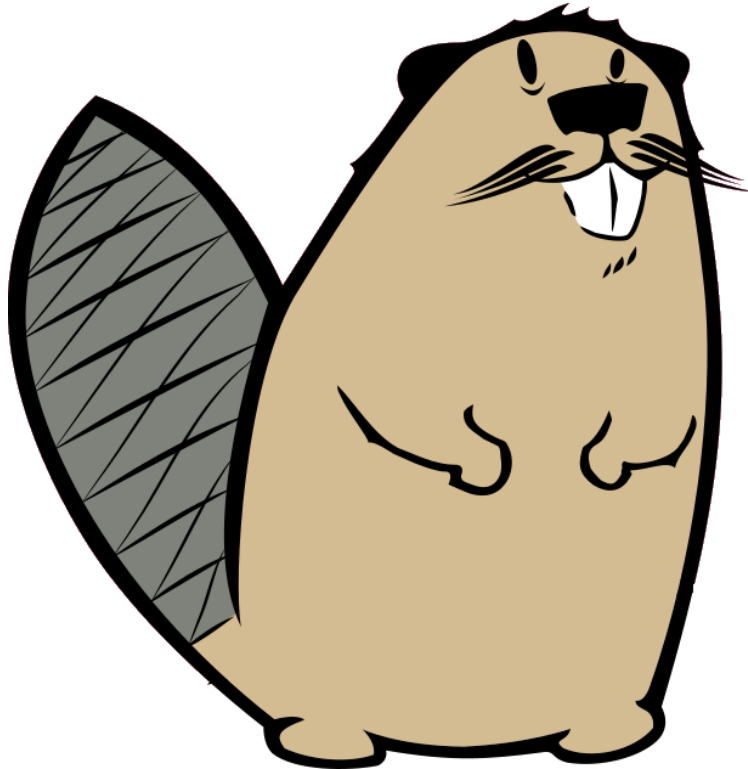


HÓDÍTSD MEG A BITEKET!

INFORMATIKAI GONDOLKODÁST TÁMOGATÓ, NEMZETKÖZI BEBRAS KEZDEMÉNYEZÉS MAGYAR
MEGVALÓSULÁSA





MI IS AZ E-HÓD?

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

A nemzetközi Bebras, melyhez 2016-ban már 50 ország kapcsolódott, 2015-ben elnyerte az Informatics Europe „Best Practices in Education” díját.

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

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

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

Magyarországon 2016-ban hatodik 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 2016-os 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 átbeszélhetőek 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;
- 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 megválaszolhatóak;
- 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 hod hetet követően kerülnek nyilvánosságra.

ÉRTÉKELÉS, PONTOZÁS

A Kishód korcsoportban 10, minden más korcsoportban kb. 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:

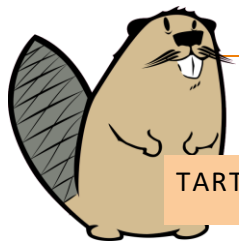
	<i>Könnyű</i>	<i>Közepes</i>	<i>Nehéz</i>
<i>Helyes válasz</i>	6 pont	9 pont	12 pont
<i>Helytelen válasz</i>	-2 pont	-3 pont	-4 pont



HÓDÍTSD MEG A BITEKET! – 2016-OS FELADATSOR

MI IS AZ E-HÓD?

Minden résztvevő kezdetben 54 pontot kap. Így 18 kérdés esetén összesen maximum 216 pontot érhet el, illetve, amennyiben minden kérdésre helytelen választ adott, pontjai száma 0-ra csökken.



TARTALOMJEGYZÉK

Mi is az e-hód?.....	2
Részvétel	2
Szabályok.....	3
Értékelés, pontozás	3
Tartalomjegyzék	5
Feladatok.....	7
Casino levél (2012-AT-14).....	8
Számfordító (2012-HU-01a).....	10
Szülinapi torta (2014-AU-01)	12
Fagylalthalmom (2014-CZ-10).....	14
Öntözés (2014-SI-07)	16
Ragasztott rajzok (2014-SK-01).....	18
Népszerűség (2015-CA-01)	20
Megosztás engedélyezése (2015-HU-01)	22
Karkötők (2015-MY-01)	24
A bűvész (2015-RU-07)	26
Csőfutam (2016-AT-03)	28
Cipőfűzés (2016-At-05).....	30
Rekurzív festés (2016-AT-06).....	33
Polcrendezés (2016-AU-03)	35
Strandászló nyelv (2016-AU-04).....	37
Alakzatjáték (2016-CA-09)	39
Át a sötét barlangon (2016-CH-04a).....	42
Szülinapi gyertyák (2016-CH-05).....	44
Virágok és napok (2016-CH-12)	46



Nim (2016-CH-23)	48
Ládapakolás (2016-CZ-02a)	50
Hierarchia (2016-CZ-03).....	52
Zászlótárolás (2016-CZ-04)	54
Hódchat (2016-DE-02)	56
IP-címek (2016-DE-03)	58
Emma, a teknősbéka (2016-DE-08b)	60
Kártyák és bóják (2016-FR-02)	62
Hidtervezés (2016-FR-03)	64
Vegetáriánus hajtóvadászat (2016-HU-02).....	66
Bonbonier (2016-HU-06)	68
Számozott mezek (2016-IE-03)	70
Kezet rázni (2016-IE-04).....	72
Egyszerre (2016-IE-05).....	74
Bulivendégek (2016-IS-02).....	76
Játék a golyókkal (2016-IT-02b)	78
Két lehetőség (2016-JP-02).....	80
3D labirintus (2016-JP-03)	82
Ebédszünet (2016-LT-03).....	84
Kix kód (2016-NL-04)	86
Középszűrő (2016-RU-02)	88
Villamoshálózat (2016-RU-04)	90
Önző mókusok (2016-RU-08).....	92
Halk vagy hangos? (2016-SK-01).....	94
Ki volt az? (2016-SK-02)	96
Virágzás (2016-SK-04)	98
Titkos üzenetek (2016-UK-06)	100



Szűk keresztmetszetek (2016-US-03b)	102
Köszönetnyilvánítás	104
Támogatóink	104
Köszönetnyilvánítás.....	104

FELADATOK

Kishód

2014-CZ-10, 2014-SK-01, 2016-SK-02

2014-SI-07, 2015-MY-01, 2016-AU-04, 2016-SK-01

2016-AT-03, 2016-HU-02, 2016-SK-04

Benjámín

2012-AT-14, 2014-SI-07, 2015-MY-01, 2016-AT-03, 2016-AU-03, 2016-AU-04

2012-HU-01a, 2016-CH-05, 2016-HU-02, 2016-IE-05, 2016-JP-03, 2016-SK-04

2015-HU-01, 2016-CH-12, 2016-CZ-03, 2016-DE-02, 2016-LT-03, 2016-UK-06

Kadét

2012-HU-01a, 2016-CH-05, 2016-CH-12, 2016-IE-05, 2016-IS-02, 2016-JP-03, 2016-SK-04

2015-HU-01, 2016-CZ-03, 2016-DE-02, 2016-FR-03, 2016-LT-03, 2016-US-03b

2015-CA-01, 2016-DE-08b, 2016-HU-06, 2016-IE-04, 2016-NL-04, 2016-RU-02

Junior

2015-HU-01, 2016-CH-23, 2016-CZ-03, 2016-FR-03, 2016-UK-06

2015-CA-01, 2016-CH-04a, 2016-FR-02, 2016-HU-06, 2016-IE-03, 2016-IE-04

2015-RU-07, 2016-CZ-02a, 2016-DE-03, 2016-IT-02b, 2016-JP-02, 2016-RU-04

Senior

2015-HU-01, 2016-DE-08b, 2016-IE-03, 2016-IE-04, 2016-NL-04, 2016-RU-02

2015-CA-01, 2016-CH-23, 2016-CZ-02a, 2016-DE-03, 2016-IT-02b, 2016-JP-02

2014-AU-01, 2016-AT-05, 2016-AT-06, 2016-CA-09, 2016-CZ-04, 2016-RU-08



CASINO LEVÉL (2012-AT-14)

BENJAMIN – KÖNNYŰ

Júlia e-mailben az alábbi reklámot kapta. Mindenképpen nyerni szeretne. Az iskolában azonban azt hallotta, hogy csak felnőtt játszhat szerencsejátékot, hogy az ember ritkán nyer és a megadott személyes adatokkal visszaélhetnek.

Mit tegyen?



- A) Törölje ki az e-mail-t?
- B) Adja ki magát az anyukájának és az ő személyes adataival vegyen részt?
- C) Kérje meg a bátyját, hogy vegyen részt a saját nevében?
- D) Egyszerűen saját személyes adataival vegyen részt?



„A” válasz a helyes: Júliának azonnal ki kell törölnie a levelet. Ez egy úgynevezett spam. A többi esetben személyes adatokat kell kiadnia, mellyel könnyen visszaélhetnek és kárt okozhatnak nekik. A legkevesebb, ha csak még több spam küldésével.

MIÉRT INFORMATIKA?

A spam-ek („levélszemetek”) drágák. Ha a felhasználók milliói minden nap munkaidőben spam-eket olvasgatnak és törölgetnek, az jelentős gazdasági károkat okozhat. A spam egy tömegmédium a károkozó szoftverek elterjesztésére. Sok felhasználó nem veszi észre, hogy a számítógépükön lévő adatok már visszaélnek. Csak mert egyszer egy spam-levélbe belekattintott, de védelmi rendszert (vírusirtót, tűzfalat, ...) nem telepített a gépére.

Az informatikusok nem csak azon dolgoznak, hogy minél hatékonyabb automatikus spam-szűrőket és védelmi szoftvereket, programokat tudjanak kifejleszteni, hanem azon is, hogy felhívják a figyelmet a kockázatokra és a megelőzés kötelezettségére mindenkinek, aki az interneten felhasználó.

Minden fiatalnak tudnia kell azt is például, hogy jó okkal van megtiltva neki az internetes szerencsejátékban való részvétel.



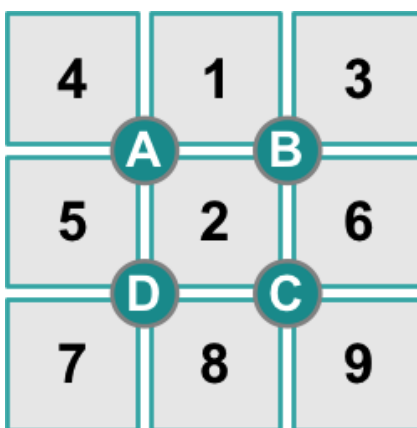
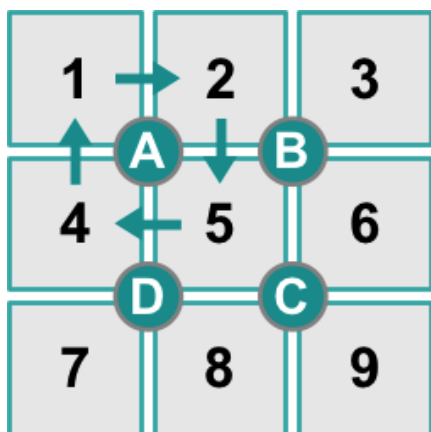
SZÁMFORDÍTÓ (2012-HU-01A)

BENJAMIN – KÖZEPES, KADÉT – KÖNNYŰ

A „számfordító” játékban a számokat 1-től 9-ig keverhetjük. A játék kezdetén a számok mindig a bal oldali képek megfelelően vannak elhelyezve.

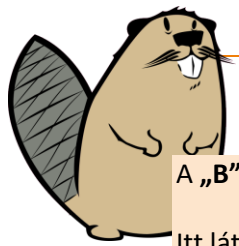
Ha az A, B, C vagy D gombok valamelyikét megnyomjuk, a gomb körül elhelyezkedő számok az óramutató járásával megegyező irányba fordulnak egyet.

Például az A gomb megnyomása után a számok a jobboldali képek megfelelően helyezkednek el.



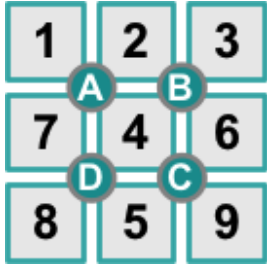
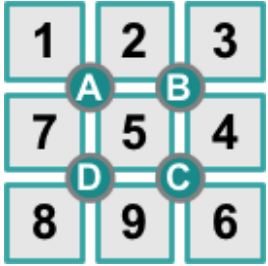
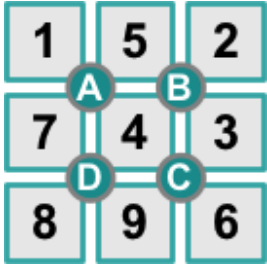
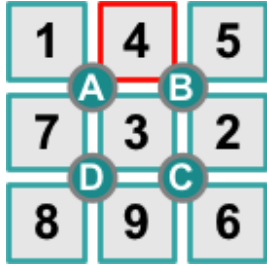
Kezdj egy új játékot, és nyomd le egymás után a D, C, B, B gombokat!

Hol helyezkedik el ezt követően a 4-es szám?



A „B” válasz a helyes.

Itt látható, hogy az egyes gombnyomásokat követően milyen helyzetbe kerülnek az egyes számok:

D lenyomása után	C lenyomása után	B lenyomása után	B lenyomása után
			

MIÉRT INFORMATIKA?

Amikor egy program lefut egy számítógépen, megváltoztatja az állapotát. Különböző részprogramok – mint itt pl. a gombok a feladataikkal – akár ugyanannak az elemnek az állapotát is megváltoztathatják.

A fenti játékban minden gomb 2 elem (számon) osztozik a táblán, a középső elemre pedig mind a 4 gomb befolyással van. Hogy ez ne okozzon felfordulást, ki kell zárni, hogy a részprogramok egy időben próbáljanak meg ugyanazzal az elemmel dolgozni.

Az informatikusok gondoskodtak róla, hogy legyen a számítógépekben az úgynevezett „atomi művelet”, mely már tovább nem bontható. Két atomi műveletnek tehát egymás után kell lefutnia. A feladatban szereplő játékban egy gomb lenyomása és ezáltal a kiváltott fordulás együtt számít egy atomi műveletnek.



SZÜLINAPI TORTA (2014-AU-01)

SENIOR – NEHÉZ

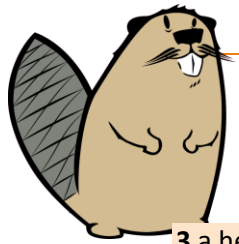
Beatrix az előző születésnapjára egy tortát akart sütni. A torta receptjében 8 fűszer is volt, de miután megsütötte a torta méregzöld színű lett. A vendégek megrémültek, amikor meglátták. De mert nagyon finom volt, Beatrix újra meg akarja sütni.

Elhatározta: a torta most már biztosan nem lesz méregzöld. Beatrix úgy sejtí, hogy csak egyetlenegy fűszertől lett a torta méregzöld.



Szisztematikusan végignézi, hogy melyik fűszer okozhatta a problémát. Eszébe jut, hogy több próbatortát süssön és közben több fűszerrel kísérletezzen.

Hány tortát kell sütnie Beatrixnak legalább, hogy teljesen biztosan meg tudja állapítani, melyik fűszer okozta a „problémát”?



3 a helyes válasz.

A 8 fűszert Beatrix a 3 próbatortán a következőképp osztotta fel:

Próbatorták: 1 2 1, 2 3 1, 3 2, 3 1, 2, 3

Fűszer: 0, 1, 2, 3, 4, 5, 6, 7

A 0-ás számú fűszert egyik próbatortán sem lehet megtalálni, az 1. fűszert csak az 1. próbatortán, a 2. fűszert csak a 2. próbatortán, a 3. fűszert az 1. és 2. próbatortán, stb. Ezzel Beatrix minden fűszert pontosan egyszer kombinált a próbatortákon. Beatrix már a harmadik próbálkozásra tudta, melyik próbatorták lesznek méregzöldek. Ezután már meg tudja határozni a problémás fűszert. Beatrix nem tud rájönni a megoldásra, ha háromnál kevesebbszer próbálkozik. Két próbatortánál csak négy kombináció létezik (--; 1; 2; 1 és 2), ebből nem lehet egyértelműen egymáshoz rendelni a 8 fűszert.

MIÉRT INFORMATIKA?

A kész tortákban információ van tárolva, Beatrixot viszont csak két érték érdekli: a „méregzöld szín” és a „nem méregzöld szín”. Beatrix kísérletében minden próbatorta olyan, mint egy bit a számítógépen.

Egy bitben pontosan egy vagy két érték lehet elmentve. Egy bit vagy „bent van” vagy „kint van”. Az informatikában ezeket az értékeket gyakran 1-es vagy 0-ás számjeggyel jelölik. Minél több bit áll a rendelkezésedre, annál több számot tud megjeleníteni – a bináris rendszer segítségével. Három bit megegyezik a fenti példával: Bitek 000 001 010 011 100 101 110 111 Szám 0 1 2 3 4 5 6 7.

Több bittel nagyobb számokat lehet előállítani, sőt bitekkel egészen más dolgokat is elő lehet állítani pl. betűket. Ehhez ezeket a dolgokat bizonyos számokhoz kell rendelni, amiket bitekkel újra elő lehet állítani. Bitekkel elő lehet állítani majdnem mindent, amit csak el tudsz képzelni. De egyvalamit biztosan nem: a végtelenséget. Ezért minden számítógépes rendszerben, mindegy mekkora, mindig véges a bitek száma.



FAGYLALTHALOM (2014-CZ-10)

KISHÓD – KÖNNYŰ

A LIFO Fagylaltozóban a kívánt fagyaltgombócokat egy tölcsérré halmozzák.

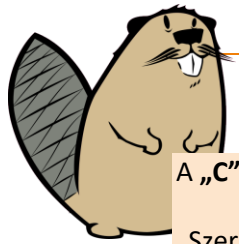
Pontosan abban a sorrendben, melyben a vendég kéri.

Mit kell mondania a vevőnek, ha a képen látható fagyalttölcsért szeretné?

Szeretnék ...

- A) ... egy csokoládét, egy mentát és egy áfonyát.
- B) ... egy csokoládét, egy áfonyát és egy mentát.
- C) ... egy áfonyát, egy mentát és egy csokoládét.
- D) ... egy áfonyát, egy csokoládét és egy mentát.





A „C” válasz a helyes.

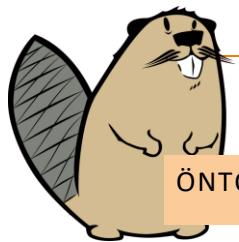
„Szeretnék egy áfonyát, egy mentát és egy csokoládét!” Amit először mond, az kerül elsőnek, legalulra a tölcsérbe. Amit utolsónak mond, az kerül utoljára, legfölülre a halomba. Az A válaszban a sorrend pont fordított. A B és a D válaszban nem a menta van középen.

MIÉRT INFORMATIKA?

A sorrend fontos! Ha a fagyaltfélétet más sorrendben mondanánk, egy másik fagyalttölcsér lenne belőle. Az informatikában megtanuljuk, milyen hasznos, ha valami rendezett. És hogy meg kell érteni, melyik rendezést melyik helyzetben kell alkalmaznunk. Ennek megértése nélkül, ahogy a fagyalmozó működik, nem tudnánk a kívánt fagyalttölcsért megrendelni. A helyzet megértése nélkül nem tudunk kellő programot készíteni. Ebben a feladatban a használt rendezés azt jelenti: „last in, first out” (LIFO, azaz utolsónak be, elsőnek ki)

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

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



ÖNTÖZÉS (2014-SI-07)

KISHÓD – KÖZEPES, BENJÁMIN – KÖNNYŰ

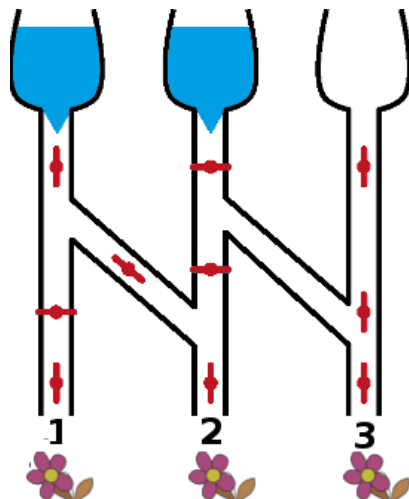
Amikor a szelep zárva van, a víz nem tud átfolyni.

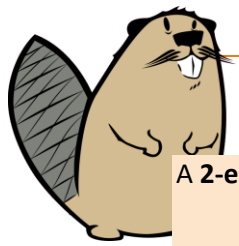


Amikor a szelep nyitva van, a víz átfolyik.

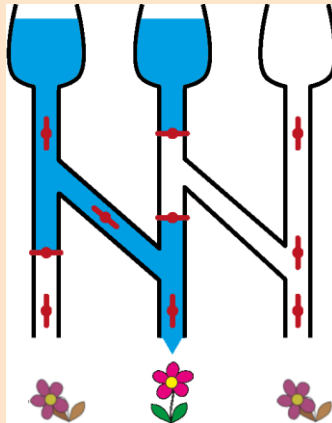


A három szomszjas virág közül melyik kap vizet, ha a szelepek így állnak?





A **2-es virág**. Az ábrán látható, hogy folyik ilyen szelep-állások esetében a víz.



MIÉRT INFORMATIKA?

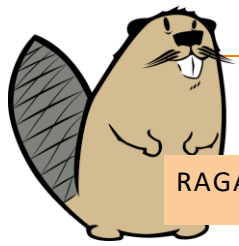
Informatikában ez az öntözőrendszer egy kapcsolás. A szelepek a kapcsolók, amelyeknek két állásuk van: be vagy ki vannak kapcsolva. A bemeneti tölcserék és a kapcsolóállások szerint terjed az információ (mint ahogy folyik vagy nem folyik a víz) a kapcsolókon keresztül a virágokig.

Az elektronikus készülékek elektronikus kapcsolásokat tartalmaznak, melyen keresztül az elektromosság folyik.

Az üvegszálkapcsolásokon keresztül az információ mint lézertény közlekedik.

Egy robot eszköznek, melynek a környezetben kell dolgoznia, az alkatrészei, kapcsolói hamar tönkre tudnak menni: erős mágneses hatás, sok nedvesség, szélsőséges hőmérséklet. Az ilyen robotoknak robusztus kapcsolókra van szüksége, melyekben hidraulikus olaj vagy sűrített levegő van.

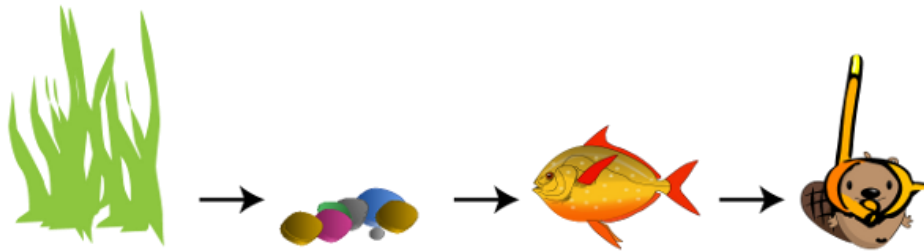
Az automatizálási technikában és a robotikában azonban a processzor-perifériákkal ellentétben (szenzorok és aktorok) folyadékos és hidraulikus kapcsolások vannak érvényben jelenleg is.



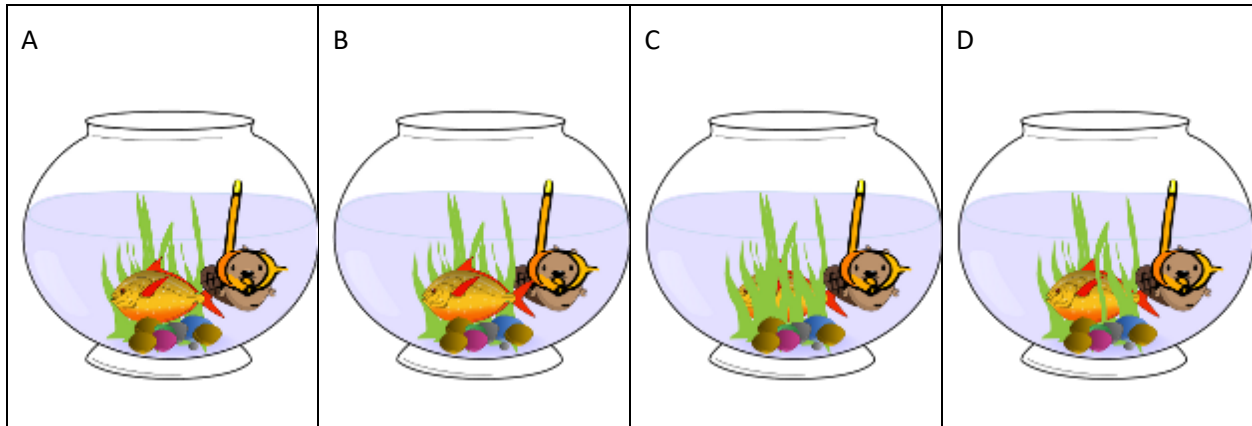
RAGASZTOTT RAJZOK (2014-SK-01)

KISHÓD – KÖNNYŰ

János rajzolt egy akváriumot. Feldíszítette kis ragasztott rajzokkal. Először ragasztott hínárt, majd köveket. Ezután a halat és végül a búvárhódot.



Hogy nézett ki a kép?



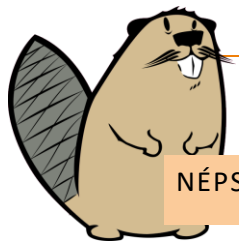


Az „A” válasz a helyes.

A képek itt vannak egymáson a helyes sorrendben. „B” rossz, mert a bűvárhód nincs egészen elől, rajta van a hal. „C” rossz, mert a hínár nincs egészen hátul. A hal mögötte van. A „D” válasz rossz, nem a hínár előtt úszik, hanem közöttte.

MIÉRT INFORMATIKA?

Az informatika több területén fontos szerepe van a sorrendnek. Itt például egy kép esetében, ami több képelemből tevődik össze. Egy másik sorrend más képet eredményezne, habár az egyes képelemek ugyanazok maradnának.

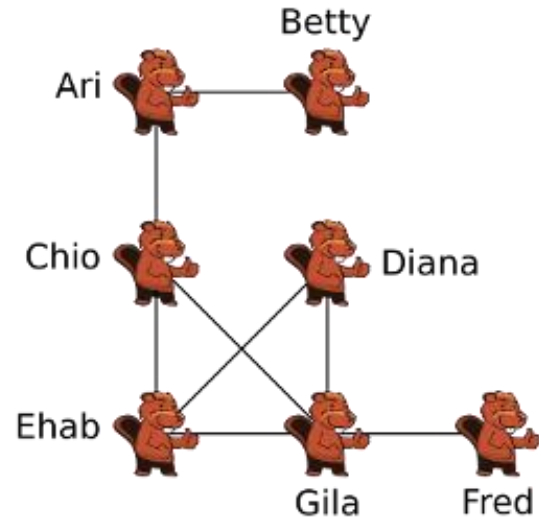


NÉPSZERŰSÉG (2015-CA-01)

KADÉT – NEHÉZ, JUNIOR – KÖZEPES, SENIOR – KÖZEPES

Hét hód regisztrált egy online hálózatra. A kép azt mutatja, a hálózaton belül mely hódok „barátok”: a barátok egy vonallal vannak összekötve. A nyári szünet után minden hód megosztott egy fotót a barátaival a hálózaton. Így a barátaik oldalán is megjelent a fénykép. Minden hód a saját oldalán és a barátaik oldalán lévő fotókat látja.

Kinek a fényképét látja a legtöbb hód?





Chio a helyes válasz.

Minden fénykép a barátok oldalán is látható és mindenki látja a barátaik oldalán lévő fotókat. Ezért a egy hód megosztott fényképét a saját barátaik és az ő barátaik láthatják.

<i>Hód</i>	<i>Barátai</i>	<i>Barátaik barátai</i>	<i>Az összes elért hód</i>
Ari	Betty, Chio	Ehab, Gila	4
Betty	Ari	Chio	2
Chio	Ari, Ehab, Gila	Betty, Diana, Fred	6
Diana	Ehab, Gila	Chio, Fred	4
Ehab	Chio, Diana, Gila	Ari, Fred	5
Fred	Gila	Chio, Diana, Ehab	4
Gila	Chio, Diana, Ehab, Fred	Ari	5

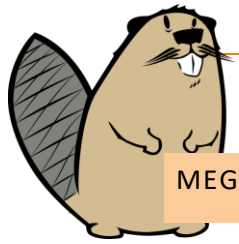
Ez egyenlő azoknak a hódoknak a számával, akiket maximum két vonallal el tudunk érni egy-egy hódtól. Ugyan Gilának van a legtöbb barátja, de ők majdnem mind csak egymás közt barátok. Chio összesen több hódot ér el.

MIÉRT INFORMATIKA?

Több jelenleg elterjedt szociális hálón használnak hasonló elgondolásokat a barátságok kezelésére. Ezekkel lehetővé válik, hogy egy megosztott képet vagy más oldalra írt bejegyzést több felhasználó lásson. A szociális hálózatok néhány éve óriási jelentést nyertek, többek között mert lehetővé teszik az emberi kapcsolatok vizsgálatát. Nagy hálózatokat csak számítógéppel lehet vizsgálni. Az informatika ehhez gráfokat, gráf-algoritmusokat használ, melyekkel például meg tudja határozni a hálózat tagjainak úgynevezett elérhetőségi fokát.

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

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



MEGOSZTÁS ENGEDÉLYEZÉSE (2015-HU-01)

BENJÁMIN – NEHÉZ, KADÉT – KÖZEPES, JUNIOR – KÖNNYŰ, SENIOR – KÖNNYŰ

A tanár az órájához szöveget keres az Interneten. Ezt le akarja fordítani egy másik nyelvre és azután a kinyomtatott fordítást a diákjai között ki akarja osztani. Talál is egy megfelelő szöveget, de az egy használati engedéllyel (CC BY-ND) és egy szerzői megnevezéssel van ellátva.

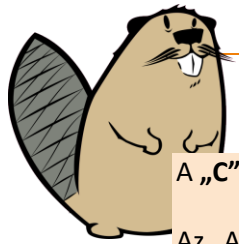


A „CC” a „Creative Common License”. Ez egy olyan engedély, ami megengedi a szövegek általános használatát és továbbadását bizonyos megszorítások, korlátozások mellett.

A „BY” megszorítás azt jelenti, hogy a szöveg továbbadásakor az eredeti szerzőt meg kell nevezni. Az „ND” megszorítás azt jelenti, hogy a szöveget változtatás nélkül lehet csak továbbadni.

Mit NEM tehet a tanár a szöveggel?

- A) A szöveg másolatát az eredeti szerző megadásával az iskola honlapján nyilvánossá teszi.
- B) A szöveget lefordítja egy másik nyelvre, a fordítást a saját számítógépén elmenti egy olyan feladattal együtt, aminek ő az eredeti szerzője.
- C) A szöveg egy oldalát egy másik nyelvre lefordítja és ebből kinyomtatott példányokat oszt szét a diákjai között.
- D) A szöveget az eredeti szerző megadásával kinyomtatja és fénymásolóval sokszorosítja.



A „C” válasz a helyes.

Az „A” válasz az „eredeti szerző megadásával” teljesíti a „BY” megszorítást. A „szöveg másolatát” teljesíti az „ND” megszorítást.

A „B” válaszban a másik nyelvre fordított változat használatát a „CC” nem korlátozza - a fordítást továbbadni nem lehet az „ND” megszorítás miatt. A fordítás miatt a tanár nem lett szerző és a szerző megadásának változtatása nincs a „BY” által korlátozva.

A „C” válaszban egy szövegrész kivágása, ugyanúgy, mint a közzéadott fordítás egy másik nyelvre az „ND” megszorítást szegi meg.

A „D” válaszban a kinyomtatás és fénymásolás a CC szerint nincs korlátozva. Más adathordozón való tárolás megengedett - amíg az itt szereplő „BY” és „ND” korlátozásokat betartják.

MIÉRT INFORMATIKA?

Az információs társadalomban és az informatikában a szerzői jogok kérdése összetett témakör. Általában nem könnyű eldönteni, hogy egy mű feltöltése, letöltése, meghatározott használata vagy továbbadása megengedett-e vagy sem. A „Creative Common License” elgondolást a szerzőkért, tervezőkért, programozókért és felhasználókért fejlesztették. Ezzel könnyebben megérthető, hogy egy adott helyzetben mit tehetünk anélkül, hogy szabályokat vagy szerződéseket szegnénk meg.

Fontos: Minden Hód feladatra érvényes a CC BY-NC-SA licenz! (Nevezd meg!-Ne add el!-Így add tovább!)

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



KARKÖTŐK (2015-MY-01)

KISHÓD – KÖZEPES, BENJÁMIN – KÖNNYŰ

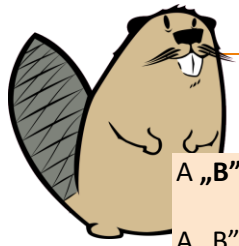
Lilinek különböző formájú gyöngyökből álló fűzött karkötője van. Egy nap elszakadt a karkötő és már nem is lehet megjavítani. Az elszakadt karkötő így néz ki:



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

Melyik teljesen ugyanolyan, mint Lili elszakadt karkötője?

A	B	C	D



A „B” válasz a helyes.

A „B” válasz karkötőjén a formák ugyanabban a sorrendben vannak, mint az elszakadt karkötőn.

Az „A” válaszban a narancssárga háromszög és a sárga csillag fel van cserélve.

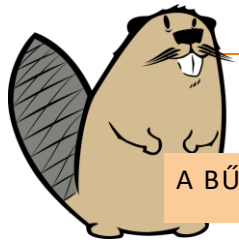
A „C” válasz karkötőjén a narancssárga háromszög és a kék hatszög van felcserélve.

A „D” válaszban többek között a sárga csillag és a zöld kör is rossz helyen van.

MIÉRT INFORMATIKA?

Az informatikában segít, ha egy mintát ismét fel tudunk ismerni. Érdekes, ha első látásra különbözőnek tűnő dolgokban felismerünk egy mintát. Ez problémák megoldására is igaz: ha egy új problémát ismerünk fel, ami egy már ismert problémához hasonlít és amit már valaki megoldott, akkor ez a megoldás segíthet az új probléma megoldásában. A feladat ennek a mintafelismerésnek egy részletével foglalkozik: arról szól, hogy megvizsgáljuk, a négy megadott megoldás közül melyiknek a sorrendje azonos a mintaformával.

Az informatikában algoritmusok egész sora létezik, amelyek automatikusan tudnak ilyet vizsgálni. Ilyeneket használnak pl. a szövegfeldolgozó alkalmazások. Összetett „reguláris kifejezések” minták meghatározott tömegét tudják azonnal felismerni.

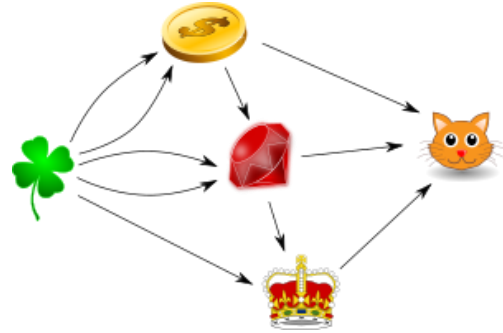


A BŰVÉSZ (2015-RU-07)

JUNIOR – NEHÉZ

A bűvész a varázslataival képes dolgokat átalakítani. Minden átalakításnál egy vagy több dolog eltűnik és valami új kerül a helyére. A bűvész négy átváltoztatásra képes:

- Két lóheréből egy érmét tud varázsolni.
- Egy érméből és két lóheréből egy gyémántot tud varázsolni.
- Egy gyémántból és egy lóheréből egy koronát tud varázsolni.
- Egy érméből, egy gyémántból és egy koronából egy kismacskát tud varázsolni.



Hány lóherére van szüksége ahhoz, hogy végül egy kismacskát varázsoljon?



11 a helyes válasz: A bűvésznak 11 lóherére van szüksége.

- 1 érme = 2 lóhere
- 1 gyémánt = 2 lóhere + 1 érme = 4 lóhere
- 1 korona = 1 gyémánt + 1 lóhere = 5 lóhere
- 1 kismacska = 1 érme + 1 gyémánt + 1 korona = 11 lóhere

MIÉRT INFORMATIKA?

A feladatban szereplő ábrát irányított gráfnak nevezzük az informatikában. Ezek csomópontokból (azok a dolgok, melyeket a varázsló át tud alakítani) és nyilakból állnak. Ennél a feladatnál egy nyíl A és B között azt jelenti, hogy „A szükséges B előállításához”. A feladatban szereplő gráf érdekessége, hogy két csomópont között akár több egyenértékű nyíl is lehet. Ezt multigráfnak nevezzük.

Egy gráf néha jobban átláthatóvá tesz egy feladatot, mint a szöveg. Gráfokkal sokféle szerkezetet modellezhetünk. Egy családfánál minden csomópont egy családtag és minden nyíl egy családi kapcsolat szülők és gyermekek között. Egy közlekedési térkép esetében minden csomópont egy állomás és minden nyíl egy közvetlen járat a két állomás között. Multigráfokkal modellezhetjük pl. az Internetet: minden csomópont egy weboldal és minden nyíl egy kattintható, másik oldalra mutató link. Itt előfordulhat, hogy egy weboldaltól több link is mutat akár ugyanarra a másik weboldalra - ezzel több nyíl szerepel két csomópont között.

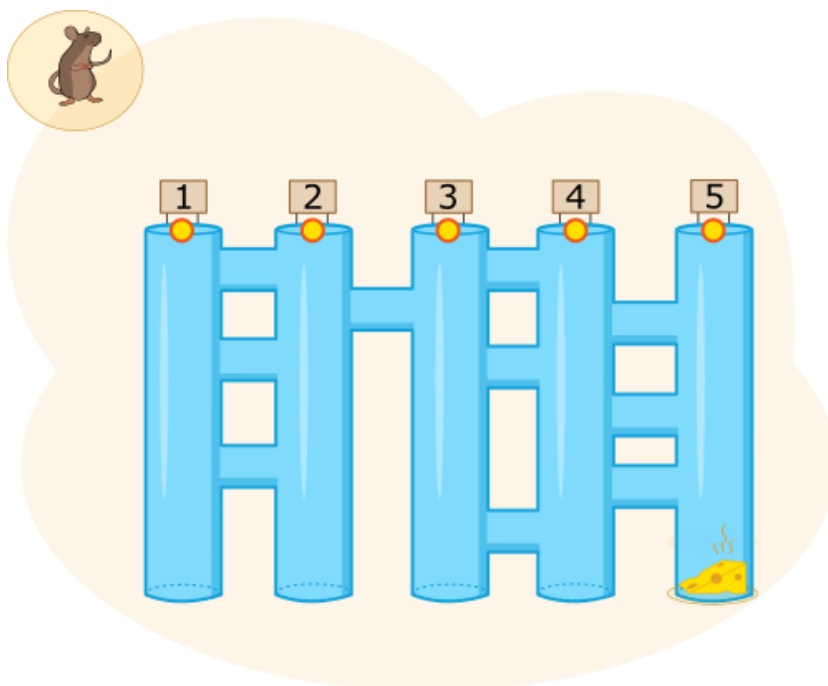
<https://hu.wikipedia.org/wiki/Gr%C3%A1f>

https://hu.wikipedia.org/wiki/Gr%C3%A1felm%C3%A9leti_fogalomt%C3%A1r



KISHÓD – NEHÉZ, BENJÁMIN – KÖNNYŰ

Egy egér áll fent, öt hosszú cső előtt. A hosszú csöveket rövidebb csövek kötik össze. Az egér szeretne eljutni a sajthoz, ami az egyik hosszú cső végén van, egészen jobbra lent.



A csövekben az egér mindig a következő szabályokat követi felváltva:

1. Addig fut lefele a hosszú csőben, amíg egy rövidebb cső nem keresztezi azt.
2. Ekkor átfut a rövidebb csövön (egy másik hosszú csőbe).

Melyik hosszú csőben induljon el, hogy eljusson a sajthoz?

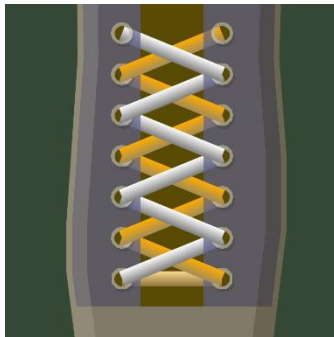


CIPŐFŰZÉS (2016-AT-05)

SENIOR – NEHÉZ

A hódok szeretik a különlegesen befűzött cipőket. Emellett egyedi cipőfűzőket használnak, amiknek az egyik fele narancssárga, a másik fehér.

Ezért készítenek egy robotkart, ami cipőfűzőket fűz be. Például a képen látható fűzés így megy:



1. Először a narancssárga véget a jobb oldali legelső karikán (fűzőlyukon) alulról felfelé fűzik be.
2. Ugyanezt megteszik a fehér fűzővéggel a bal oldali legelső lyuknál.
3. Majd a narancssárga véget egy lyukkal tovább viszik, át a másik oldalra és alulról fűzik felfele.
4. Ugyanezt megteszik a fehér fűzővéggel.

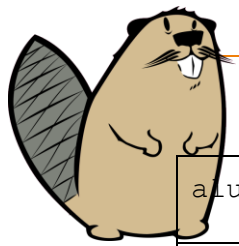
5. Ezt a két lépést (3. és 4.) addig ismétlik, amíg a cipő fűzése készen nem lesz.

Mindezt így tudják leírni a robotkarnak:

```
narancs: alulról
fehér: alulról
{
  narancs: előre csere alulról
  fehér: előre csere alulról
}
```

A robotkar ezeken felül még jó pár utasítást ismer és tud végrehajtani:

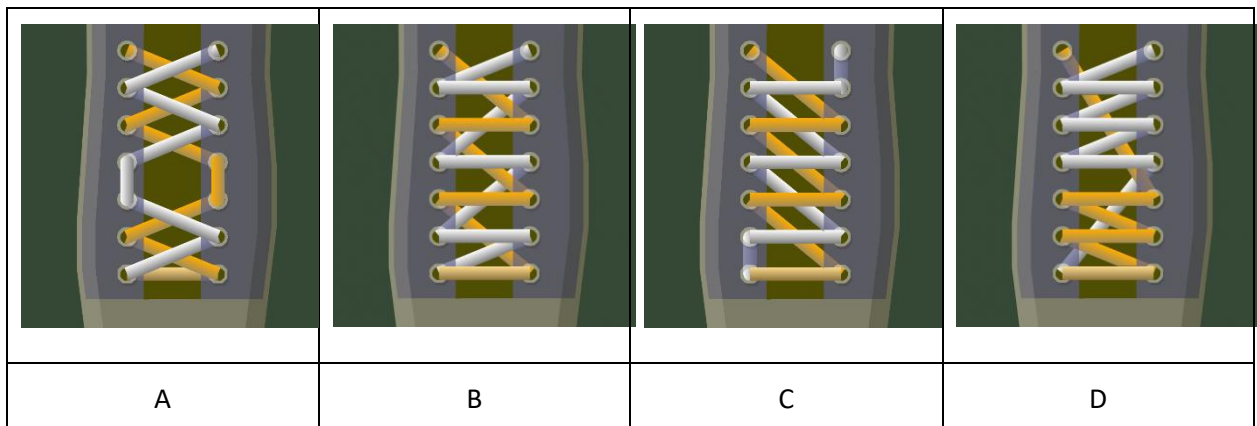
PARANCS	JELENTÉSE
{ ... }	A zárójel tartalma olyan gyakran kerül végrehajtásra, amilyen sokszor csak lehet.
*n{ ... }	A zárójel tartalma n-szer kerül végrehajtásra
narancs:	A narancs színű cipőfűzőre vonatkoznak a következő utasítások.
fehér:	A fehér színű cipőfűzőre vonatkoznak a következő utasítások.



alulról:	A fűzőt a lyukon alulról felfele fűzi át.
felülről:	A fűzőt a lyukon felülről lefele fűzi át.
előre:	A fűző a sorban eggyel előrébb kerül majd befűzésre.
hátra:	A fűző a sorban eggyel hátrább kerül majd befűzésre.
csere:	A fűző átkerül a másik oldalára a lyuk-sornak.

Hogy lesz befűzve a cipő, ha a robotkar a következő utasításokat kapja:

```
narancs: alulról  
fehér: alulról  
*2{  
narancs: előre csere alulról  
fehér: előre csere alulról  
}  
narancs: előre felülről  
fehér: előre felülről  
{  
narancs: előre csere alulról  
fehér: előre csere alulról  
}
```





Az „A” válasz a helyes.

Az első két lyuknál ugyanaz a fűzés, mint a példánkban (1-6. sor).

Ezután mindkét fűzővég eggyel előrébb kerül oldalváltás nélkül. (7. és 8. sor)

Innen már látható, hogy csak az első kép mutatja a helyes választ. Az utasítások szerint ugyanúgy megy tovább a fűzés, mint az elején.

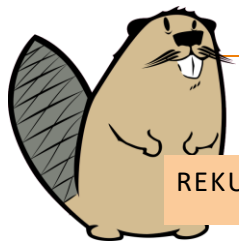
Ötlet: Próbáld meg leírni a többi fűzést!

MIÉRT INFORMATIKA?

Ebben a feladatban a robotkarhoz egy saját programozási nyelv került kifejlesztésre. Más programnyelvekhez hasonlóan utasításokat és programozási szerkezeteket tartalmaz: az egymás után végrehajtandó utasításokat (szekvenciák) és az ismételt végrehajtásokat (ciklusok, iterációk).

Bizonyos programozási szerkezetek és lehetőségek hiányoznak (mint pl. az elágazás, változók, ...), de gyakran előfordul, hogy valamilyen eszközhöz olyan programozási nyelvet fejlesztenek ki, amivel az könnyen programozható, de máshol az adott nyelv nem használható.

<http://www.klbg.com/lacing/>



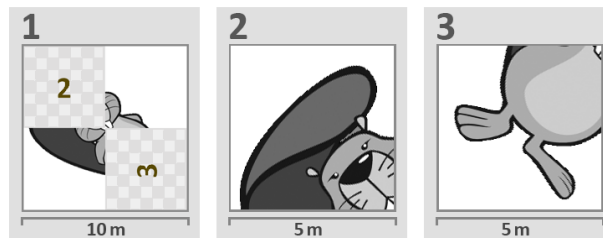
REKURZÍV FESTÉS (2016-AT-06)

SENIOR – NEHÉZ

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

A művésztől egy mondatnyi festési utasításkártyát kapnak a híres festőkártya-nyelvükön, utalásokkal a képelemekre, méretekre és forgatásokra. Némelyik festési utasításkártyán szerepelnek számozott mezők, amelyek további festési utasításkártyákra utalnak.

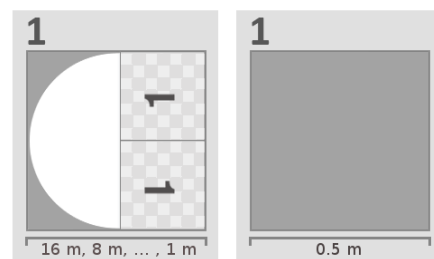
Itt szerepel egy példa egy korábbi festőkártya projektből. Ha az ember helyesen hajtja végre e három festési utasításkártyát, egy hód képe áll elő:



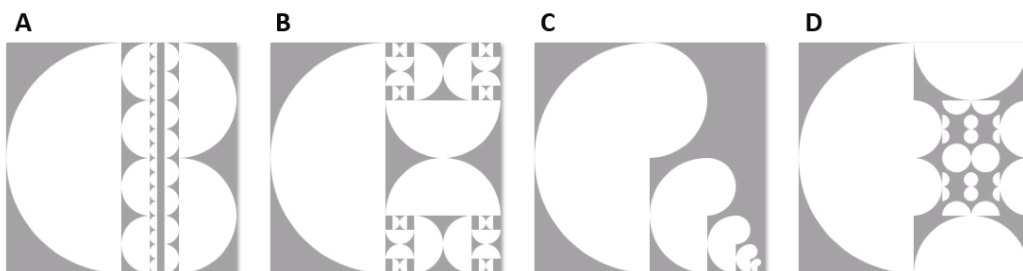
A különleges kiállításra Tina és Ben a következő festési utasításkártyákat kapják:

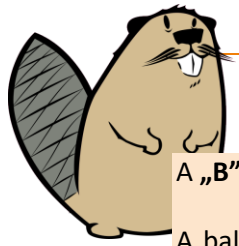
Ben ráncolja a homlokát. „Ez meg mit jelentsen? A bal oldali kártya önmagára utal vissza, ezen kívül mindkét kártyának ugyanaz a száma!”

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



Hogyan fog kinézni a kiállítóterem padlója?





A „B” válasz a helyes.

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

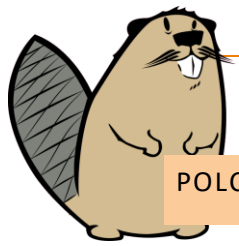
Az egyik képelemnél az egyes 90° -kal balra van forgatva. Ezért a képelemet szintén balra kell forgatni, a félkör görbülete befelé mutat. A másik képelemnél az egyes 90° -kal jobbra van forgatva. Ezért a képelemet szintén jobbra kell forgatni, a félkörfelület görbülete befelé mutat.

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

MIÉRT INFORMATIKA?

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

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



POLCRENDEZÉS (2016-AU-03)

BENJÁMIN – KÖNNYŰ

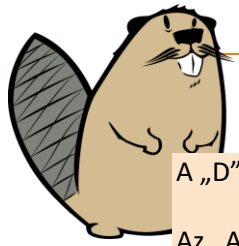
Beatrix átrendezi a polcát.

Két szabályt vesz figyelembe:

1. Négyzetes tárgyak NE kerüljenek egymás mellé.
2. Gömbölyű tárgyak NE legyenek négyzetesek mellett.

Jelöld meg azt a polcot, amelynél Beatrix betartotta a szabályokat!

<p>A</p>	<p>B</p>
<p>C</p>	<p>D</p>



A „D” válasz a helyes.

Az „A” polc azért nem helyesen rendezett, mert a bal oldali végén négyszögletes tárgyak állnak egymás mellett.

A „B” polc azért nem helyesen rendezett, mert a labda négyszögletes tárgy mellett helyezkedik el.

A „C” polc sem lehet helyesen rendezett, mert a labda négyszögletes tárgy mellett van.

Csak a „D” polcon érvényesül mindkét szabály.

MIÉRT INFORMATIKA?

Az adatoknak többféle típusa létezik. Például képek, szövegek, számok. Ha alaposan szemügyre vesszük az adatokat, akkor felfedezhetünk bizonyos tulajdonságokat és szabályszerűségeket. Ezáltal szabályokat állíthatunk fel, megjósolhatunk valamit vagy megoldhatunk egy problémát. A szabályok felismerése alapozza meg a számítástechnikai programozást.



STRANDZÁSZLÓ NYELV (2016-AU-04)

KISHÓD – KÖZEPES, BENJÁMIN – KÖNNYŰ

Albert vízi mentő egy strandon. Gyermekei, Beatrix és Richárd tőle távol játszanak.

A tenger zúg, a szél fúj, így a kiabálás nem hallatszik és nincs térerő a mobiltelefonokhoz. Hogy mégis tudjon a gyermekeinek szólni, Albert 3 állású zászlórudat és különféle zászlókat alkalmaz.

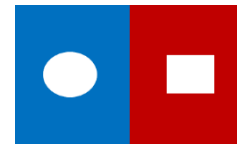
A **felső zászló** a következőket jelentheti:



Richárdnak szól az üzenet.



Beatrixnak szól az üzenet.



Richárdnak és Beatrixnak is szól az üzenet.

A **középső zászló** a következőket jelentheti



Étel érkezett.



Ital érkezett.



Ráérsz.



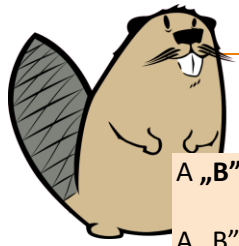
Siess!

A **legalsó zászló** a következőket jelentheti

Mit jelent, ha a következő zászlók lengenek a rúdon?

- A) Beatrix, étel érkezett, siess!
- B) Richárd, étel érkezett, siess!
- C) Richárd és Beatrix, étel érkezett, ráértetek!
- D) Beatrix, ital érkezett, ráérsz!





A „B” válasz a helyes.

A „B” válasz azért helyes, mert a 3 zászló jelentése felülről lefelé haladva:

Kék zászló, tehát az üzenet Richárdnak szól.

Zöld zászló, tehát étel érkezett.

Piros zászló, tehát sietni kell.

MIÉRT INFORMATIKA?

Ebben a feladatban több részből álló üzenetek kerültek kódolásra zászlók segítségével. Ehhez a zászlók színei, formája és helyzete határozta meg a jelentést. A hírek ebben a helyzetben a legmegfelelőbb (optimális) csatornán kerülnek elküldésre, és csak egy irányba küldhetőek – a gyerekeknek nincsenek zászlóik. Ha üzeni szeretnének, oda kell szaladniuk és kiabálniuk.

Az informatikusok, ha kódolásról van szó, legtöbbször az üzenő csatornával foglalkoznak, mely alapulhat akár elektromágnesességen, lézerefényen vagy digitális elektronikán is. Az internet közös nyelve mindenhol elérhető, hogy minden küldő és fogadó megérthesse az üzenetek jelentését.

Természetesen vannak titkosított kommunikációk. El tudjuk képzelni, mi lenne, ha mindenki ismerné a zöld zászló jelentését és jönne elfogyasztani az ételt, ami érkezett és amit Albert csak a gyermekeinek szánt. Jobb, ha családi titok marad a zászló jelentése.



ALAKZATJÁTÉK (2016-CA-09)

SENIOR – NEHÉZ

Aliz geometriai alakzatokkal játszik. Egy-egy alakzatsort átvált alakzatok egy másik sorára.

Aliz minden játékkörben saját átváltási szabályokat rögzít. Minden átváltáskor a szabályt olyan sokszor alkalmazza, ahányszor csak lehetséges. Minden játékkört egy alakzattal kezd.

Az egyik korábbi körben Aliz letett egy négyzetet és a következő váltási szabályokat alkalmazta:



Három lépésben a jobb oldali alakzatsort hozta létre:

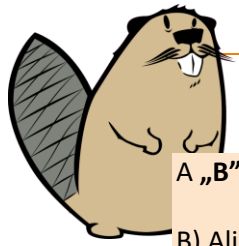


Egy másik körben Aliz a következő alakzatsort hozta létre:



Az alábbi szabályrendszerek közül melyik lehet érvényben, ha a fenti kártyasorozat jött ki és tudjuk, hogy a kiindulópont a három geometriai alakzat egyike volt?

A	B	C	D



A „B” válasz a helyes.

B) Aliz háromszöggel kezdett és három lépésben jutott el az alakzatsorhoz:



A többi válaszlehetőség a következő átváltások megfontolásával kizárható:

A) Háromszöggel vagy körrel kezdve soha nem kaphat négyzetet. Négyzettel kezdve a következő történik:



Mivel minden egyes átváltás növeli a sorhosszt, nincs lehetőség a kívánt eredmény elérésére.

C) Háromszöggel kezdve Aliz nem kaphat sem négyzetet, sem kört. Négyzettel kezdve a következő történik:

A sorkezdeten álló dupla háromszöget már soha többé nem lehet háromszög-négyzet párossá átalakítani, pedig ez lenne szükséges a cél eléréséhez. Körrel kezdve is előjön a kezdő dupla háromszög feloldhatatlan problémája:

D) Aliznak semmiképpen sem adódik lehetősége két kört egymás mellé helyezni, pedig ez lenne a kívánatos cél.

MIÉRT INFORMATIKA?

Az alakzatjáték szimbólumai és szabályai együtt a szimbólumláncok egy átváltásrendszerét alkotják. Az elméleti informatikában nagy jelentőséggel bírnak az ilyen átváltási rendszerek. Az átváltásrendszereket nyelvtannak is szokás nevezni. A szimbólumsorozatok, amint itt az alakzatjátékban is, előállíthatók a szabályok alkalmazásával, és így egy nyelv szavai lesznek. Az elméleti informatikában egy nyelv egyben szavak tömege is, a nyelvtan pedig előállít egy ilyen formális nyelvet.

Minden nyelvtan teret ad a szóprobléma megfogalmazódásának: egy adott szó az adott nyelvtannal jellemezhető nyelvhez tartozik vagy sem? A szóprobléma megoldásának nehézségi fokát döntően befolyásolja, hogy a nyelvtan szabályai milyen formát öltöttek. A szabályok formájától függően a nyelveket különféle tulajdonságokkal rendelkező osztályokba sorolják. Például:

- A rendezett nyelvtanok olyan nyelveket hoznak létre, amelyek szabályai előírásokká fogalmazhatók és végső soron automatizálhatók. Az ilyen rendezett nyelvtanok, ill. kifejezések rendkívül hasznosak pl. keresési algoritmusok felállításánál.



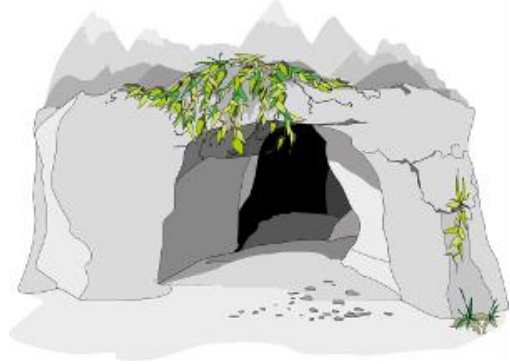
- A kontextusfüggetlen nyelvtanok olyan nyelveket hoznak létre, amelyekben a szóprobléma hatékonyan megoldható. A programnyelvek nyelvtana éppen ezért rendszerint kontextusfüggetlen. Az összeállítók által elfogadott program érvényesül formális szempontból a nyelv szavainál.
- A korlátmentes nyelvtanok olyan nyelveket hoznak létre, amelyekben a szóprobléma csak részlegesen oldható meg. Egy adott szó korlátmentes nyelvtannal jellemezhető nyelvhez tartozását alkalmasan tervezett Turing-géppel bizonyíthatjuk. Ugyanaz a Turing-gép nem lesz alkalmas az adott nyelven kívüli szóra.



ÁT A SÖTÉT BARLANGON (2016-CH-04A)

JUNIOR – KÖZEPES

Anna és Beni a szüleikkel túrázni mennek. Útközben egy barlangon kell keresztülmászniuk. Tapasztalatból tudják, hogy ez mindannyiuknak különböző időbe telik: Annának 10 perc, Beninek 5 perc, az anyukájuknak 20 perc és az apukájuknak 25 perc.



A keskeny és sötét barlangon biztonságosan csak egy vagy két hód mászhat át egy időben. Mivel csak egy zseblámpájuk van, több fordulót kell tenniük. Amikor ketten mennek, mindig annyi időre van szükségük, amennyire a lassabbnak.

Az egész család szeretne a barlang másik oldalán lenni egy órán belül, mielőtt az elemlámpa lemerül. Anna úgy gondolja, sikerülhet, méghozzá 5 átmenettel!

*Kik **NEM** mehetnek együtt, ha azt akarják, hogy Anna terve sikerüljön?*

- A) Beni és apukája.
- B) Apuka és anyuka.
- C) Anna és Beni
- D) Mindegy hogy mennek, 1 óra alatt akkor sem sikerül.



Az „A” válasz a helyes.

Anna megoldotta az alagútproblémát egy frappáns ötlettel! Ez az ötlet meg kell feleljen bizonyos feltételeknek (az átjutóknál mindig kell legyen egy lámpa), és ennek a tervnek a végrehajtásához figyelembe kell venni a korlátozó tényezőket (lámpa működési ideje és az idő).

Csak úgy lehetséges, hogy mindannyian 60 perc (1 óra) alatt átérjenek, ha az anya és az apa (a két leglassabb) csak egyszer – és egyszerre – vannak a barlangban. Ez csak akkor megoldható, ha korábban Anna és Beni már átment és az egyikük hozza vissza a zseblámpát. Az, hogy melyikük hozza vissza a szülők előtt és melyikük utánuk a lámpát, az idő szempontjából mindegy (lásd a táblázatokat).

<i>út</i>	<i>hód</i>	<i>hód</i>	<i>perc</i>	<i>út</i>	<i>hód</i>	<i>hód</i>	<i>perc</i>
1. oda	Anna	Beni	10	1. oda	Anna	Beni	10
2. vissza	Beni		15	2. vissza	Anna		20
3. oda	Hódpapa	Hódmama	40	3. oda	Hódpapa	Hódmama	45
4. vissza	Anna		50	4. vissza	Beni		50
5. oda	Anna	Beni	60	5. oda	Anna	Beni	60

MIÉRT INFORMATIKA?

Az informatikában egyre összetettebb rendszereket fejlesztenek, melyek saját működésüket maguktól (autonóm módon) meg kell, hogy tervezzék. Mindezt adott feltételek mellett, rendelkezésre álló eszközök segítségével. Ezek a rendszerek jóval szélesebb körűek, mint ez a feladat. Például az önjáró autó: meg kell terveznie az útvonalat, hogy célba érjen, szem előtt tartva, hogy mindig legyen elég energiája.

A független és biztos tervezésnek etikailag és jogilag is elfogadhatónak kell lennie, ha az autonóm rendszer működése emberekre és más élőlényekre is kihat.

Hasonló elgondolások vezették az író, Isaac Asimovot 1942-ben, az első számítógépek készítésekor. Ő vezette be, többek között ezt a törvényt: „A robotnak nem szabad kárt okoznia emberi lényben vagy tétlenül túrnie, hogy emberi lény bármilyen kárt szenvedjen.” Mivel az önjáró autó is egy robot, szintén be kell tartania az előbb említett szabályt. Az önvezető autó navigációját egyébként számos érzékelő és modern navigációs eszköz, így például radar, lézerradar, GPS segítségével oldják meg.








SZÜLINAPI GYERTYÁK (2016-CH-05)

BENJÁMIN – KÖZEPES, KADÉT – KÖNNYŰ





Ma van Bodos 11. születésnapja. Bodos Anyuka viszont csak öt gyertyát talált. Szerencsére tudja, miként lehet az 5 gyertyával a 11-et ábrázolni. Fel is állította szépen a gyertyákat a torta tetejére, sorban egymás mellé:

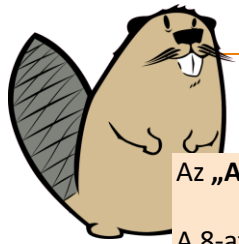
- A jobboldali gyertya egyet ér!
- Minden másik gyertya a tőle jobbra levő gyertya dupláját éri!
- Csak az égő gyertyákat számoljuk!

Például:

1	2	4	$2+1=3$	$16+4+1=21$
				

Mely gyertyák égnek Bodos tortáján a 11. születésnapján?

A	B	C	D
			



Az „A” válasz a helyes.

A 8-at, 2-t és 1-et érő gyertyák égnek: $0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11$.

B) Mivel a 8-at, 4-et és 2-t érők égnek: $0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 14$.

C) Mivel csak a 16-ot érő ég: $1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = 16$.

D) Mivel a 16-ot, 8-at és 2-t érő ég: $1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 26$.

MIÉRT INFORMATIKA?

Az „ég a gyertya” vagy „nem ég a gyertya” fejezi ki, hogy egy számérték hozzáadódik-e vagy sem az összeghez. A gyertya helyzete határozza meg a számértéket.

Mindez akár egyesekkel („ég a gyertya”) vagy nullákkal („nem ég a gyertya”) is megvalósítható. Bodos szülinapi gyertyái tehát így is leírhatóak: 01011. Ez a 11-es szám ábrázolása bináris (kettes) számrendszerben.















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



VIRÁGOK ÉS NAPOK (2016-CH-12)

BENJÁMIN – NEHÉZ, KADÉT – KÖNNYŰ

Barbara két pecsétet kapott. Az egyik virágot, a másik napot nyomtat. Elgondolta, hogy tudná a nevét csak a virágokkal és napokkal lepecsételni. A különböző betűket virágok és napok különböző sorozatával határozta meg.

Betű	B	A	R	E	Y
Sorozat		 	  	   	   

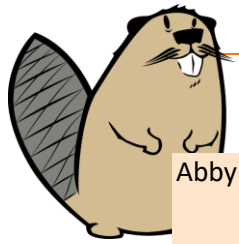
A saját nevét, „Barbara” tehát így pecsételte le:



Ezután lepecsételte angol barátja nevét is:



Hogy hívják a barátját?



Abby a helyes válasz. Barbara így pecsételte volna le a többi nevet:

Arya:

Barry:

Ray:

MIÉRT INFORMATIKA?

Amikor adatokat, mint pl. az ábécé betűit más adatokkal (mint itt a virágok és napok sorozatával) írunk le, az informatikában kódolásról, kódokról beszélünk.

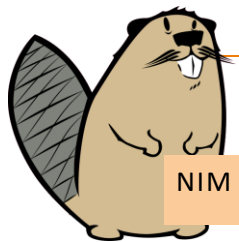
Adatok, melyek a számítógép segítségével tárolásra és feldolgozásra kerülnek, binárisan kell, hogy kódolva legyenek, azaz 0-k és 1-es értékek sorozataként.

Egy ilyen bináris kódot egyszerű létrehozni, ha minden adat-elem, pl. minden betű, ugyanannyi biten kerül tárolásra. Sokáig szokás volt a szöveg-elemeket 8 biten kódolni – a 8 bit sorozatát byte-nak is nevezzük. A most használt kódok, mint pl. az utf-8 több bitet is felhasznál egy-egy elem kódolására, hiszen az egész világ összes nyelvének elemeit kell leírnia.

Ha túl sok adatunk van, érdekesebb helyspórolással kódolni. Az egyes nyelvi jelek (betűk) nem ugyanolyan gyakorisággal fordulnak elő a szövegekben. Az „e” vagy az „n” például sokkal gyakoribb, mint az „x”. Ha a gyakoribb jeleket kevesebb biten tároljuk, mint a ritkébbakat, akkor változó hosszú kódolásról beszélünk. Ahhoz, hogy egyértelműen felismerhetőek legyenek az egyes jelek, egy jel kódja sosem egyezhet meg egy másik jel kódjának az elejével. Az ilye szabályokat megvalósító kódokat prefix kódnak is hívjuk. Talán a legismertebb a morze kód.

<https://hu.wikipedia.org/wiki/Huffman-k%C3%B3dolás>

<https://hu.wikipedia.org/wiki/Morzek%C3%B3d>



NIM (2016-CH-23)

JUNIOR – KÖNNYŰ, SENIOR – KÖZEPES

Bíbor Nim játékot játszik az egyik barátjával. 13 gyufa van az asztalon. A játékosok felváltva vesznek el tetszőlegesen 1, 2 vagy 3 gyufát. Aki az utolsó gyufát elveszi, az nyer.



Tipp: Ha 4 gyufa maradt, amikor Bíbor következik, akkor ő már nem nyerhet. Ezt a helyzetet szeretné elkerülni.

Bíbor kezdi a játékot.

Hány gyufát kell Bíbornak felemelnie, hogy megnyerje a játékot?



1 a helyes válasz. Ekkor ugyanis 12 gyufa marad az asztalon, A barátja 1, 2 vagy 3 gyufát vehet el. Bíbor pedig ezután annyit, hogy 8 maradjon. A barátja ismét 1, 2 vagy 3 gyufát vesz el, Bíbor pedig annyit, hogy 4 maradjon. Innen a barátja már nem tud nyerni, hiszen 4 gyufát nem vehet el, de ha már egyet elvesz, Bíbor hárommal megnyeri a játékot.

A taktika lényege, hogy az ellenfélnek mindig négygel osztható gyufája maradjon.

MIÉRT INFORMATIKA?

A játékelméletben az olyan játékokat, mint a Nim vagy az ismert fogolydilemma mintaként használják, hogy a való stratégiai problémákat elemezzenek és megoldásokat találjanak.

A piacgazdálkodásban ez a megismerés elvezethet például ahhoz, hogy az árakat optimálisan határozzák meg. Egy árcsökkenés megnövelheti ugyan az eladást, de egyidejűleg csökkenti a nyereséget az egyes termékeken. Fordítva az áremelkedés növelheti az egyes termékekre eső nyereséget, de csökkentheti az eladást és ezzel az összes nyereséget is.

A játékelméleti modellekkel az eladók lehetséges reakciói (válaszai) megjósolhatóak az áringadozások tekintetében.

Az, hogy milyen jelentőségűek ezek a modellek a gazdaságban, az is mutatja, hogy több Nobel-díjat is kiosztottak játékelméleti munkákért.

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

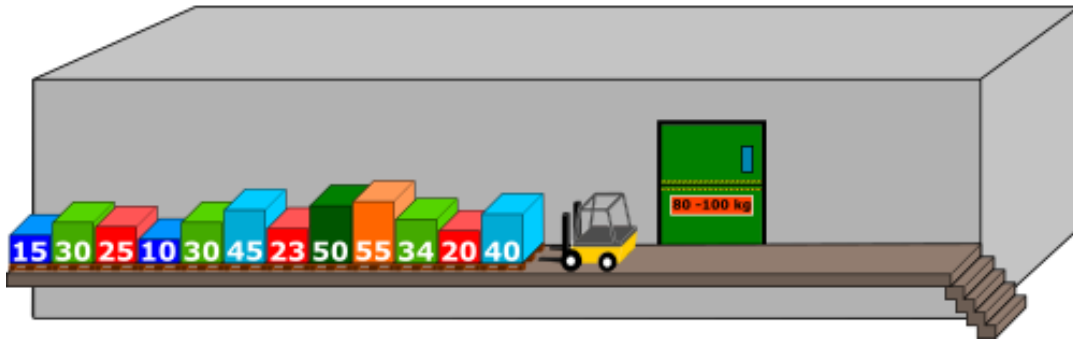
<https://hu.wikipedia.org/wiki/Fogolydilemma>



LÁDAPAKOLÁS (2016-CZ-02A)

JUNIOR – NEHÉZ, SENIOR – KÖZEPES

Egy rakodórámpan, a liftajtótól balra ládák sorakoznak. A ládák tömege (kg-ban, balról jobbra): 15, 30, 25, 10, 30, 45, 23, 50, 55, 34, 20, 40.

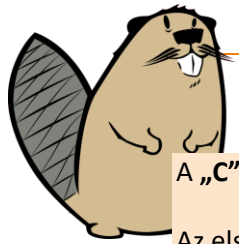


A ládákat egy villástargonccal berakodják a liftbe. A lift akkor indul el felfele, ha legalább 80kg és legfeljebb 100kg rakomány van benne. Ezután ismét üresen visszajön.

A rámpa olyan keskeny, hogy a ládákat nem lehet egymás mellett átemelni. Ezért mindig a sorban a legelső ládát kell feldolgozni. Ezt a ládát a liftbe rakodják, ha a rakomány súlya ezzel még nem lépi túl a 100 kg-ot. Máskülönben a ládát a rámpa jobb oldalára, egy új sorba állítják be. Amikor a bal oldalon minden láda feldolgozásra került, a jobb oldali sorban álló ládák ugyanezen a módon kerülnek feldolgozásra.

Minden ládát a fent leírt módon a lifttel a raktárba visznek. A következő állítások melyike igaz?

- A) A második útján a lift rakománya 98 kg-ot nyom.
- B) A ládák a rámpa jobb oldalán nem kerülnek beszállításra a raktárba.
- C) A lift egyik útján a rakománya 100 kg-ot nyom.
- D) A lift összesen ötször fordul.



A „C” válasz a helyes.

Az első úton a rakomány az első három ládából áll össze, és súlya $40\text{kg} + 20\text{kg} + 34\text{kg} = 94\text{kg}$.

A második út során először a legközelebbi ládát (55kg) pakolják be a liftbe. A következő ládát (50kg) ezután már nem rakodhatják be ($55\text{kg} + 50\text{kg} > 100\text{kg}$), és a rámpa túlsó oldalára tolják. A következő ládát (23kg) fölrakják az emelőre; azonban a rakomány még túl könnyű: $55\text{kg} + 23\text{kg} = 78\text{kg}$. A következő láda (45kg) nem rakható be a liftbe, és a rámpa jobb oldalára tolják. Ugyanez történik a következő ládával (30kg). Végül az ezután következő ládát (10kg) berakhatják a liftbe. A lift a második útját $55\text{kg} + 23\text{kg} + 10\text{kg} = 88\text{kg}$ rakománnyal teszi meg.

Az utolsó három ládát a sorból ($25\text{kg} + 30\text{kg} + 15\text{kg} = 70\text{kg}$) mind berakodják a liftbe. Most a rámpa jobb oldalán álló ládákat dolgozzák fel. Az első ládát ebből a sorból (30kg) felrakják az emelőre, amely pontosan 100kg -os terheléssel megteszi harmadik útját.

Ekkor már csak két láda van a rámpa jobb oldalán: $45\text{kg} + 50\text{kg} = 95\text{kg}$. A lift ezekkel a ládákkal teszi meg negyedik útját.

A rakomány a harmadik úton 100kg -ot nyom, tehát a C válasz állítása igaz. Az A, B, D és E válaszok állításai helytelenek, amint azt a magyarázat mutatja.

MIÉRT INFORMATIKA?

Az informatikában használnak olyan adatszerkezeteket, melyek megegyeznek egy alkalmazás modelljével. Ezek közül egy, melyet mindenképpen fontos megismerni, a verem (angolul stack).

Egy veremnél két esemény (művelet) megengedett:

- 1) Helyezz el egy elemet a verem tetejére.
- 2) Vegyél le egy elemet a verem tetejéről.

Ebben a feladatban két verem szerepel. Talán első pillantásra nem ismerhetőek fel, mert vízszintesen állnak. Az egyik a liftajtótól balra, melyben kezdetben benne van mindegyik láda. Legfölül a 40kg -os. A másik verem a liftajtó bal oldalán kezdetben üresen (és így láthatatlanul) szerepel, és a villástargonca tölti fel.

A villástargonca reprezentálja, valósítja meg az eseményeket (műveleteket), melyeket PUSH-nak és POP-nak is szoktak nevezni.

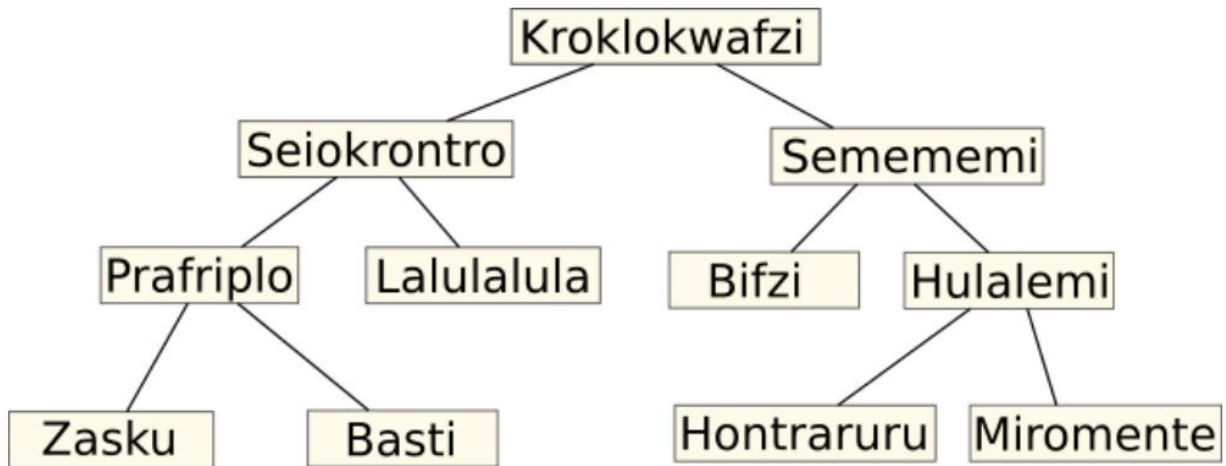
[https://hu.wikipedia.org/wiki/Verem_\(adatszerkezet\)](https://hu.wikipedia.org/wiki/Verem_(adatszerkezet))



HIERARCHIA (2016-CZ-03)

BENJÁMIN – NEHÉZ, KADÉT – KÖZEPES, JUNIOR – KÖNNYŰ

Az ábra a Morgenstern bolygó állatainak viszonyát írja le. Két állatcsoport közötti vonal azt jelenti, hogy az alsó típus minden állata egyben a felső állatcsoportba is beletartozik.



Például minden „Hulalemi“ egyben „Semememi“ is. A „Seiokronro“ egyedei között viszont vannak olyanok, amelyek nem tartoznak a „Basti“ családba.

A következő állítások közül csak egy igaz. Melyik?

- A) Minden Basti egyben Seiokronro is.
- B) Némelyik Hontraruru nem Semememi.
- C) Minden Zasku egyben Bifzi is.
- D) Minden Prafriplo egyben Basti is.



Az „A” válasz a helyes.

A: A Basti típus minden állata a Prafriplo típusnak is tagja. Minden Prafriplo egyben Seiokronro, következésképpen minden Basti a Seiokronro csoportnak is tagja.

B: A Hontraruruk a Hulalemik tagjai, amik pedig a Semememik közé tartoznak. Így minden Hontraruru egyben Semememi.

C: Mivel egy Zasku mindig Prafriplo, nem lehet ugyanakkor Basti is.

D: Minden Basti Prafriplo és nem fordítva.

MIÉRT INFORMATIKA?

A számítógépes programok információkat dolgoz(hat)nak fel a világ összes objektumáról, mint például emberekről, gépekről, bankjegyekről, biztosítási kötvényekről, online áruházakban kapható árukról és még sok másról. Az objektumok különböző fajtáit (informatikában osztályokról és nem fajtákról beszélünk) a programfejlesztők az osztályba tartozó összes objektum közös tulajdonságaival és funkcióival írják le.

Egy osztály objektumai egyidejűleg egy felsőbb osztályba is tartozhatnak: a joghurt ugyanúgy tejtermék, mint a vaj. A felsőbb osztály tulajdonságai az alatta lévő osztályokra is igaz lesz, öröklődik: a bolti árunyilvántartó tudja a tejtermékekről, hogy azokat hűteni kell, és így mindezt tudja a joghurtról és a vajról is. Mivel más termékek is igényelhetnek hűtést (pl. húsáru), ez a tulajdonság szerepelhet a hűtött áruk osztályánál, mely a tejtermékeknek és a húsárúnak is felsőbb osztálya lehet. Emiatt a húsáruk még korántsem tejtermékek és nem minden hűtött áru joghurt. Hasonlóan a Morgenstern bolygó állatfajtaíhoz.

A számítógépnek teljesen mindegy, hogy nevezzük el a programban leírt osztályokat. A programban a hűtött árukat akár Seiokronro-nak, a tejtermékeket Prafriplo-nak és a joghurtot Basti-nak, a vajat Zasku-nak is hívhatjuk, minden változatlanul működni fog.

Csak a Morgenstern-programot böngésző embereknek okozhat mindez nehézséget, de remélhetőleg mulatságot is.



ZÁSZLÓTÁROLÁS (2016-CZ-04)

SENIOR – NEHÉZ

A számítógépes képek képpontok soraiból állnak. Ha el akarjuk tárolni a képeket, a legegyszerűbb esetben minden ilyen képpont színét írjuk le.

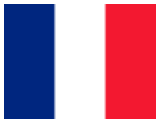


A GIW (kitalált) formátummal a képeket össze tudjuk tömöríteni, azaz a kisebb helyen tudjuk tárolni. Ez (a tömörítés) így működik:

- Minden képpont-sor egyenként kerül leírásra.
- Minden színt három betűs rövidítéssel jelölünk.
- Azonos színű képpontok sorozatát egy zárójelbe írjuk, ami a szín rövidítését és az azonos színű képpontok számát tartalmazza.

Például a (zöl, 20)(feh,13) módon leírt képpont-sorban először 20 zöld, majd 13 fehér képpont szerepel.

Lejjebb négy képen zászlókat láthatsz. Mindegyik zászló képen négy-négy szín található és mindegyik zászló ugyanannyi képpontból áll. GIW formátumban szeretnék tárolni ezeket.

Melyik zászló tárolásakor lesz a legnagyobb a GIW állomány?

			
Csehország	Franciaország	Örményország	Svédország



Franciaország a helyes válasz.

Az olyan sorok, ahol csak egy szín szerepel, egy zárójel-kifejezéssel felírhatóak. Az örmény zászló esetében minden sorban csak 1-1 szín szerepel. A GIW állomány tehát minden sorra csak 1 zárójel-kifejezést tartalmaz.

Azon a zászlón, ahol egy sorban színváltás történik, minden színváltás 1 újabb zárójel-kifejezést jelent. 2 szín esetén két zárójeles rész kell majd soronként. A cseh zászló esetében pontosan egy színváltás szerepel soronként (kékből fehér, illetve kékből piros). A GIW állomány így minden sorra 2 zárójel-kifejezést jelent.

A svéd zászlónál a legtöbb sorban két színváltás szerepel, ami a GIW állományban ezekre a sorokra 3 zárójel-kifejezést jelent. A zászló (kép) közepén van pár sor, ahol csak egy szín szerepel, így ott minden sorra csak egy zárójel-kifejezést kell tárolnunk.

De több színváltásos sorunk van, mint egyszínű.

A francia zászlónál minden sorban két színváltás történik. A GIW állományban ez soronként tehát 3 zárójel-kifejezést jelent.

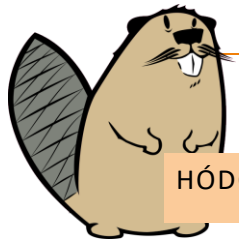
MIÉRT INFORMATIKA?

Az adattömörítés az informatika fontos területe. Minél kisebb az az adatmennyiség, amit digitális adatokkal leírunk, annál kevesebb helyre van szükségünk a tárolásukhoz és annál gyorsabban tudjuk továbbítani (pl. egy hálózaton keresztül).

A tömörítési eljárások a továbbítási költségeket jelentősen csökkenthetik. Ha egy rádióadásban a zenét tömörítés nélkül továbbítanánk, akár tízszer akkora adatmennyiséget kellene továbbítani, mint egy általánosan használt tömörítést használva. Mivel a továbbítandó adatok tömege egyre nagyobb lesz (pl. az egyes adásokhoz különböző nyelveken elküldött szövegek), az informatikusok tovább kutatnak az adattömörítés területén.

A feladatban szereplő tömörítés a futáshossz kódolás (run length encoding) alapján működik.

<https://hu.wikipedia.org/wiki/Adatt%C3%B6m%C3%B6r%C3%ADt%C3%A9s>



HÓDCHAT (2016-DE-02)

BENJÁMIN – NEHÉZ, KADÉT – KÖZEPES

A Hódchat egy ingyenes alkalmazás, mely reklámokból tartja fenn magát. A Napsütés Utazási Iroda igyekszik a célcsoportnak megfelelő reklámot választani. Minden üzenetéről automatikus értékelést készítenek, melyekben meghatározott szavakra keresnek, és ezekhez pontszámokat rendelnek.

- Például valamely megszokott üdvözlés az ifjú hódok között (Hello, Hi, Cs) **-2 pontot** ér.
- Az idősebbek higgadt megszólításai (Kedves, Tisztelt) ezzel szemben **2 pontot** jelentenek.
- Minden rövidítés (lol, yolo, pill, tom) **-1 pont**.
- Minden 10 betűs vagy annál hosszabb szó **1 pont**.

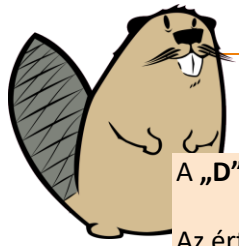
A hódok üzeneteik összpontszáma alapján besorolódnak a célcsoportok egyikébe:

<i>Pontszám</i>	<i>Célcsoport</i>	<i>Mutatott kép</i>
Pozitív	Senior	Strandkép
Negatív	Fiatal	Szörfös kép
0	Nincs besorolás	párizsi Eiffel torony



Az alábbi üzenetek közül melyikhez fogja az értékelő a párizsi Eiffel tornyot hozzárendelni?

- A) Sziasztok Kedves Barkácstársak, felteszem a mai kérdésemet, használnátok-e horganyzott csavarokat egy víz alatti projektben? Ricsi
- B) Nem tom, bocsi. Rozsdálnak a cuccok?
- C) Cs, vki?
- D) @Liza: <3 <3 <3



A „D” válasz a helyes.

Az értékelésnél semmilyen támpontot nem találunk, így nem tudunk pontot adni.

Az „A” üzenetben teljes megszólítás és hosszabb szavak is szerepelnek. Ezzel mindenképpen pozitív értékelést ér el. Így a strandkép jelenik majd meg.

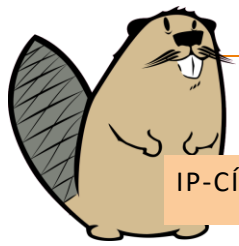
A „B” és a „C” üzenetben is rövidítések és „fiatalos” megszólítások szerepelnek, ezért több mínusz ponttal végeznek. Így a szörfös kép jelenik majd meg.

MIÉRT INFORMATIKA?

Egy szöveg meghatározott szabályok alapján történő kiértékelése könnyen elvégezhető számítógépes programokkal. Szövegrésztetek keresése egyszerű példa a minta-egyezéses keresésre („pattern matcing” angolul). Ez nem csak a szövegfeldolgozásnál fontos, de pl. a képfeldolgozás területén is számos számítógépes alkalmazás használja.

Társadalmi fontosság: Az internetes felhasználói profilok manapság számtalan automatizált vizsgálatban kerülnek kiértékelésre, például azért, hogy a „megfelelő” a vásárlói ajánlatokkal kereshessék meg a felhasználókat. Ezért is egyre fontosabb, hogy az internethasználók tudatosak legyenek és a személyes adataikat figyelmesen kezeljék, adják ki.

Mindannyian döntéshelyzetben vagyunk: amennyiben kiszolgáltatók magunkról adatokat, azt fel tudják használni előrejelzésekhez, segíthetik a megelőzést több területen is, mint pl. az egészségügy, bűnüldözés, Ha vigyázunk az adatainkra, akkor viszont csökkentjük annak kockázatát, hogy visszaélhessenek vele, kárt okozhassanak nekünk vagy akár hozzátartozóinknak.



IP-CÍMEK (2016-DE-03)

JUNIOR – NEHÉZ, SENIOR – KÖZEPES

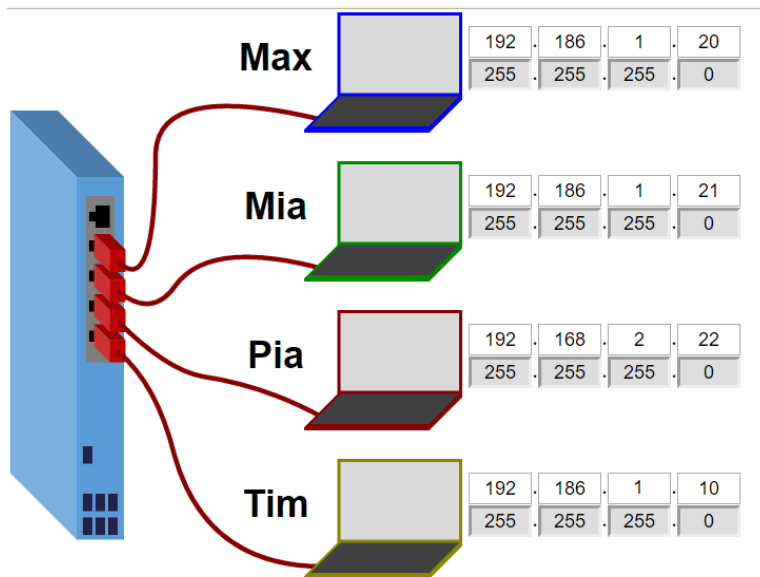
Egy hálózatban minden számítógépnek egyedi IP-címe van. Ez négy részből áll, ahol mindegyik rész egy 1 és 254 közötti szám. Például: 192.168.1.25

A hálózati maszk is négy részből áll, amiben legtöbbször a 255 vagy a 0 szerepel. Például: 255.255.0.0

A hálózati maszk választja szét az IP-címeket a hálózat és az eszköz (pl. számítógép) azonosítójára. A 255 azt jelenti, hogy az a rész a hálózathoz tartozik. A 0 jelentése, hogy az a rész a számítógép azonosításához tartozik.

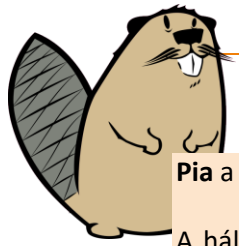
Ahhoz, hogy a számítógépek egy hálózaton belül egymással kommunikálni tudjanak, csak a számítógép-azonosító részben térhetnek el egymástól. A hálózati résznek minden esetben azonosnak kell lennie.

Például a 255.255.0.0 hálózati maszkkal a 172.16.0.10 és a 172.16.0.11 IP-című gépek egymással tudnak kommunikálni. De a 172.31.0.12 IP-című gép az előbbiektől egyikkel sem tud kommunikálni.



Max, Mia, Pia és Tim egy többjátékos játékot szeretnének elkezdni. Hálózati kábelek segítségével összekötik a számítógépüket egy úgynevezett adatátviteli kapcsolóval (switch-csel). Hálózati maszkként a 255.255.255.0-t állítják be. De amikor az IP-címüket is beállítják, csak hármójuk gépe látja egymást.

Kinek az IP-címén kell változtatni, hogy mindannyian tudjanak játszani?



Pia a helyes válasz.

A hálózati maszk (255.255.255.0) megadja, hogy az első három résznek az IP-címben ugyanannak kell lennie. A negyedik részben különbözhetnek.

De mindegyik részben 1 és 254 közötti számnak kell szerepelnie.

Pia IP-címének 2. részében a 168 szerepel, míg a többiekénél 186.

MIÉRT INFORMATIKA?

Az IP (Internet protokoll)-t először 1974-ben írták le. Ennek segítségével a számítógépek rugalmasan és váratlan helyzetekben különböző utakon is adatokat tudnak cserélni egymással.

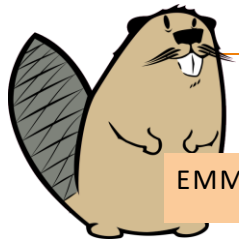
A TCP-vel (Transmission Control Protocol, magyarul: Adatátviteli protokoll vagy szabályrendszer) együtt (közösén) az IP az internet alapját képezi. Ahhoz, hogy egy gép egy másikkal az interneten keresztül kommunikálni tudjon, az operációs rendszerének tudnia kell a TCP/IP-t kezelni és rendelkeznie kell egy IP-címmel.

Az itt felvázolt rendszer az internet protokoll jelenleg leginkább használt változatát, az IPv4-et mutatta meg. Ezek az IPv4-címek 4 byte-ból állnak. Minden byte-on 254 különböző érték szerepelhet. Ezért maximum (legfeljebb) $254 \cdot 254 \cdot 254 \cdot 254 = 4.162.314.256$, azaz kb. 4 milliárd különböző IPv4-cím létezhet. De már egy ideje jóval több gép van az internetre kötve.

Milyen előnyös, hogy a gépek egy helyi (lokális) hálózaton (pl. iskola, otthon) egy úgynevezett router mögé elbújhatnak. Így csak a router-nek kell egy a világon egyedülálló címmel rendelkeznie, a többi gép címének elég a belső hálózaton belül egyedinek lennie.

Ennek ellenére kevés IPv4-cím létezik.

Az internet protokoll újabb generációja az IPv6, ami már sokkal több IP-címet tesz lehetővé. Jelenleg mindkét változat használható, de meglehetősen hamar az IPv4 el fog tűnni.

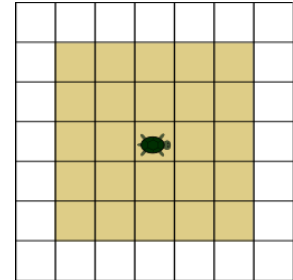


EMMA, A TEKNŐSBÉKA (2016-DE-08B)

KADÉT – NEHÉZ, SENIOR – KÖNNYŰ

Emma teknős Rácsországban él egy ötször öt cellás mezőn. Szívesen fogyaszt friss salátaleveleket.

A mezőn minden nap új salátalevelek nőnek. Emma nem tudja, melyik cellákban, de mindet meg akarja enni! Minden nap a mező közepéről indulva az egész mezőt be szeretné járni.



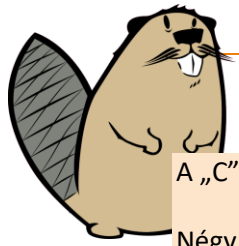
Ehhez készített egy tervet, amiben a következő utasításokat használta:

- fordulj jobbra: ilyenkor jobbra fordul 90 fokot;
- fordulj balra: ilyenkor balra fordul 90 fokot;
- R cellát menj előre: ilyenkor annyi cellát megy előre, ahányadszor végrehajtja a feladatokat.

Emma elhagyhatja a mezőt, de Rácsországot nem!

Melyikkel tervvel járja be az egész mezőt?

<p>Csináld meg ötször</p>	<p>Csináld meg ötször</p>	Mindkettővel	Egyikkel sem
A	B	C	D



A „C” válasz a helyes.

Négy különböző lehetősége van Emmának, hogy a mező összes celláját bejárja ezekkel az utasításokkal és az ötszörös ismétlés lehetőségével. Mind a négy esetben spirál-alakban fogja ezt megtenni.

Ezek közül kettő szerepel a megoldásokban, a másik kettő pedig ezek párja az ellentétes (balra-jobbra) fordulásokkal.

Természetesen ha nem ragaszkodunk az öt ismétléshez és ahhoz, hogy ne lépünk kétszer ugyanarra a cellára (hiszen a feladat ezt nem írta elő), több lehetséges megoldást is találhatunk.

MIÉRT INFORMATIKA?

Ennél a hód-feladatnál Emmának egy programot kell készítenie, melyekben szereplő utasításokat sorban kellett elvégeznie (még hozzá ötször).

Amennyiben a program helyes, Emma pontosan azt tette, amit eltervezett. Amennyiben a program nem helyes, Emma akkor is végre tudja hajtani az utasításokat, csak az eredmény lesz más.

Ugyanez a helyzet a számítógépekkel, a számítógépes programokkal. Egy számítógép nincs abban a helyzetben, hogy felismerje, hogy egy adott program helyes-e.

Ebben a feladatban egy olyan utasításblokkal oldottuk meg a feladatot, melyben négy utasítás szerepel.

Mind a négy utasítást pontosan ötször kellett végrehajtani. Az ilyen ismételt végrehajtásokat az informatikában ciklusnak hívjuk.

A feladatban azt is kihasználtuk, hogy egy ciklusnak van egy úgynevezett ciklusváltozója, amely megmondja, hányadszor hajtjuk végre az utasításokat - és minden végrehajtás után eggyel nő.

Általában a programnyelvekben nem csak ismétléseket, ciklusokat, de más vezérlést segítő szerkezeteket is használhatunk, mint például egy feltételhez kötött végrehajtás vagy egy másik programrész meghívása.



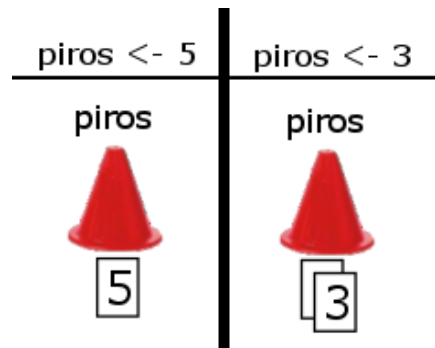
KÁRTYÁK ÉS BÓJÁK (2016-FR-02)

JUNIOR – KÖZEPES



A hódok kártyákkal játszanak, melyekre számokat írnak. A kártyákat színes bóják alá (halomba) rakják. Ha felemelnek egy bóját, csak a legfelső kártya látható.

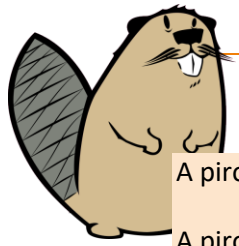
Azt, hogy mit csinálnak, a következőképpen írják le:



Most rajtad a sor. Amikor a hódok ezt írják le:

```
piros <- 3  
sárga <- 5  
piros <- 6  
sárga <- 8  
kék <- 1  
sárga <- 3
```

Milyen számok láthatóak a bóják alatt?



A piros alatt 6, a kék alatt 1, a sárga alatt pedig 3.

A piros bója alá egy 3-as, majd erre rá egy 6-os kártya kerül.

A kék bója alá csak egy kártya kerül az 1-es számmal.

A sárga bója alá először egy 5-ös, majd arra egy 8-as, legfölülre végül pedig egy 3-as kártya kerül.

Csak a legfelső kártyák láthatóak.

MIÉRT INFORMATIKA?

Ez a feladat jó példa arra, hogyan működnek a programok.

Gondoljunk úgy a bójákra, mint változókra. A változók értékeket tárolnak (ebben az esetben egész számokat).

Amikor egy programot végrehajtunk, a változóknak (bóják alatt) tárolt számokat kicseréljük, azok megváltoznak.

Ezek a bóják úgy működnek, mint egy verem: a halom tetején lévő értékek folyamatosan változnak, és amikor megnézzük mi a látható kártya, az a legutoljára oda helyezett. Ez azt jelenti, hogy a legutoljára lehelyezett kártya az első, amit kivehetünk, kiolvashatunk.

Sőt az értékek akár másolhatóak is lehetnek. Ezt a programozásban érték szerinti paraméterátadásnak hívjuk.

Több programozási nyelv a cím szerinti paraméterátadást (is) támogatja, amikor nem történik másolás, hanem csak egy hivatkozást teszünk a bója alá, ami megmondja, melyik másik bója alatti értéket kell figyelembe vennünk.

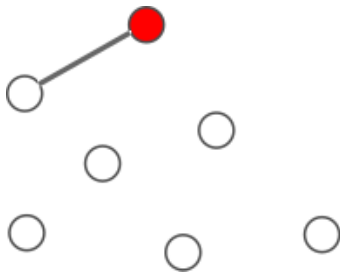


HIDTERVEZÉS (2016-FR-03)

KADÉT – KÖZEPES, JUNIOR – KÖNNYŰ

Hódpapa vízszonyos lett. A várából minden családtag hódvárába hídon szeretne átmenni. A hódok támogatják Hódpapát és a következőket tartják szem előtt a hidak tervezésénél:

- Hódpapa a várából legfeljebb (maximum) két hídon keresztül jusson el mindenkihez;
- minden elért hódvárból legfeljebb (maximum) két másik hódvárba lehet továbbmenni (anélkül, hogy visszatérnének az előző várba).



A hódok nekiállnak a tervezésnek. Minden várat egy körrel jelölnek. Hódpapa vára a piros kör. Az első hidat be is rajzolják Hódpapa várából. De aztán már nem tudják, hogyan tovább.

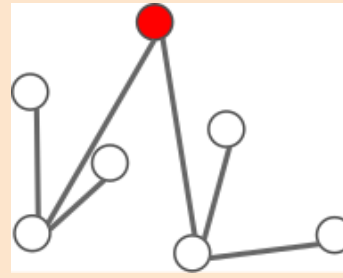
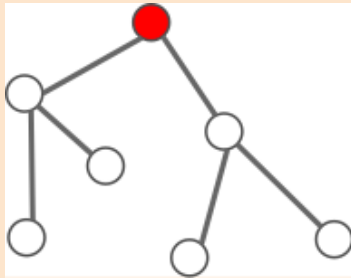
Egészítsd ki a tervet úgy, hogy minden elvárásuknak megfeleljen. Ezt többféleképpen is megteheted.

Legkevesebb hány hidat kell MÉG építeniük?



5 a helyes válasz.

Két lehetséges megoldást láthatsz az ábrákon.



De ezeken kívül még jó pár megoldást találhatsz. Az egyértelmű megoldáshoz a hódoknak további megszorításokat kell tenniük.

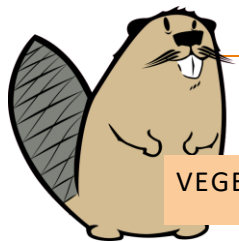
Kevesebb híddal nem oldható meg a feladat, mivel csak 2 híd vezethet tovább minden várból.

Érdekes kérdés, hogy ennél több várat el tudna-e érni Hódpapa...

MIÉRT INFORMATIKA?

A várak és a hidak együtt egy struktúrát (szerkezetet) alkotnak, melyben az egyik csomópontból (Hódpapa vára) az összes többi csomópont (többi vár) pontosan egyféleképpen érhető el éleken (hidak) és más csomópontokon (várak) keresztül.

Az ilyen szerkezeteket (struktúrákat) az informatikában Fának nevezzük. A kiemelt csomópont (mint pl. Hódpapa vára) a gyökér. A fa gyökerétől valamelyik csomópontba vezető legmesszebb vezető út hosszát úgy adhatjuk meg, hogy megmérjük az élek számát az úton. Ez a fa magassága.



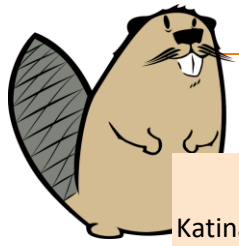
KISHÓD – NEHÉZ, BENJÁMIN – KÖZEPES

A hódok kerti party-ra készülnek, melyre egy salátát is szeretnének elkészíteni. Sajnos a szakács szabadságon van. Kati tudja milyen fontos, hogy az összetevőket a megfelelő sorrendben keverjék össze.

A kertben észreveszi, hogy a szakács segített neki: minden összetevőhöz egy útmutatót tett, mely a következő összetevőhöz vezet.



Melyik összetevőt kell Katinak először a salátába tennie?



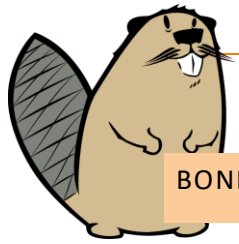
Katinak először a répát () kell a salátába tennie.

Az első összetevő csak az lehet, amelyik egyik táblán sem szerepel. Mivel akkor nincs olyan összetevő, amit előtte kellene a salátába rakni. Például, ha az alma lenne az első összetevő, az alma nem állhatna a répa melletti táblán.

MIÉRT INFORMATIKA?

A sorrend ilyen meghatározását az informatikában láncolt listának nevezzük. Gyakran akkor használják, ha egy kezdő elemtől megadott sorrendben szeretnének tovább lépegetni az egyes elemekhez. Ha minden elemünk adott és az első keressük, az nehéz lehet – mint azt tapasztalhattad a feladat során. Ezért is szokták külön az első elemet is megjegyezni.

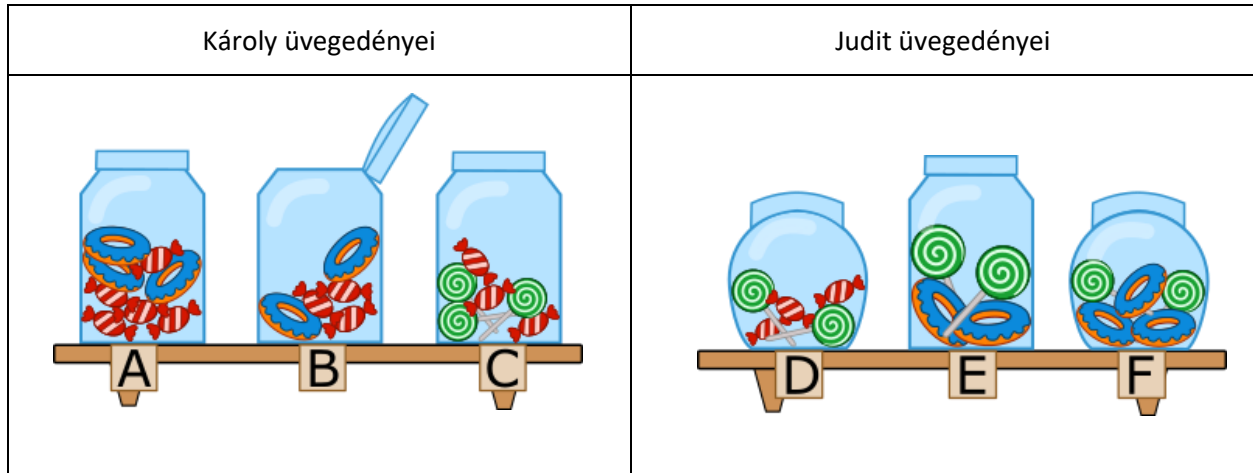
https://hu.wikipedia.org/wiki/L%C3%A1ncolt_lista



BONBONIER (2016-HU-06)

KADÉT – NEHÉZ, JUNIOR – KÖZEPES

Károlynak és Juditnak van 3-3 üvegedénye, melyekben édességeket tartanak.



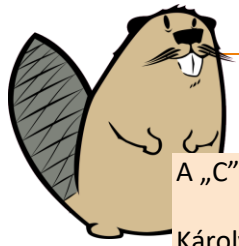
Minden üvegedény a következő tíz tulajdonságok közül többel is rendelkezhet:

- Vagy nyitva (1) vagy zárva (2) van.
- Vagy tartalmaz piros-fehér csíkos cukrot (3) vagy nem (4).
- Vagy tartalmaz kék cukorfánkot (5) vagy nem (6).
- Vagy tartalmaz zöld spirál-nyalókat (7) vagy nem (8).
- Vagy gömbölyű (9) vagy szögletes (10).

Károly „A” üvegedényének például a 2, 3, 5, 8 és 10-es tulajdonságai vannak.

Nézd meg jól! Károly üvegedényeinek vannak közös tulajdonságaik. Judit üvegedényeinek is vannak közös tulajdonságaik.

Egy üvegedénynek azonban Károly üvegedényeinek közös tulajdonságai és Judit üvegedényeinek közös tulajdonságai is megvannak. Melyik ez az edény?



A „C” edény a helyes válasz.

Károly üvegedényeinek (A, B és C) két közös tulajdonsága van:

- Az üvegedények szögletesek (10).
- Az üvegedények tartalmaznak piros-fehér csíkos cukrot (3).

Judit üvegedényeinek (D, E és F) ugyancsak két közös tulajdonságuk van:

- Az üvegedények tartalmaznak zöld spirál-nyalókát (7).
- Az üvegedények zárva vannak (2).

Csak a „C” jelű edénynek van meg mind a 2, 3, 7 és 10-es tulajdonsága: zárva van, szögletes, tartalmaz piros-fehér csíkos cukrot és zöld spirál-nyalókát. A 6-os tulajdonsággal is rendelkezik, az most számunkra lényegtelen.

MIÉRT INFORMATIKA?

Az informatika az adatok modellezésénél tulajdonságok alapján csoportosítja össze az objektumokat (elemeket). Ebben a feladatban tíz tulajdonsággal és két hármassal volt dolgunk. Olyan objektumot keresünk, melynek a tulajdonságai mindkét csoportra jellemzőek. A relációs adatbázisoknál ezt úgy is nevezik: „két halmaz metszetét képezzük”.

Nagy, átláthatatlan adathalmazokból leszűrhetjük („részhalmazát képezhetjük”) kívánt tulajdonságok alapján keresett objektumoknak.

Például egy webáruházban célzottan kereshetünk meghatározott kijelzőmérettel rendelkező okostelefonra, de tovább szűkíthetjük a keresésünk az összes tárolt tulajdonság alapján is.

Egy adatbázis felépítésénél kiemelt figyelmet kell arra fordítani, milyen tulajdonságok kerülnek elő az adatmodellben. Ha fontos tulajdonságokat kifelejtünk, az összes későbbi keresés kevésbé lesz hatékony. Ha felesleges tulajdonságokat is modellezünk, a későbbi karbantartás, tárolás költségesebb lesz anélkül, hogy a hasznosságot növeltük volna.



SZÁMOZOTT MEZEK (2016-IE-03)

JUNIOR – KÖZEPES, SENIOR – KÖNNYŰ

Az alábbi ábrákon számozott mezekben két hód-csapat sorakozott fel. Mindkét csapatban 15 játékos van. Az első csapat játékosai a mezükön látható számok szerint állnak sorban.

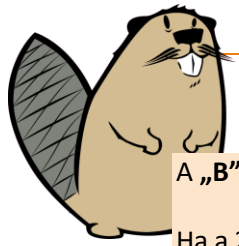


A második csapat játékosainak sorrendje nincs rendben.



Hogyan állapíthatjuk meg a leggyorsabban, hogy hány ugyanolyan számú mez szerepel az első és a második csapatban is?

- A) Az első csapat számain (1, 4, 5, ...) végighaladunk és mindegyiknél megnézzük, szerepel-e a második csapatban.
- B) A második csapat számain (8, 28, 12, ...) haladunk végig. Minden számnál megnézzük, hogy szerepel-e az első csapat mezein.
- C) Mindegy melyik csapattal kezdjük, ugyanolyan gyorsan fog menni.
- D) Először azt kell megállapítanunk, melyik szám nem szerepel mindkét csapatnál. Húzzuk ki ezeket a számokat és megkapjuk, a keresett számot.



A „B” válasz a helyes.

Ha a 2. csapat sorszámain kezdted, akkor az ott szereplő számok mindegyikét a rendezett 1. csapatban kell megkeresned. Egy rendezett sorban gyorsabb egy adott elemet (számot) megtalálni, mint egy rendezetlenben. Ez több okból is így van. Például azért, mert abbahagyhatjuk a keresést, amint az első olyan számot megtaláltuk, ami nagyobb, mint a keresett szám, hiszen a későbbiekben ez már biztosan nem szerepel.

Az 1. csapat számait sorban a 2. csapatban megkeresni sokkal több időbe kerülne. Ezért gyorsabb a „B” válasz, mint az „A”, és ezért hamis a „C” válasz is.

A „D” válaszban leírt módszer (ha a b válaszhoz hasonlóan a 2. csapat elemeit húzzuk ki az 1. csapatban keresve) ugyanolyan sokáig tartana, de még a kihúzás „művelete” időben lassítana is a folyamatot.

MIÉRT INFORMATIKA?

Egy rendezett sorban gyorsabban tudsz keresni, mint egy rendezetlenben.

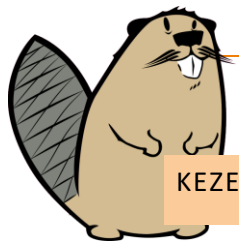
Hogy keresed meg például a 9-es számú mezt az első csapatban? Legelőször a sor közepén álló játékosra nézel. A 17-es száma van. Ennél a keresett szám (9) kisebb, tehát balra indulsz tovább – a balra álló játékosok felénél nézed meg a mez számát. Ez a 7-es, tehát jobbra indulsz...

A trükk az, hogy a keresési területet minden lépésben megfelezed és ezzel kisebbé teszed. Ezt az eljárást az informatikában bináris vagy logaritmikus keresésnek hívják.

Gondolj bele, hogy egymillió elem között kell keresned: ezzel a módszerrel 20 lépésben megtalálhatod az elemet, míg ha végiglépekedsz egyesével, akkor egymilliószor kell hasonlítanod két elemet.

Adathalmazokban való keresés az egyik leggyakoribb számítógépes feladat. Ezért a rendezésre és a keresésre többféle eljárást fejlesztettek ki, melyekről pontosan lehet tudni, milyen körülmények között melyik a leggyorsabb, legelőnyösebb.

[http://wiki.prog.hu/wiki/Logaritmikus_keres%C3%A9s_\(algoritmus\)](http://wiki.prog.hu/wiki/Logaritmikus_keres%C3%A9s_(algoritmus))

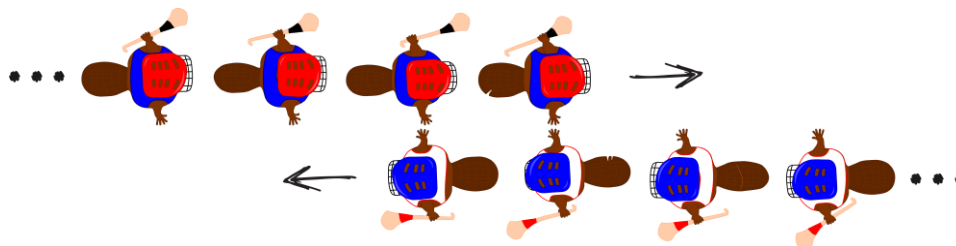


KEZET RÁZNI (2016-IE-04)

KADÉT – NEHÉZ, JUNIOR – KÖZEPES, SENIOR – KÖNNYŰ

A hódok szívesen játszó az ír eredetű hurling-ot. A játék befejeztével mindkét csapat egymás után feláll egy sorba. Ezután elmennek egymás mellett, kezét ráznak és azt mondják: „Köszönjük a játékot!”.

A kézrázás a következőképpen folyik: először a két első játékos ráz kezét, majd az első játékosok a másik csapat második játékosával (lásd képen).



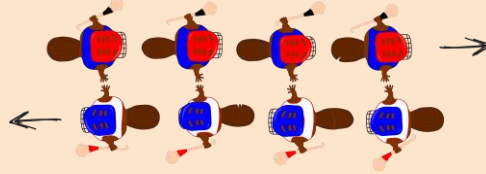
Ez így megy tovább, amíg a két utolsó játékos is kezét nem ráz egymással.

A hurling-nál 15 játékos van egy csapatban. Az, hogy két játékos kezét ráz és a következő játékoshoz lép, 1 másodpercig tart.

Mennyi ideig tart a kézfogás-ceremónia összesen?

**29 másodpercig.**

Amikor az első játékos elér az utolsóig és megrázza a kezét, az pontosan annyi idő, ahány játékos van egy csapatban. Például egy négy fős csapatnál ez 4 másodperc, és a játékosok így állnak:



Mindkét csapatban az utolsó játékos éppen a másik csapat első játékosával ráz kezét. Ezután még a másik csapat összes többi játékosával kezét kell ráznia. Ez négy fős csapatoknál eggyel kevesebb, mint négy, azaz 3 kézrázás, ami 3 másodperc.

Általánosan a kézrázások ideje az egyik csapat játékosainak számának kétszeresénél eggyel kevesebb. 15 játékos esetében $(2 \times 15) - 1 = 29$ másodperc.

MIÉRT INFORMATIKA?

A 15 játékosal rendelkező hurling csapatok esetében ki tudjuk számolni, hogy fut le a kézrázó-algoritmusuk. 29 másodperc még jól átlátható. De mi a helyzet például a jégchoki csapatok 22 játékosával? Jó lenne egy általános meghatározás az algoritmus lefutási idejére.

Az informatikusok intenzíven foglalkoznak az algoritmusok lefutási idejének általános meghatározásával, becslésével.

Az ilyen futási-idő vizsgálatok elvezethetnek egy bemeneti n változót tartalmazó matematikai kifejezéshez. A hurling-kézrázásnál a kifejezésünk a válaszban elrejtett utolsó előtti mondat alapján? $2n-1$, ahol n a csapat játékosainak száma.

Ezzel más játékos-számú csapatsportok kézfogási idejét is ki tudjuk számolni.

A $2n-1$ kifejezés mögött egy lineáris függvény áll. Ezzel a hurling-kézrázó algoritmus a lineáris futási idejű algoritmusok közé (osztályába) tartozik, melyeket $O(n)$ -nel jelölünk.

Egy olyan kézrázó-algoritmus, ahol mindenki mindenkivel kezét rázna egymás után, kevésbé lenne gyors. Az már az $O(n^2)$ osztályba tartozna, és a hurling-csapat esetében $15^2 = 225$ másodpercet, azaz majdnem 4 percet jelentene.

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



EGYSZERRE (2016-IE-05)

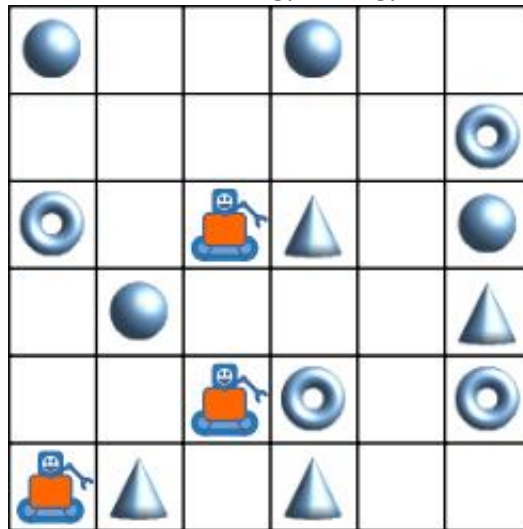
BENJÁMIN – KÖZEPES, KADÉT – KÖNNYŰ

Három robot csapatban dolgozik együtt. A csapatot a következő parancsokkal irányíthatod: N, S, O vagy W.

Egy paranccsal mindhárom robotot egyszerre irányítod: egy mezővel tovább a megadott irányba.

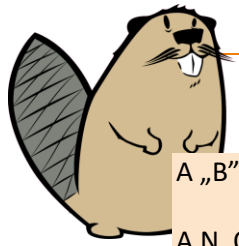
Amikor a robotok egy tárgyhoz érnek, azt felveszik. Hogy ne vegyenek fel felesleges tárgyakat, úgy kell irányítanod őket, hogy azokat kikerüljék.

Például: ha a robotokat a N, N, S, S, O parancsokkal utasítod, akkor összesen két kúpot és egy karikát vesznek fel.



Ha azt szeretnénk, hogy pontosan egy labdát, egy karikát és egy kúpot vegyenek fel a robotok, melyik parancssorral utasítsuk őket?

- A) N, O, O, O
- B) N, O, O, S, O
- C) N, N, S, O, N
- D) N, O, O, S, W



A „B” válasz a helyes.

A N, O, O, S, O parancssorral a zöld vonal mentén haladnak a robotok. A végén egy labdát, egy karikát és egy kúpot vesznek fel.

Az A válasszal (N, O, O, O) a robotok a kék vonalon mozognak. A végén két karika és egy kúp lesz náluk.

A C válasz (N, N, S, O, N) esetében a sárga vonal mentén két labdát és egy kúpot vesznek fel.

A D válasz (N, O, O, S, W) során, a piros vonalon haladva két kúp és egy karika lesz az eredmény.

MIÉRT INFORMATIKA?

A három robot egyidejűleg, azaz párhuzamosan mozog. Így a kívánt dolgokat gyorsabban célba tudják juttatni, mintha egymás után dolgoznának, vagy egy robotnak kellene mindent megvalósítania. Párhuzamosan gyorsabban megy!

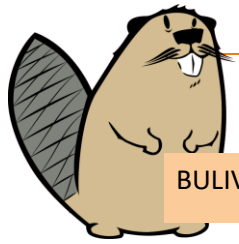
Az informatikai rendszerekben már régóta megvalósították a párhuzamosságot. Óriási szuperszámítógépek több processzorral, melyek párhuzamosan pl. időjárás-szimulációkat számolnak a pontosabb előrejelzés érdekében.

Amíg korábban a szuperszámítógépek óriási szerkezetek voltak, addig manapság a kis számítógépeket egy úgynevezett klaszter (angolul: cluster) hálózatba kötik össze. De párhuzamosság van a modern okostelefonokban is, melyben több processzor-mag dolgozik. A reklámokban pl. négymagos, nyolcmagos processzorként hallhatsz róluk.

Ha minden párhuzamosan dolgozó processzor ugyanazt a feladatot végezné saját adatokkal, sosem kereszteznék egymást, ahogy a példabeli három robot. Ez alapján az elv alapján építették régebben az úgynevezett vektorprocesszorokat, és a modern számítógép-klaszterekben az egyes számítógépek is ilyen szabályok alapján ugyanazt a munkát végzik. A párhuzamosságnak ezt a fajtáját könnyű megszervezni, de csak korlátozottan használható.

Rugalmasabbak azok a párhuzamos számítógépek, melyek ugyanazokon az adatokon dolgoznak, illetve pl. adatokat tudnak egymás között cserélni. De ezeknek a munkáját nehezebb megszervezni és az adatcsere is időt vesz igénybe. Az informatikusok azonban sokféle módszert kidolgoztak, melyekkel ezeket a problémákat megoldhatják.

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



BULIVENDÉGEK (2016-IS-02)

KADÉT – KÖNNYŰ

Sára bulit szervez. Szeretné meghívni a barátait: Alízt, Bernátot, Cilit, Dávidot és Emilt. De már jól ismeri őket és tudja, néhányan mindig tudni szeretnék, hogy bizonyos barátai eljönnek-e:

- Dávid tudni szeretné, hogy Alíz jön-e.
- Bernát tudni szeretné, hogy Emil jön-e.
- Cili tudni szeretné, hogy Bernát és Dávid jön-e.
- Alíz tudni szeretné, hogy Bernát és Emil jön-e.

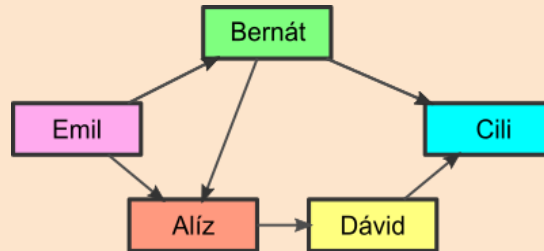
Tehát mielőtt Dáviddal beszélne Alízt kell megkérdeznie.

Milyen sorrendben kérdezze meg barátait Sára?



A helyes sorrend: **Emil, Bernát, Alíz, Dávid, Cili**.

Indoklásként rajzoljuk fel Sára barátait és a „szeretné tudni, jön-e a buliba“-feltételt jelöljük köztük nyíllal. Például Alíztól Dávidhoz vezet egy nyíl, hiszen Dávid tudni szeretné, jön-e a buliba Alíz.



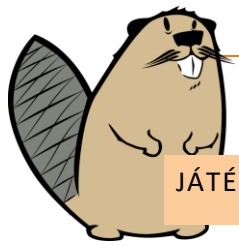
Ezzel egy gráf képe jelenik meg, ahol a személyek (barátok) a csomópontok és a feltétel köztük az irányított él. Hogy megoldjuk a sorrendiség problémáját Sárának egy utat kell találnia, mely az egyik csomópontból a nyilak irányában végigmegy minden többi csomóponton (hiszen mindegyik barátját meg szeretné hívni). Az egyetlen ilyen út az Emil → Bernát → Alíz → Dávid → Cili

MIÉRT INFORMATIKA?

Az életben gyakran megtörténik, hogy csak akkor tehetünk meg dolgokat, ha előbb bizonyos feltételeket már teljesítettünk.

Jó példa erre a főzés: a spagetti tésztát csak akkor tehetjük bele a vízbe, ha az már forr és beleszórtuk a sót is. A szósz csak akkor tehetjük a tésztára, ha az már megfőtt. De az, hogy a spagettit vagy a szósz egyidőben főzzük-e meg vagy egymás után, az mindegy.

A buli-meghívást és a főzést magunk is megtervezhetjük. De több lépés és feltétel esetében már nehezebben átláthatóak a dolgok. Ebben (is) segít az informatika: az ütemezéseket (angolul scheduling) végző számítógépes programok az iparban is használatosak. Az összetett folyamatok – mint például egy autó építése – hatékony, például várakozási idő nélküli megvalósításakor. Mindezzel pénz spórolható, nem szükséges nagy mennyiségű raktérezszközt kezelni és a gyártósorok szünet nélkül üzemelhetnek.



JÁTÉK A GOLYÓKKAL (2016-IT-02B)

JUNIOR – NEHÉZ, SENIOR – KÖZEPES

Emilnek van egy új játéka a számítógépén. A játék legalább három színes (piros vagy kék) golyóval kezdődik, amelyek egy csőben sorakoznak. Miután rákattint egy gombra, mindkét alsó golyó kiesik a csőből, valamint felülről új golyók esnek bele a csőbe. Ekkor két lehetőség van, a kattintás előtt legalul lévő golyó színétől függően:

Ha a legalsó golyó piros volt, egy kék színű golyó esik felülről a csőbe.	Ha a legalsó golyó kék volt, három golyó esik felülről a csőbe, piros-kék-piros sorrendben.

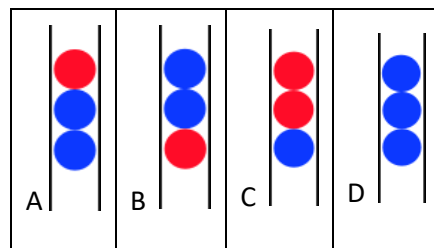
Amíg legalább három golyó van a csőben, Emil újra és újra rákattint a gombra. A játék akkor ér véget, ha kevesebb, mint három golyó marad a csőben.

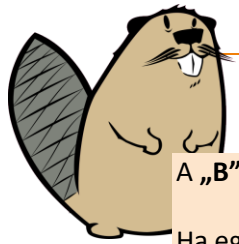
Egy példa: ha Emil ezzel a golyósorrenddel kezd, öt kattintás után csupán két kék golyó marad benn, és a játék véget ér.

Emil felfedezi, hogy léteznek olyan golyósorrendek a játék kezdetén, amelyekkel a játék sohasem ér véget, mindegy hányszor kattint. Az ilyen golyósorrendeket „végtelen sorozatoknak” hívja.



*Az alábbi hármas kezdő-sorozatok melyike **NEM** „végtelen sorozat”?*





A „B” válasz a helyes.

Ha egy hármas sorozatban a legelső golyó piros, már az első kattintás után véget ér a játék: ekkor csak két golyó marad a csőben.

Ezzel szemben minden olyan hármas sorozat, amelyben a legelső golyó kék, végtelen sorozat. Legfeljebb négy kattintással ugyanis a négy, kék legelső golyóval felépített lehetséges hármas sorozat mindegyike a piros-kék-piros-piros-kék-piros hatos sorozattá alakul.

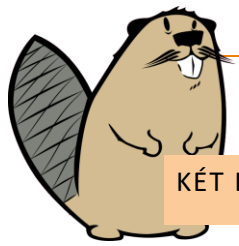
További négy kattintás után aztán ismét előáll ugyanez a sorozat, és ez így megy tovább a végtelenségig.

MIÉRT INFORMATIKA?

Emi Leon Post (1897-1954) lengyel matematikus és logikatanár volt, aki az elméleti informatika úgynevezett ítéletlogika területén számos cikket jelentetett meg. A jelen feladatban bemutatott játékot Post példaként használta arra, hogy bemutassa: az olyan rendszerekben, melyekben szimbólumláncokat meghatározott szabályok szerint manipulálnak, véget nem érő folyamatok léphetnek fel.

Ezekben a rendszerekben szimbólumláncok részeit más szimbólumláncokkal helyettesítik. Az elméleti informatikában „helyettesítéses rendszereknek” hívják őket. A lehetséges helyettesítéseket véges mennyiségű szabály teljes mértékben meghatározza. Az átalakításokra, amelyeket egy helyettesítéses rendszer megenged, egyfajta számolásként is tekinthetünk – és ezáltal válnak fontossá az informatika számára. A Post által kitalált helyettesítéses rendszerek, valamint más helyettesítéses rendszerek, illetve az úgynevezett részleges-Thue-rendszerek tekintetében bebizonyítható, hogy egyenértékűek a Turing-gépekkel. Ez azonban azt is jelenti, hogy léteznek a helyettesítéses rendszerek által el nem dönthető kérdések; a Turing-gép megállási problémája itt a „szóproblémának” felel meg. Az informatika így ismer olyan, erősen korlátozott helyettesítéses rendszereket, pl. kontextusmentes nyelvtanokat, amelyekkel többek közt programnyelvek is leírhatók.

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

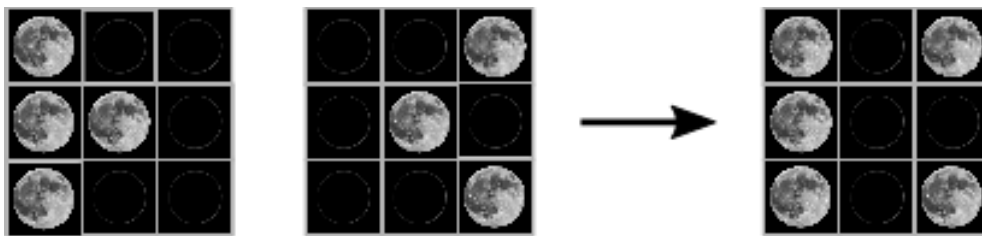


KÉT LEHETŐSÉG (2016-JP-02)

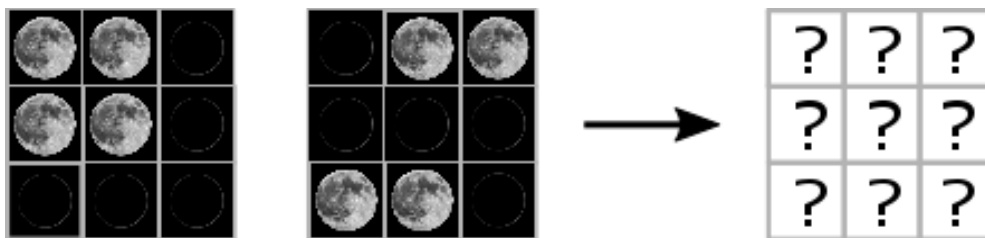
JUNIOR – NEHÉZ, SENIOR – KÖZEPES

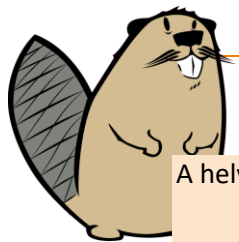
A bölcс Anaxagorasznak, a tudósnaк olyan mágikus holdkártyái vannak, melyek mind a kilenc lehetséges mezőjükön egy meghatározott szabály szerint működnek.

Amikor két kártya minden mezőjére megad egy hold-szimbólumot: újhódat vagy telihódat, azonnal előtűnnek a holdszimbólumok a harmadik kártyán. Íme egy példa. Balra láthatod a két megadott kártyát, jobbra pedig azt, amelyiken megjelennek a szimbólumok:



Felismered a szabályt? Akkor mondd meg, hogy a két megadott kártya alapján milyen lesz a harmadik!





A helyes válasz:



A szabály az, hogy ha a két kártyának ugyanazon a mezőjén pontosan egyszer telihold van, akkor a harmadik kártya ugyanazon mezőjén is telihold szimbólum jelenik meg. Különben pedig az újhold.

MIÉRT INFORMATIKA?

Az informatika több területén használják előszeretettel a kétértékű logikát ahhoz, hogy döntsenek egy kijelentés igaz vagy hamis voltáról egy modell alkalmazásában.

Ez csak olyan feltételekkel működik, ha a modellünkben pontosan két lehetőség (két érték) létezik. Harmadik lehetőség ki van zárva: tertium non datur.

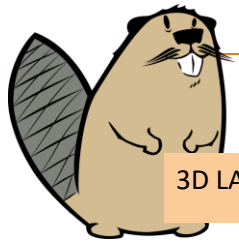
Néhány példa ilyen két-értékű-halmazra modellekben: {nyitva, zárva}, {egyenes, nem egyenes}, {üres, teli}, {fekete, fehér}, {igen, nem}, {0,1}.

Erre alapozva már a modellünkben logikusan összeköthetőek egymással a kijelentéseink. Egy logikai függvény ezek után az „igaz” vagy „hamis” értéket fogja visszaadni. Mivel logikai függvényeket is össze tudunk egymással kapcsolni, így tetszőlegesen sok létezik belőle.

Ebben a feladatban a XOR („kizáró vagy”, angolul: „exclusive OR”) szerepel, ami meglehetősen sokszor előfordul és könnyen megérthető. A XOR így működik: amennyiben a kijelentések közül pontosan egyben szerepel „igaz” érték, a többiben „hamis”, akkor a kimenet „igaz”, különben „hamis”.

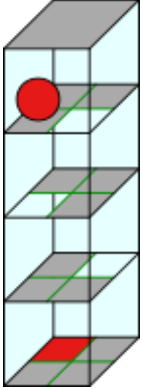
Érdekesség: AnaXagORasz ismerte fel először, hogy a Hold nem világít. Egy Holdkráter is viseli a nevét az elrejtett XOR-ral :)

https://hu.wikipedia.org/wiki/Kiz%C3%A1rt_harmadik_elve



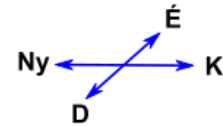
3D LABIRINTUS (2016-JP-03)

BENJÁMIN – KÖZEPES, KADÉT – KÖNNYŰ



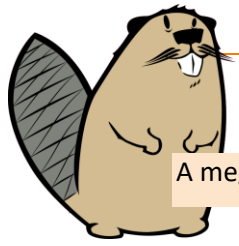
Egy 3D labirintus négy szintből áll, szintenként négy mezővel. A legfelső szinten van egy golyó, amit a legalsó szinten lévő célba (piros mező) kell eljuttatni.

A golyót az É, D, K és Ny utasításokkal irányíthatod (lásd kép).



A fehér mezőkön a golyó egy szinttel lejjebb esik. A labirintus zárt, oldalra nem gurulhat ki a golyó.

Milyen utasításokkal tudod célba juttatni a golyót?



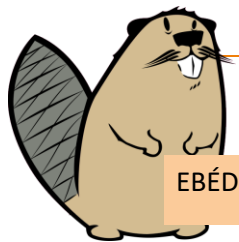
A megoldásban több lehetséges válasz is helyes . Pl. a KNyÉKNy, KNyKÉNy vagy akár a KÉNyDÉKNy is.

MIÉRT INFORMATIKA?

Az utasítások sorozata, melyekkel a golyót a labirintusban irányítjuk, olyan, mint egy rövid számítógépes program.

A hozzá tartozó programozási nyelv csak négy utasítást ismer, az É, D, K és Ny parancsokat.

Ebben a nyelvben egy program az egyes utasítások sorozata, melyek egymás után (informatikus nyelven szekvenciálisan) kerülnek végrehajtásra. A legtöbb "valódi" programnyelvben az utasítások sorozata (szekvencia) az egyik legalapvetőbb lehetőség. További lehetőséget jelent az ismétlés, a feltételes utasítás vagy a gyakran használt programrészek alprogramokba (szubrutinok) való kiemelése.



EBÉDSZÜNET (2016-LT-03)

BENJÁMIN – NEHÉZ, KADÉT – KÖZEPES

Ebédészünete alatt (12:00-13:00) Alexandra a következőket szeretné elintézni:

- venni egy könyvet a könyvkereskedésben;
- venni egy üveg tejet az élelmiszerüzletben;
- az újonnan vett könyvet feladni a postán;
- inni egy kávé a kávézóban.

E négy tevékenység mindegyikéről Alexandra meghatározta, mennyi időt vesznek igénybe. Az alábbiakban listázott időtartamok azonban csak a csúcsidőn kívül érvényesek. Ezért Alexandra megpróbálja elkerülni a csúcsidőt.

Hely	Időtartam	Csúcsidő
könyvesbolt	15 perc	12:40 – 13:00
élelmiszerüzlet	10 perc	12:00 – 12:40
posta	15 perc	12:00 – 12:30
kávézó	20 perc	12:30 – 12:50

Milyen sorrendben intézze el Alexandra a teendőit, hogy elkerülje a csúcsidőt?



A helyes sorrend: **kávézó, könyvesbolt, posta, élelmiszerüzlet.**

Ez a probléma felvet néhány megkötést. Táblázatban ábrázolva ez a következőképpen néz ki (sötétpiros: csúcsidő, világoszöld: csúcsidőn kívüli):

hely	látogatás időtartama	12:00	12:05	12:10	12:15	12:20	12:25	12:30	12:35	12:40	12:45	12:50	12:55
könyvesbolt	15p					X	X	X					
élelmiszerüzlet	10p											X	X
posta	15p								X	X	X		
kávézó	20p	X	X	X	X								

Alexandrának 12:40 előtt már meg kellett járnia a könyvesboltot. Az élelmiszerüzletbe 12:40 után kell mennie. A postára csak akkor mehet, ha a könyvesboltban már volt. A postát legkorábban 12:30 után látogathatja meg. A kávézóban 12:30 előtt kell lennie, mivel 12:50 után már nincs elég ideje az ebédszünetből. Ha a könyvesboltban kezdene, a kávézóra már nem jutna ideje (belecsúszna a csúcsidőbe).

Az egyetlen lehetséges időbeosztás tehát (az ábrán „X”-szel jelölve) leolvasható.

MIÉRT INFORMATIKA?

Az informatika egyik legfőbb célja, hogy a problémákra olyan megoldásokat találjon, amelyek összeférnek a megadott megszorításokkal. A mi esetünkben a csúcsidőszakokat kell elkerülni az üzletekben. Más, megszorításokkal korlátozott problémánál gyakran felmerül a kérdés, vajon van-e egyáltalán megoldás, illetve figyelembe vehető-e egyszerre az összes megszorítás.

Ezeket a kérdéseket az informatikában ütemezési problémának (scheduling problem) nevezik. Az ütemezés alatt egy helyes vagy optimális sorrend meghatározását kell érteni bizonyos feladatok megoldásához. Ipari felhasználásban nagyobb projekteknél, vagy részegységek termelésénél is használják. A számítógépekben is gyakran alkalmazzák, például amikor több számítást kell elvégezni több processzormagon.



KIX KÓD (2016-NL-04)

KADÉT – NEHÉZ, SENIOR – KÖNNYŰ

Hollandiában az irányítószámok négyjegyűek: számokat és betűket is tartalmazhatnak. Egy egyedi vonalkódja is van minden irányítószámnak, amit Kix kódnak neveznek.

A Kix kód minden sorában van egy felső rész, két hosszú és két rövid vonással, valamint egy alsó rész ugyancsak két hosszú és két rövid vonással. Középen fedik egymást a rövid vonások. A képen (a táblázatban) láthatod a 0, a 7, a G és az Y Kix kódját.

A G7Y0 irányítószám Kix kódja tehát így néz ki:



Milyen irányítószámot jelöl az alábbi Kix kód?



	0	1	2	3	4	5
	6	7	8	9	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



BC16 a helyes válasz.

MIÉRT INFORMATIKA?

Hollandiában a posta valóban a Kix kódot használja. A gépek számára olvasható kódok lehetővé teszik levelek, csomagok gyorsabb osztályozását, válogatását.

Ilyen és ehhez hasonló kódokkal gyakran találkozatsz. Ilyenek a vonalkódok (sokáig bar-kódként használták a „bar” angol vonal szóból), melyek a pénztárakban segítenek azonosítani az árukat. Vagy például a QR kódok, melyeket az autóiparban fejlesztettek ki azonosításra, de már a világon mindenütt elterjedtek és reklámokban is leolvasható üzeneteket, elérhetőségeket is tárolhatnak.

<https://en.wikipedia.org/wiki/RM4SCC>

<https://hu.wikipedia.org/wiki/Vonalk%C3%B3d>



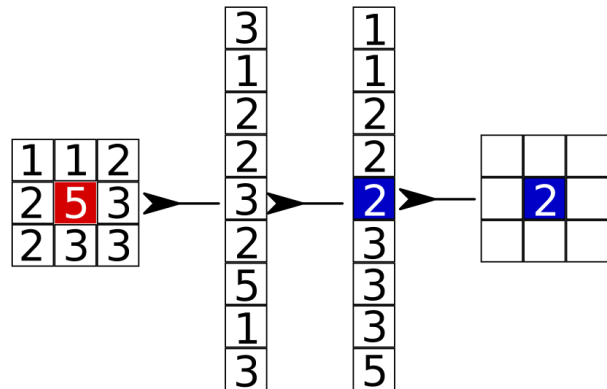
KÖZÉPSZŰRŐ (2016-RU-02)

KADÉT – NEHÉZ, SENIOR – KÖNNYŰ

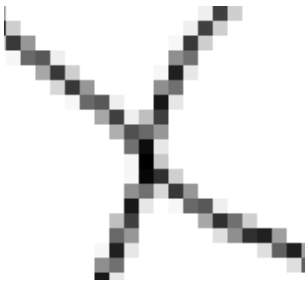
Egy képet táblázatként tárolunk, ahol minden cella egy képponthez tartozik. Az 1-es számú cellaérték fekete, míg az 5-ös számú cellaérték fehér képpontot jelent, és a köztes értékek (2, 3, 4) a szürke különböző árnyalatait jelölik.

Egy úgynevezett „Középszűrő” az eredeti képpel azonos méretű szűrt képet készít.

Ehhez minden képpontnál – a példában pirossal megjelölt – az eredeti képen szereplő értéket és a nyolc szomszédos cellákban szereplő értéket sorbarendezi.



Ezután a középsőt, azaz a kilenc érték közül az ötödiket adja új értéknek a szűrt képen. A példánkban ezt késsel jelöltük.



Hogyan fog megközelítőleg kinézni az alábbi kép középszűrés alkalmazása után?

A	B	C	D



Az „A” válasz a helyes.

Az efféle középszűrők esetén a 9 képpontos négyzetekben dolgozzuk fel a képet. A középső képpont új értéket kap. A példán látható négyzetet megvizsgálva láthatjuk, hogy a 9 képpontból csak 3 fekete (mint a szűrő használata előtt a képünk közepén), a többi pedig világosabb nála, így a fekete képpontból világosabb lesz.

A feladat képén minden 9 képpontú négyzetben kevesebb a fekete képpont, ezért a szűrt képen nem szerepelhet fekete képpont.

Ennek csak az „A” megoldás felel meg.



MIÉRT INFORMATIKA?

A képek feldolgozásakor szeretnénk gyorsan megfelelő hatást elérni. Gyakran élesebb kép vagy élettelibb színek a célunk, néha viszont szeretnénk művészebb hatásokat is elérni, hogy a kép egy jellegzetes jegyet viseljen magán. Ezeket a hatásokat képszűrőkkel érhetjük el.

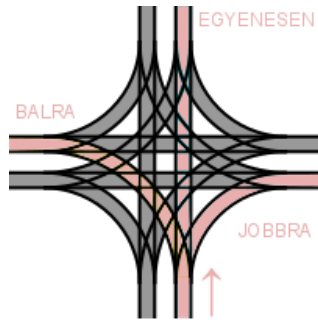
A középszűrő egy ilyen szűrő. Hasznos például olyan esetben, amikor néhány, a képrögzítő hibájából adódó képponthibát szeretnénk kijavítani. A kép ilyenkor egy kissé simítottnak hat, és nem lesz többé megtalálható az a néhány feltűnő, kilógó pötty. A zaj bizonyos formái így csökkenthetők.



VILLAMOSHÁLÓZAT (2016-RU-04)

JUNIOR – NEHÉZ

Szentpétersvár utcáin villamosok közlekednek.

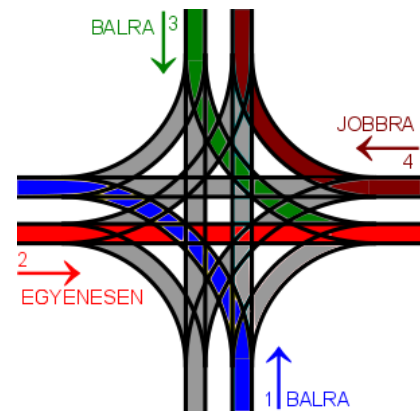


Az egyik kereszteződésben minden lehetséges: a villamosok mind a négy irányból érkehetnek és mindhárom másik irányba tovább haladhatnak.

A váltó állása meghatározza, hogy haladhatnak tovább a villamosok. A váltóállítást a következőképpen írhatjuk le: az óramutató járásával megegyezően adják meg az eredeti, érkezési irányhoz képest, hogyan kell továbbhaladni: EGYENESEN, JOBBRA vagy BALRA.

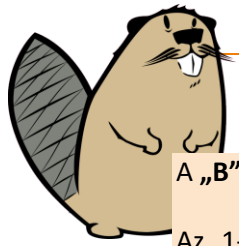
Egy példa: BALRA-EGYENESEN-BALRA-JOBBRA azt jelenti, hogy az 1-es vágányról érkező villamos balra kanyarodik, a 2-es vágányról érkező egyenesen megy tovább, a 3-as vágányról balra és a 4-es vágányról érkező jobbra kanyarodik.

Kétségtelenül ez a váltóállás ütközéshez is vezethet: az 1-es és a 2-es vágányon érkező és a 2-es és a 3-as vágányon érkező villamosok között.



A következő váltóállások közül melyik vezethet ugyancsak ütközéshez?

- A) JOBBRA - JOBBRA - JOBBRA - JOBBRA
- B) JOBBRA - JOBBRA - BALRA - BALRA
- C) BALRA - JOBBRA - BALRA - JOBBRA
- D) JOBBRA - BALRA - JOBBRA - BALRA



A „B” válasz a helyes.

Az 1-es és 3-as vágányokról érkező villamosok összeütközhetnek, hiszen ugyanoda kanyarodnak. Hasonlóan összeütközhetnek a 2-es és 4-es vágányról érkező villamosok is.

A többi váltóállás ütközésmentesen irányítja tovább mindegyik villamost:

Az „A” esetben mindegyik villamos csak külső íven kanyarodik.

A „C” és „D” esetben két-két villamos párhuzamos kanyarral kerüli el egymást. A balra kanyarodó villamosok a belső kanyaron nem zavarják egymást. Az efféle műveletet érintőleges vagy amerikai balkanyarnak is nevezzük.

MIÉRT INFORMATIKA?

A villamoskereszteződés példa olyan erőforrásra, amit több fogyasztó használ egyszerre.

Párhuzamosan futó folyamatok közös felhasználásának irányítása és optimalizálása az informatika egyik fontos feladata.

Ahogy a villamossíneknél, úgy a digitális technológiák esetében is oda kell figyelni a lehetséges veszélyforrásokra, és a biztonságos közös használathoz szabályokra van szükség. Különben akár adatvesztéshez vagy a hardver sérüléséhez is vezethet egy-egy „ütközés”.

A villamoskereszteződés esetében jó ötlet, ha csak biztonságos váltóállásokat használunk. Vagy jelzésekkel biztosítjuk a síndarabokat, melyeken ütközés történhet. Az informatikában is használnak ilyen jelzéseket, az úgynevezett szignálokat (angolul: signal), melyek jelzik, hogy ha meghatározott erőforrások foglaltak, és így elkerülhető a kritikus egyidejű használat.

https://hu.wikipedia.org/wiki/P%C3%A1rhuzamos_algoritmus



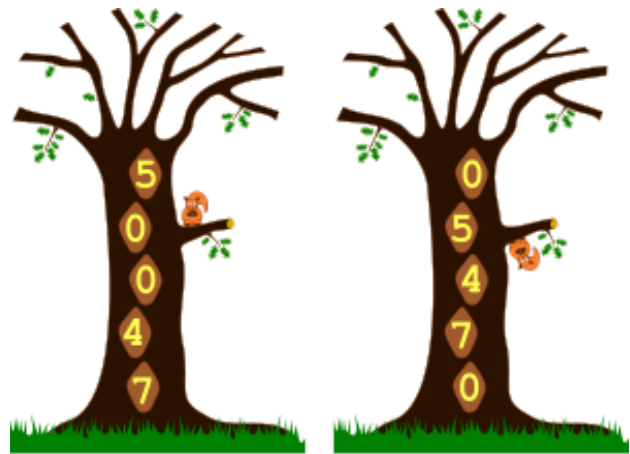
ÖNZŐ MÓKUSOK (2016-RU-08)

SENIOR – NEHÉZ

A mókusok odukban élnek. Egy fán öt odú helyezkedik el egymás felett. Ezen a fán tizenhat mókus él, ami azt jelenti, hogy együtt élnek ebben az öt odúban.

Minden nap minden mókus ellenőrzi, melyik szomszédos odúban található a legkevesebb mókus. Ez azt jelenti, hogy megszámolják, hány szomszédjuk van a felettük vagy az alattuk lévő odúban. A következő éjszaka eltöltésére minden mókus azt a szomszédos odút választja, amelyikben a legkevesebben aludtak. Ha az egymás alatti odukban ugyanannyian aludtak, a mókusok a saját odujukat részesítik előnyben a felettük vagy alattuk lévőhöz képest.

Ha például ma 5, 0, 0, 4 és 7 mókus alszik az odukban (fentről lefelé haladva), akkor holnap a következőképpen fog kinézni a helyzet: mind az 5 mókus, aki a legfelső odúban töltötte az éjszakát, a közvetlenül alattuk lévő odúba fog költözni (mivel 0 lakótárs jobb, mint 4). A legalsó barlang 7 mókusa feljebb fog költözni (4 lakótárs jobb, mint 6), és a 4, a legalsó odú fölötti odúban alvó mókus egyvel feljebb költözik (0 szomszéd jobb, mint 3).



Ha ma kezdetben (fentről lefelé haladva) 6, 3, 3, 0 illetve 4 mókus található az egyes odukban, akkor hány nap múlva lesz végül minden mókus ugyanabban az odúban?



Három nap múlva.

Ma: 6, 3, 3, 0, 4

1 nap múlva: 0, 9, 0, 7, 0

2 nap múlva: 9, 0, 7, 0, 0

3 nap múlva: 0, 16, 0, 0, 0

MIÉRT INFORMATIKA?

Ez a probléma a rajintelligencia (kollektív intelligencia) egy példája. Az ilyen algoritmusok alapötlete az, hogy a komplex problémák nagyon egyszerű eszközökkel megoldhatók, ha ezekből az eszközökből igen sok áll rendelkezésre. Például a hangyák egyszerű szabályok által meghatározottan és egymástól függetlenül viselkednek. Azonban, ha sok hangya áll rendelkezésre, akkor ezek már igen nagy kihívást jelentő dolgokat is megtehetnek, mint például a hangyabolyépítés, az optimális út megtalálása egy diagramon, megoldás találása az eladóproblémára, vagy akár levelek feldarabolása.

Ebben a feladatban szintén „nagy” mennyiségű (jelen esetben mókusok által képviselt) eszközzel van szó, amelyek viselkedése egyszerű szabályokat követ. Jelen esetben mindazonáltal kollektív viselkedésük nagyon távol áll az „intelligens” léttől. Olyan sok helyet szeretnének maguknak, amilyen sokat csak lehet, végül mégis ugyanabban az odúban lyukadnak ki. Ebből le lehet vonni a következtetést, miszerint a hangyaszerű viselkedés nem fordítható le egy az egyben hangya-algoritmussá. Alkalmanként ugyanis jobb együttműködni, mint önző módon viselkedni.



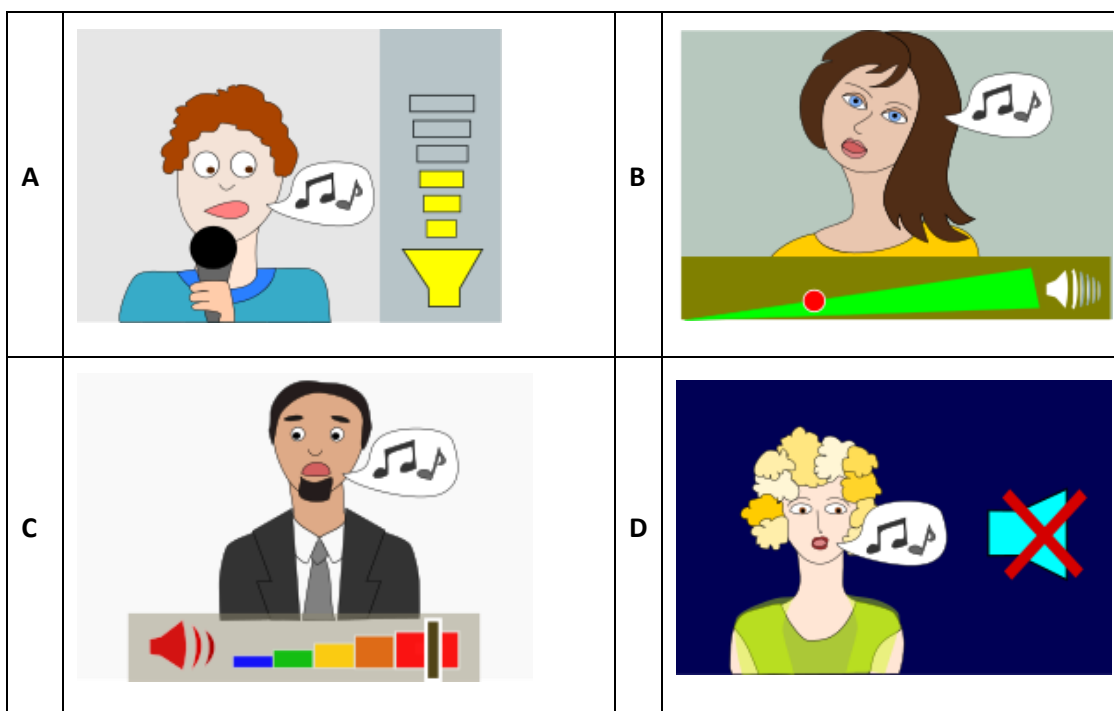
HALK VAGY HANGOS? (2016-SK-01)

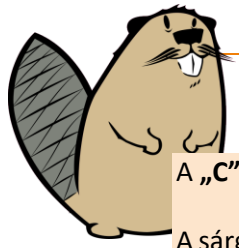
KISHÓD – KÖZEPES

Egy zenei alkalmazás kísérőzenét játszik. Hangosan vagy halkan. Éppen úgy, ahogy a négy képen a hangerőszabályzó be van állítva.

Egy ember pedig igyekszik hozzá szöveget és dallamot illeszteni. A karaoke jó mulatság

Melyik képen van a hangerő a legmagasabbra állítva?





A „C” válasz a helyes.

A sárga hangerőszabályzós képen letről (halk) felfele (hangos) nagyjából a feléhez van beállítva a csuszka.

Azon a képen, ahol balról (halk) jobbra (hangos) szélesedő zöld sáv a hangerőszabályzó, a csuzkát jelző piros golyó a nagyon halkra van állítva.

A képen, amelyiken az egyre növekvő színes négyszögek kéktől (halk) pirosig (hangos) jelzik a hangerőt a függőleges fekete csuszka egészen jobbra, teljes hangerőre van állítva.

Az áthúzott hangszóró-szimbólumos képen a hangszóró némára van állítva.

Az elnémított szólót énekel.

MIÉRT INFORMATIKA?

Kisebb-nagyobb képernyőinken a grafikus szimbólumok tömegével találkozunk.

Némelyik a program állapotát mutatja meg. Némelyik a használható programfunkciókhoz tartozik.

A grafikus szimbólumokat a felhasználónak meg kell tanulnia és értenie kell.

Ez fárasztó és drága. Ezért használnak a gyártók az ember-gép interakciókhoz gyakran a valós életben használtakat.

Az informatika kifejlesztette az operációs rendszereket, melyek egyrészt munkahatékonyak a felhasználók számára, másrészt az esztétikai élmény érdekében teljesen átstilizáltak.

Ettől paradigmátikus. Tipikus paradigma például az irodai íróasztal dossziékkal, dokumentumokkal, papírkosárral vagy a pilótafülke eszközökkel, kapcsolókkal és kijelzőkkel vagy az intelligens digitális kézitáska app-okkal és állományokkal.

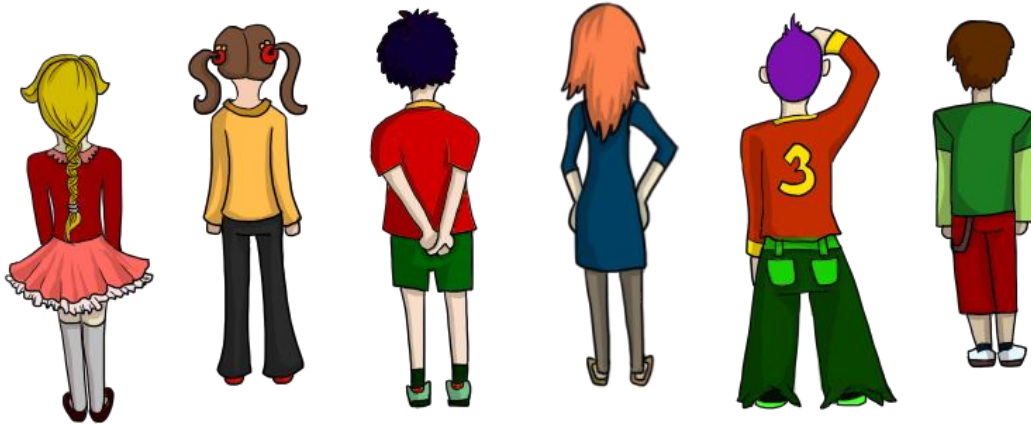


KI VOLT AZ? (2016-SK-02)

KISHÓD – KÖNNYŰ

Hat gyerek játszik kint egy labdával.

Egyszer csak üvegtörés hangja hallatszik. Odafutok az ablakhoz és észreveszem, hogy az kitört.



Jana

Eva

Jojo

Anne

David

Tobi

Egy gyerek elrohan. Hosszú haja van és hosszú nadrágot visel.

Ki volt az?



Eva a helyes válasz.

Három gyereknek van hosszú haja: Jana, Eva és Anne.

Hárman hordanak hosszú nadrágot: Eva, David és Tobi.

Mivel csak Eva tartozik mindkét csoportba, így csak ő lehetett az, aki elrohant.

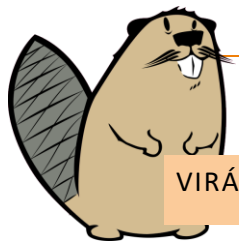
MIÉRT INFORMATIKA?

A feladat az információk rendszerezéséről szól. A közös tulajdonságú adatokat táblázatosan akár egy adatbázisban is tárolhatjuk.

Ilyen közös tulajdonság a feladatban szereplő gyerekek esetében a haj illetve a nadrág hossza, a póló színe stb.

Minden gyerek esetében minden tulajdonsághoz egy érték tartozik. Például Jana-nak és Eva-nak is van hajszíne, de ennek értéke Jana esetében "szőke", Eva esetében pedig "barna".

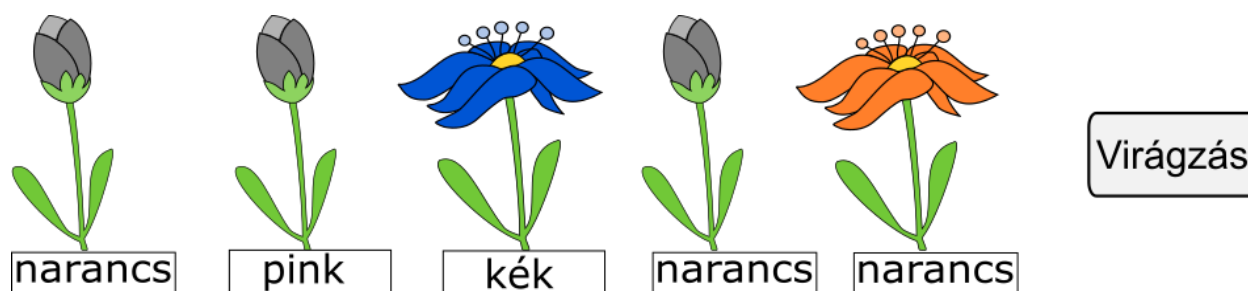
Az informatikában mindig szaktudáson, általános műveltségen és élettapasztalaton alapuló kihívás az adatbázisok számára lényeges tulajdonságok (attribútumok) meghatározása és ezek alapján a kívánt információk lekérdezése, kikeresése.



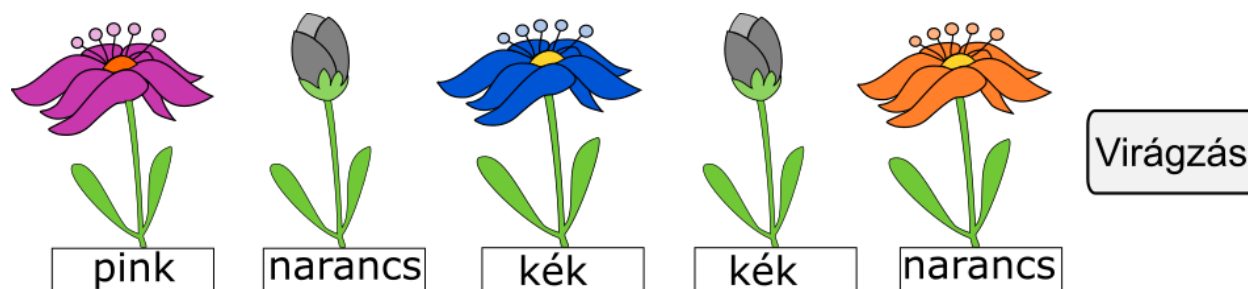
VIRÁGZÁS (2016-SK-04)

KISHÓD – NEHÉZ, BENJÁMIN – KÖZEPES, KADÉT – KÖNNYŰ

Jana egy számítógépes játékkal játszik. A számítógép öt virágnak titokban választott színt. A lehetséges színek a kék, a narancssárga és a pink. A választott színek nem változnak a játék során. Jana is mindegyik virághoz kiválasztott egy színt és a Virágzás gombra kattintott. A helyes színnel kijelölt virágok kinyíltak, míg a többi nem.



Ezután Jana néhány virág színét megváltoztatta és újra a Virágzás gombra kattintott. A második próbálkozás végeredményét az alábbi kép mutatja.



Melyik elrendezésben választhatta ki a számítógép a virágok színeit!



A helyes válasz: **pink – kék – kék – pink – narancs**.

A két találgatás után három kinyílt virág látható, tehát már látjuk, hogy a számítógép milyen színt választott az első, a harmadik és az ötödik virágnak. A második virágnál Jana először pinket tippelt és nem nyílt ki; majd narancssárgát tippelt és még mindig nem nyílt ki. Mivel összesen háromféle szín van, a második virág csakis kék lehet.

Hasonlóan, Jana a narancssárgát és kéket választotta a negyedik virág színének, de egyik esetben sem nyílt ki, így csak pink lehet.

MIÉRT INFORMATIKA?

Számos probléma megoldásakor hasznos képesség, hogy megtörtént – vagy pont meg nem történt – eseményekből következtetéseket vonjunk le. A feladat a Mastermind színkeresős táblajáték egyszerűsített változata. Az egyszerűsítés az, hogy a játékos a tippelés után minden információt megkap a virágok színének helyességéről. Ha a játékos minden tippelés során új színt választ a még ki nem nyílt virágok számára, akkor a harmadik tippben mindig helyesen tudja kiválasztani az összes virág színét.

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



TITKOS ÜZENETEK (2016-UK-06)

BENJÁMIN – NEHÉZ, JUNIOR – KÖNNYŰ

Borisz és Berta ügynökök titkos üzeneteken keresztül tartják a kapcsolatot, hogy rajtuk kívül senki se értse miről beszélnek.

Borisz a következő üzenetet szeretné elküldeni Bertának:

ASZOKOTTHELYEN6KOR

A szöveg betűit egy táblázatba írja, melynek 4 oszlopa van, a táblázatot a betűkkel balról jobbra haladva, fentről lefelé tölti fel. Az üres cellákat a táblázat végén #-kokkal tölti ki. Az eredmény a következő:

A	S	Z	O
K	O	T	T
H	E	L	Y
E	N	6	K
O	R	#	#

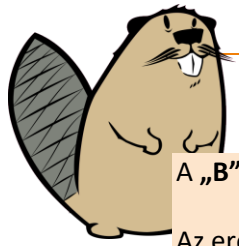
Ezután a kódolt üzenetet úgy készíti el, hogy a betűket oszloponként fentről lefelé, majd balról jobbra olvassa össze: AKHEOSOENRZTL6#OTYK#

Berta ugyanezzel a módszerrel válaszol Borisznak és így a következő üzenetet küldi el:

PETNGARKIORCEFKPESO#

Milyen választ küldött Berta?

- A) KERESKMEGTELEFONT
- B) PARPERCETKESNIFOGOK
- C) OK#ESTIGFOGOKKOPOGNI
- D) PERECETISENNIFOGOK



A „B” válasz a helyes.

Az eredeti szöveget a következőképp találhatjuk ki:

Töltsük fel a 4 oszlopos táblázatot, de ezúttal úgy, ahogy a titkosításkor kiolvastuk a kódot, vagyis a kódolt üzenet betűit oszloponként haladva fentről lefelé írjuk be. Így a végén balról jobbra olvasva az eredeti üzenet jelenik meg.

P	A	R	P
E	R	C	E
T	K	E	S
N	I	F	O
G	O	K	#

MIÉRT INFORMATIKA?

Számítógépes hálózaton küldött leveleink könnyen „elfoghatók”. De természetesen azt szeretnénk, hogy csak a valódi címzett tudja az üzenetünket megérteni és senki más.

Ehhez az üzenet szövegét kódolnunk kell és a titkos üzenetet kell elküldenünk. Ha csak a valódi címzett tudja, hogy kell az üzenetet megfejteni (dekódolni), és így az eredeti üzenet szövegét visszaállítani, a titoktartás sikeres volt.

Évszázadok óta van oka az embereknek, hogy üzeneteiket titkosítsák. Ezért rengeteg eljárás létezik a titkosítás (és a megfejtés) megoldására. Ebben a feladatban a kódolás alapja az „áthelyezés” (transzpozíció). Maguk a betűk változatlanok maradnak, csak a sorrendjük módosul.

Az ilyen és ehhez hasonló áthelyezéssel kódolás azonban aránylag könnyen megfejthető.

A kriptográfia (kódfejtés tudománya) a mai kutatások egy széleskörűen alkalmazott és vizsgált ágazata, az informatika és a matematika határvonalán. A modern titkosítási eljárásokat már általában számítógépekkel végzik.

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



SZŰK KERESZTMETSZETEK (2016-US-03B)

KADÉT – KÖZEPES

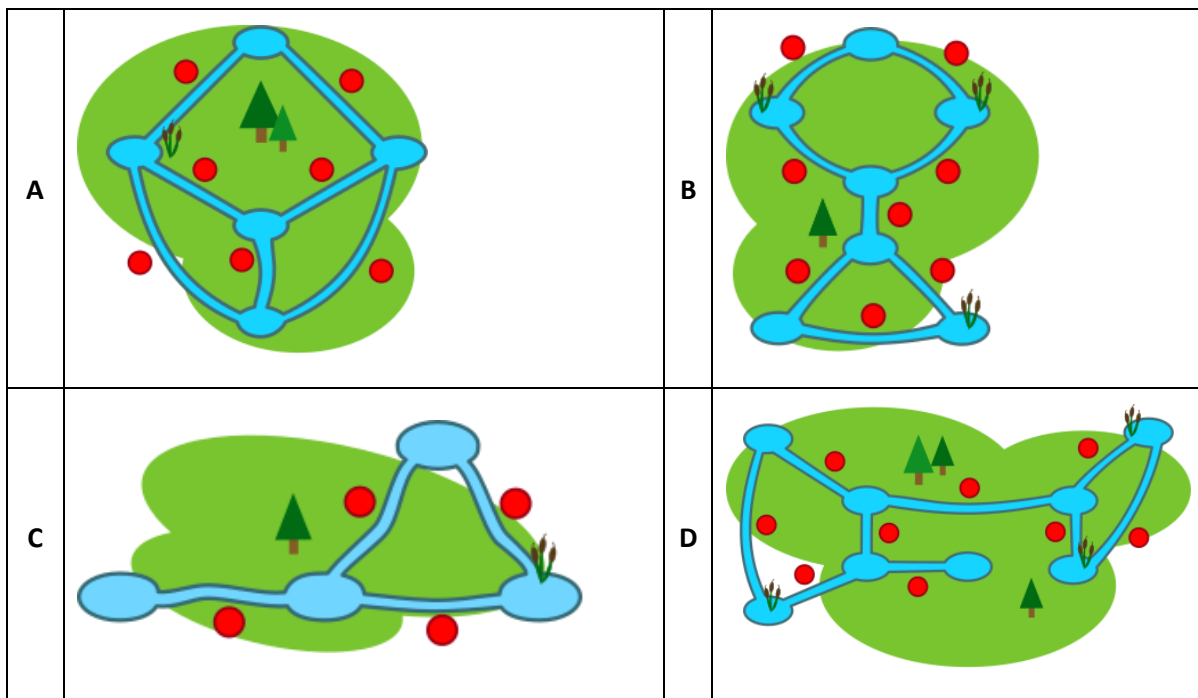
A hódok gátakat építenek. Amikor végig úsznak a csatornáik mentén, muszáj minden gátnál a szárazföldön átkelniük. Ezt viszont egyáltalán nem szeretik. Akkor inkább más csatornákon is végig úsznak, hogy elérjenek a céljukhoz.

Azokat a csatornákat, amelyeken némelyik útjuk során mindenképpen át kell haladniuk, mivel nincs másik útvonal, amely ugyanahhoz a célhoz vezetne, szűk keresztmetszeteknek hívják. Az ilyen helyeken inkább nem építenek gátat.



A képeken a hódok néhány területe látható. A piros jelzések mutatják az építendő gátak lehetséges helyét a csatornák mentén.

Csupán egyetlen területen nincsen szűk keresztmetszet. Melyik ez a terület?





Az „A” válasz a helyes.

Ezen a területen minden csatornának van egy alternatív (másik lehetséges) útvonala.

A B terület közepén található egy szűk keresztmetszet.

A C területen a szűk keresztmetszet egészen bal oldalt van.

A D területen két szűk keresztmetszet is van középén.

MIÉRT INFORMATIKA?

A hódok területe egy, csatornákból és tavacskákból álló hálózat. Ezt az internethez lehet hasonlítani. Ott a számítógépek, mobiltelefonok, televíziók stb. a tavacsák. A vezetékek illetve a rádiókapcsolatok a folyók. Az internet eredeti célja az volt, hogy USA-beli egyetemi helyszíneket kössön össze. Már az internet alapítóinak is fontos volt, hogy elkerüljék a szűk keresztmetszeteket, mivel ha egy szűk keresztmetszetet képző kapcsolat kiesik, nincsen olyan kapcsolat, ami ezt helyettesíteni tudná.

A hálózati problémák vizsgálatához az informatika a gráfelméletet hívja segítségül. A gráfokat csomópontok (tavacsák) és összekötő élek (csatornák) rendszereként definiálják. A gráfok segítségével az ember mindenféle hálózatot modellezhet, pl. egy közlekedési hálózatot vagy egy kommunikációs hálózatot. Számos algoritmust dolgoztak ki a hálózati problémák elemzésére. Egy ismert probléma a „hidak” megkeresése a gráfokon – a hidak éppen azok a szűk keresztmetszetek, amelyekről ebben a feladatban szó volt.



KÖSZÖNETNYÍLVÁNÍTÁS

TÁMOGATÓINK



KÖSZÖNETNYÍLVÁNÍTÁS

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

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