiindulnia a kincessel jelölt körből. De ezen a térképen csak 4 kiindulás szerepel.

MIÉRT INFORMATIKA?

Ebben a feladatban egy térképet gráf segítségével reprezentálunk. Egy ilyen gráf a térkép absztrakcióját, leképezését jeleníti meg.

Mint minden absztrakciónál, így itt is, a lényegtelen információk elvesznek: pl. az egyes tartományok földrajzi elhelyezkedése. Habár a B tartomány a D tartománnyal azonos vonalban (magasságban) helyezkedik el, a gráfban különböző szinteken szerepelnek.

A fontos információk, mint a határok, továbbra is elérhetőek lesznek – az absztrakt modell tartalmazni fogja azokat. De egy helyes absztrakciós modellben is benne van az a veszély, hogy egy látszólag lényegtelen információ elvesz: a következő gráf (melyet egy másik államról készítettünk), forgásszimmetrikus, tehát nem világos, hol lehet elrejtve a kincs.



Egy gráf csomópontokból (ebben a feladatban a körök) és élekből (ebben a feladatban a vonalak) áll. Ebben a feladatban a gráf élei arra szorítkoznak, hogy két csomópontot kössenek össze. Lehetőségünk lenne arra is, hogy két csomópontot több éllel kössünk össze, így megjeleníthetnénk az E és G tartományok közötti mindkét határszakaszt.

Az informatikában a gráfokat rendszeresen használjuk információk elvonatkoztatására. Sok esetben maga az absztrakció (elvonatkoztatás) adja a probléma megoldását.

KULCSSZAVAK

Gráf, térkép, reprezentáció

WEBOLDALAK

<https://hu.wikipedia.org/wiki/Gráf>

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

TÁMOGATÓINK, KÖSZÖNETNYÍLVÁNÍTÁS

A feladatok készítésében közreműködtek:

 Tony René Andersen	 Maria Suyana Datzko	 Felipe Jiménez
 András Béni	 Sarah Estrella Datzko	 Anna Laura John
 Haim Averbuch	 Susanne Datzko	 Mile Jovanov
 Michelle Barnett	 Robin Deklerck	 Ungyeol Jung
 Michael Barot	 Lanping Deng	 Ilya Kaysin
 Wilfried Baumann	 Marissa Engels	 Adem Khachnaoui
 Jan Berki	 Olivier Ens	 Injoo Kim
 Linda Bergsveinsdóttir	 Gerald Futschek	 Jihye Kim
 Daniela Bezáková	 Bence Gaál	 Vaidotas Kinčius
 Bence Márton Bolgár	 Arne Goeteyn	 Mária Kiss
 Laura Braun	 Sonali Gogate	 Jia-Ling Koh
 Graeme Buckie	 Arnheiður Guðmundsdóttir	 Sophie Koh
 Špela Cerar	 Martin Guggisberg	 Dennis Komm
 Mony Chanroath	 Vernon Gutierrez	 Anja Koron
 Marios Choudary	 Alexandra Hamza	 Bohdan Kudrenko
 Anton Chukhnov	 Juraj Hromkovič	 Regula Lacher
 Sébastien Combéfis	 Alisher Ikramov	 Greg Lee
 Kris Coolsaet	 Thomas Ioannou	 Inggriani Liem
 Allira Crowe	 Tiberiu Iorgulescu	 Judith Lin
 Andrew Csizmadia	 Takeharu Ishizuka	 Lynn Liu
 Valentina Dagienė	 M. Faiz Ahmad Ismail	 Violetta Lonati
 Christian Datzko	 Yong-ju Jeon	



HÓDÍTSD MEG A BITEKET – 2019-ES FELADATSOR

TÁMOGATÓINK, KÖSZÖNETNYÍLVÁNÍTÁS

 Vu Van Luan	 Wolfgang Pohl	 Maciej Sysło
 Karolína Mayerová	 Sergei Pozdniakov	 Márta Szabó
 Kata Mázsár	 Stavroula Prantsoudi	 Bundit Thanasopon
 Guillaume de Moffarts	 Nol Premasathian	 Monika Tomcsányiová
 Mattia Monga	 J.P. Pretti	 Peter Tomcsányi
 Samart Moodleah	 Milan Rajković	 Nicole Trachsler
 Anna Morpurgo	 Chris Roffey	 Jiří Vaníček
 Madhavan Mukund	 Andrea Schrijvers	 Troy Vasiga
 Tom Naughton	 Eljakim Schrijvers	 Csilla Verebéllyné Erdei
 Pia Niemelä	 Humberto Sermenó	 Ela Veza
 Tomohiro Nishida	 Vipul Shah	 Le Anh Vinh
 Assylkan Omashev	 Daigo Shirai	 Márton Visnovitz
 Henry Ong	 Taras Shpot	 Florentina Voboril
 Csaba Patkó	 Jacqueline Staub	 Michael Weigend
 Margot Phillipps	 Nikolaos Stratis	 Jing-Jing Yang
 Emese Pluhár	 Gabrielė Stupurienė	 Xing Yang
 Zsuzsa Pluhár		 Khairul A. Mohamad Zaki



HÓDÍTSD MEG A BITEKET – 2019-ES FELADATSOR

TÁMOGATÓINK, KÖSZÖNETNYÍLVÁNÍTÁS

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

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

