

Bebr[a]s

'24

2024. gada 1. kārtas uzdevumi ar atbildēm

11.-12. klase

INFORMATĪVIE ATBALSTĪTĀJI



Izglītības un zinātnes
ministrija

start(it)

www.startit.lv

Saturs

| | |
|--------------------------------------|----|
| Saturs..... | 1 |
| Atrodi dārgumus..... | 2 |
| Bruņurupuča zīmējumi..... | 4 |
| Balonu mašīna..... | 7 |
| Ēdiena karte..... | 9 |
| Uzmini domino kauliņu..... | 13 |
| Olu krāsošana..... | 15 |
| Lielākais virknes punktu skaits..... | 17 |
| Draugi..... | 21 |
| Izpēte..... | 23 |
| Ķieģeļu siena..... | 27 |
| Superbebras..... | 29 |
| Paslēptās bildes..... | 31 |
| Dzelzceļa tīkls..... | 34 |
| Palago..... | 37 |
| Turpini mainīt..... | 42 |

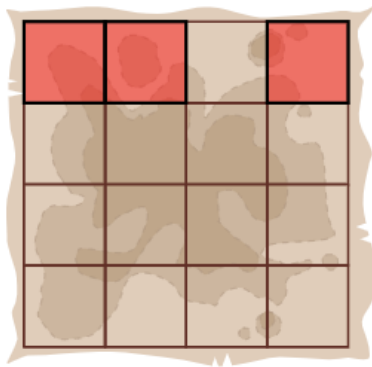
Atrodi dārgumus

Ungārija

Pirāts Pips meklē salā paslēptus dārgumus.

Pipam ir karte, kurā norādīts, kur atrodas dārgumi. Karte ir sadalīta 16 kvadrātos. Pips īpašā ierīcē var ievadīt jebkuru kvadrātu skaitu, un ierīce viņam pateiks, vai dārgums atrodas kādā no ievadītajiem kvadrātiem.

Piemēram, ja ierīce saka "jā", kad Pips ievada izceltos kvadrātus, tas nozīmē, ka dārgums atrodas vienā no šiem trim kvadrātiem:



Pips pēc iespējas ātrāk vēlas noskaidrot, kurā kvadrātā atrodas dārgumi.

Jautājums

Cik reizes Pipam ir jāizmanto ierīce, lai atrastu dārgumus pēc iespējas ātrāk?
Ievadi skaitli un nospied "saglabāt", kad esi pabeidzis

Skaidrojums

Pareiza atbilde ir 4.

Pipam vispirms ierīcē jāievada puse no reģioniem. Ir divi scenāriji:

A. Ja ierīce norāda, ka dārgums atrodas norādītajos reģionos, Pips sadala reģionus uz pusēm un ierīcē ievada vienu pusi.

B. Ja ierīce norāda, ka dārgums nav norādītajos reģionos, viņš sadala atlikušo daļu divās daļās un vienu no tām ievada ierīcē.

Pipa turpina šo procesu, līdz atrod reģionu, kurā atrodas dārgumus.

Piemēram, ja viņš ierīcē ievada A,B,C,D,E,F,G,H un ierīce saka "nav", viņš var ievadīt K,L,O,P (jebkuru atlikušo 4 reģionu grupu). Ja ierīce atkal atbild "nav", viņš turpina sadalīt atlikušos reģionus un ierīcē ievada jebkurus 2 atlikušos reģionus (piemēram, J,N). Ja ierīce atbild "nav", dārgums ir paslēpts vienā no atlikušajiem reģioniem - I vai M. Ja viņš ievada M un ierīce atbild "nav", viņš droši zina, ka dārgums ir I reģionā.

Lai pēc četriem jautājumiem būtu droši zināms, kurā laukumā atrodas dārgums, Pips var izmantot šādu stratēģiju: Viņš uzdod jautājumu par tieši pusi no laukumiem, kuros, pamatojoties uz iepriekšējiem jautājumiem, vēl varētu atrasties dārgums.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | A | B | C | D | A | B | C | D | A | B | C | D |
| E | F | G | H | E | F | G | H | E | F | G | H | E | F | G | H |
| I | J | K | L | I | J | K | L | I | J | K | L | I | J | K | L |
| M | N | O | P | M | N | O | P | M | N | O | P | M | N | O | P |

Pips nevar izmantot mazāk jautājumu. Ja viņš nesadala atlikušos kvadrātiņus divās vienādās daļās, tad dārgums varētu atrasties lielākajā daļā no tiem. Lai pajautātu par šo daļu, Pipam būtu jāuzdod tik daudz jautājumu, cik par vienu pusi.

Šī ir informātika

Metodi, ko Pips izmanto, lai atrastu dārgumu, datorzinātnē sauc par bināro meklēšanu. Termins "binārais" cēlies no latīņu valodas vārda "bis" (divreiz). Binārajā meklēšanā objekts tiek meklēts kopā, atkārtoti sadalot to uz pusēm, t. i., sadalot to divās daļās - no tā arī cēlies termins "binārais". Kopu var labi sadalīt uz pusēm, ja tajā esošie objekti ir sakārtoti, piemēram, pēc lieluma; tas attiecas uz jebkuru skaitļu kopu, arī uz citām lietām. Tad kopa satur vidējo objektu, un jūs varat salīdzināt vidējo objektu ar meklējamo objektu. Ja vidējais objekts nav tas, ko meklējat, jūs vismaz zināt, kurā pusē atrodas meklētais objekts, un atkārtoti meklējat šo pusi bināri. Šādā veidā meklējamo objektu var atrast ļoti ātri.

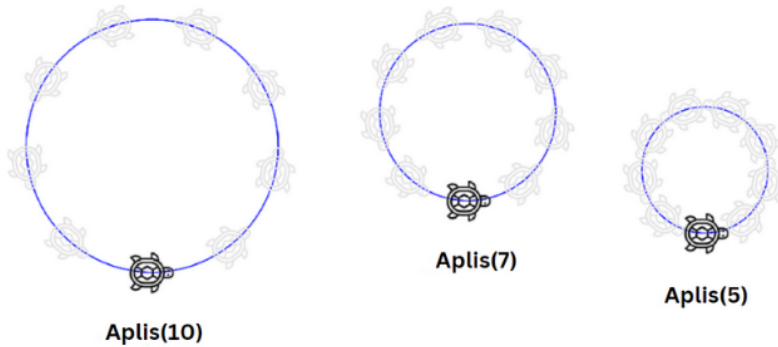
Ja ir 1 000 objektu, ir nepieciešami 10 meklēšanas soļi $2^9 < 1000 < 2^{10}$, bet 1 000 000 objektu gadījumā - 20 soļi. Vispārīgi var teikt, ka n objektu meklēšanai ir nepieciešami aptuveni $\log_2(n)$ soļi; logaritma funkcija ir logaritms pie bāzes 2. Tā kā binārā meklēšana ir tik ātra, to bieži datorprogrammās izmanto meklēšanai sakārtotu datu kopās.

Šajā uzdevumā meklēšanas telpa ir kvadrātukopa. Kvadrātus var sakārtot, tos numurējot no augšas uz leju un no kreisās puses uz labo. Tomēr šajā gadījumā var darboties arī, sadalot kopu uz pusēm kvadrātos, kuros vēl var atrast dārgumu. Tas tikai nedaudz apgrūtina atcerēšanos, kuru kvadrātu kopu vēl ir iespējams izmantot nākamajā meklēšanas posmā un kura atkal jāsadala uz pusēm.

Bruņurupuča zīmējumi

Somija

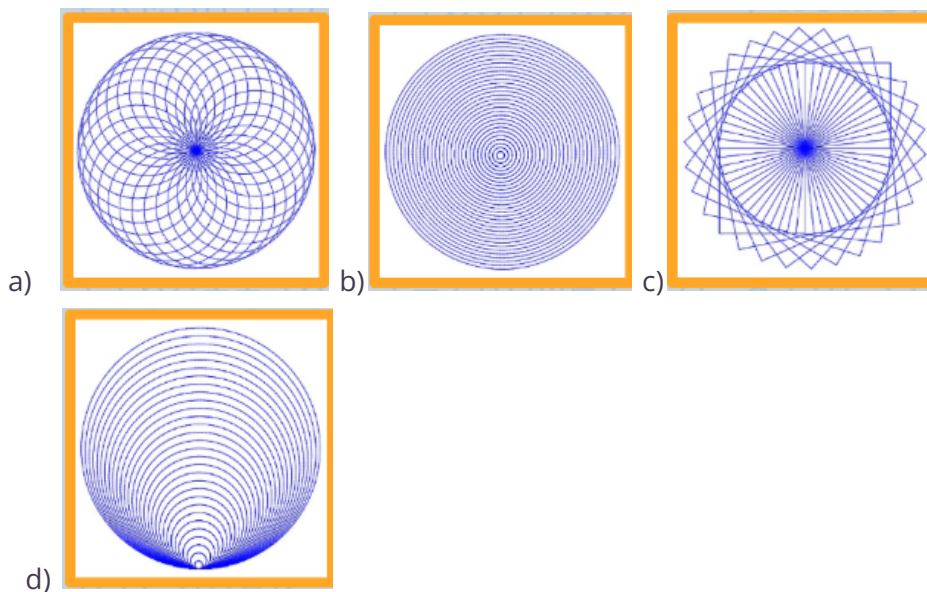
Robots, kurš ir bruņurupuča formā, uz zīmēšanas dēļa izveido zīmējumu atbilstoši instrukcijai. Kad tas saņem komandu zīmēt apli: "aplis(d)", tas uzzīmēs apli, kura izmērs mainīsies atkarībā no parametra d. Mainot parametru d, var tikt uzzīmēti dažāda izmēra apli (skat. attēlu).



Jautājums

Ja parametrs palielinās par 1 pēc katra uzzīmēta apļa, kādu rakstu robots izveidos, ja izpildot komandu zīmēt aplis, parametrs būs $d = 1, 3$?

Atbilžu varianti



Pareizā atbilde: D

Bruņurupuča robots izpildīja programmu 30 reizes. Pirmajā cilpā tas uzzīmēja ļoti mazu apli un atgriezās sākotnējā pozīcijā. Otrajā cilpā tas uzzīmēja lielāku apli un atkal atgriezās sākotnējā pozīcijā. Šiem diviem apliem jāatbilst robota sākuma pozīcijai. Pēc tam, kad visi 30 apli ir uzzīmēti, visiem tiem jābūt ar vienu kopīgu sākuma punktu, jo robota sākumpunkts ir nemainīgs. Tātad gala attēlā jābūt 30 dažāda izmēra apliem, un visiem apliem jābūt vienam kopīgam punktam, kā parādīts variantā D.

Kāpēc pārējās atbildes ir nepareizas:

A: visi uzzīmētie apli attēlā ir vienāda izmēra, taču tiem jābūt dažāda izmēra.

B: šiem apliem nav kopīga punkta, no kura robots sāktu tos zīmēt.

C: šis attēls nav veidots no apliem, bet no kvadrātiem (paceļoties no centra).

Tā ir informātika:

Dators darbojas, pamatojoties uz konkrētām datorprogrammas komandām vai instrukcijām. Viens no datorprogrammas elementiem ir funkcija. Funkcijas palīdz sadalīt programmas daļās, kurām ir konkrēti uzdevumi, padarot programmu organizētāku un vieglāk saprotamu. Funkcijai ir vairākas sastāvdaļas, tostarp nosaukums un parametri. Funkcijas nosaukums kalpo kā unikāls identifikators, nodrošinot, ka dators izpilda pareizo uzdevumu. Piemēram, šajā uzdevumā funkcija, kas atbild par aplu zīmēšanu, tiek saukta "aplis". Parametri nodrošina būtisku informāciju, lai funkcija varētu precīzi izpildīt savu uzdevumu. "Aplis" funkcijas gadījumā parametrs "d" norāda apļa diametru, kas jāuzzīmē, kad tiktu pielietota funkcija. Piemēram, izsaucot funkciju "aplis(10)", dators tiek instruēts zīmēt apli ar 10 vienību lielu diametru. Kā minēts iepriekš, funkcijas padara programmu organizētāku. Priekšrocības ir arī tādās, ka ir vieglāk veikt programmas izmaiņas, piemēram, labojot kļūdas. Turklāt funkcijas padara programmu atkārtoti izmantojamu. Šajā uzdevumā funkciju "aplis" var izmantot daudzas reizes, izmantojot ciklus. Papildus funkcijām šis uzdevums ir saistīts arī ar datorgrafiku. Datorgrafika datorzinātnē attiecas uz pētījumu un prakses jomu, kas ietver vizuālā satura veidošanu, manipulāciju un attēlošanu, izmantojot datorus. Tā aptver dažādas tehnikas, algoritmus un tehnoloģijas attēlu, animāciju un grafisko lietotāja saskarņu (GUI) ģenerēšanai un attēlošanai. Datorgrafika var būt dažādās formās, tostarp rastra grafika un vektorgrafika.

Šī ir algoritmiskā domāšana:

Šis uzdevums prasa pielietot abstrakciju, kas nozīmē atrast būtisko aprakstītajā situācijā. Piemēram, saprast, kā augošā parametra vērtība ietekmē dažāda izmēra aplus. Abstrakcija arī ietver nebūtisku lietu atstāšanu malā. Piemēram, ka apļa centram nav nozīmes un ka koeficients ietekmē apļa izmēru. Ir arī nepieciešama algoritmizācija, jo ir aprakstīts

algoritms, tādēļ ir jāsaprot, piemēram, ka pēc viena apla uzzīmēšanas nākamais aplis jāzīmē tajā pašā punktā tādā pat virzienā.

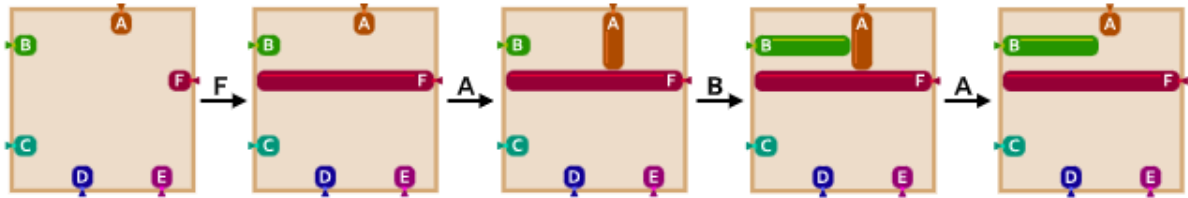
Balonu mašina

Vācija

Bebriem ir mašina, kas var radīt attēlus, piepūšot balonus kvadrātveida formā. Uz baloniem ir uzrakstīti burti A, B, C, D, E un F. Mašina nolasa burtus pa vienam. Kad tā nolasa burtu:

1. Ja balons, kas apzīmēts ar šo burtu, ir tukšs, tas tiek piepūsts, līdz tas pieskaras citam balonam vai formas pretējai malai.
2. Pretējā gadījumā balons, kas apzīmēts ar šo burtu, tiek izpūsts.

Piemēram, ja sākumā visi baloni ir izpūsti un mašina nolasa F, A, B un pēc tam A, tā rīkojas šādi:



Uzdevums

No sākuma katrs balons tiek izpūsts, un tad mašina nolasa deviņu burtu secību. Kā rezultāts tiek iegūts turpmāk redzamais attēls. Ievadiet šo deviņu burtu secību.

- a) BEBCACBDB
- b) BECBACBDB
- c) BEBCABCDB
- d) BECBABCDB

Skaidrojums

Iespējamās 4 pareizās atbildes ar deviņiem burtiem:

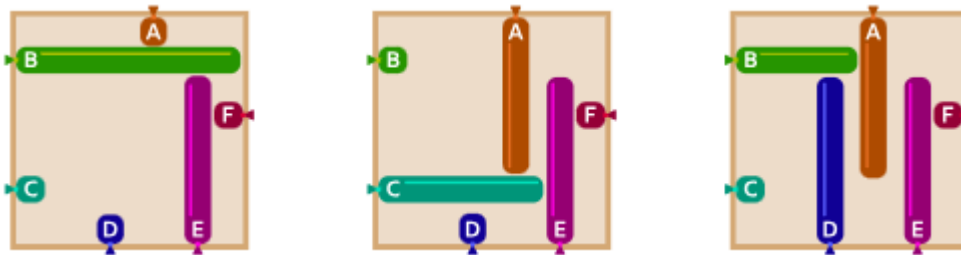
- B E B C A C B D B
- B E C B A C B D B
- B E B C A B C D B
- B E C B A B C D B

Mēs varam aplūkot detalizētu skaidrojumu vienai no pareizajām atbildēm: B E B C A C B D B. Attēlos zemāk tiek parādīts šis uzdevums, kurš ir sadalīts 3 daļās. Pirmajā attēlā redzams

stāvoklis pēc B E izpildes. Var redzēt, ka pirms E ir nepieciešams uzpūst B, lai B kļūtu par piemērotu šķērsli E uzpūšanai.

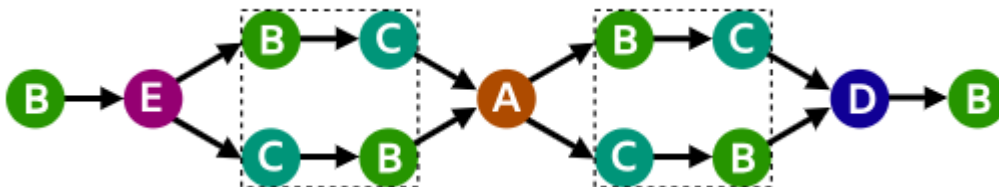
Otrajā attēlā parādīts stāvoklis pēc B E B C A. Balons C kalpo kā šķērslis A, tāpēc tas ir jāpiepūš pirms A. Turklāt balons B ir jāizpūš, pirms var piepūst A, citādi tas nerasniegs vajadzīgo garumu.

Trešajā attēlā redzams stāvoklis pēc B E B C A C B D. B kalpo kā šķērslis D, un C ir jāizpūš, pirms D tiek piepūsts.



Lai atrastu visas pareizās atbildes, ir lietderīgi problēmu attēlot, izmantojot virzienu grafiku. Burti attēlo balonus. Bultiņa no B uz E norāda, ka balons B ir jāpiepūš pirms balona E u.t.t. Ir divi gadījumi, kad B un C secība var mainīties (norādīts ar pārtrauktu līniju).

Katra pareizā atbilde (piepūšamo balonu secība) ir šī grafika topoloģiskā secība. Šajā gadījumā ir četri topoloģiskie sakārtojumi, tāpēc ir četri iespējamie pareizie risinājumi.



Piepūstiem baloniem jāparādās nepāra reizes \rightarrow A, D un E jāparādās nepāra reizes. Izpūstiem baloniem jāparādās pāra reizes (ieskaitot 0) \rightarrow B, C un F.

Šī ir informātika

Burtu secība ir datorprogramma, kas vada balonu mašīnu. Katrs burts strādā kā komanda, kas liek mašīnai piepūst balonu vai izlaist gaisu no tā. Tāpat kā lielākajā daļā šāda veida datorprogrammu, būtiska ir komandu secība. Piemēram, secība B E liek mašīnai izveidot citu attēlu nekā secība E B.

Atbildes skaidrojumā izmantotajam orientētajam grafam nav ciklu, tāpēc to sauc par orientētu aciklisku grafu (angliski - directed acyclic graph (DAG)). Jebkuram DAG ir vismaz viens topoloģiskais sakārtojums, un ir zināmi ātri algoritmi topoloģiskā sakārtojuma konstruēšanai. DAG ir daudz pielietojumu informātikā, piemēram, uzdevumu plānošanā.

Ēdiena karte

Itālija

Bebrs Niks ir paslēpis savas pārtikas krājumus zem 9 no 17 kokiem, kas ieskauj dīķi. Niks ir izveidojis karti un uzrakstījis uz kvadrātiem cik koku ar pārtikas krājumiem pieskaras šim kvadrātam.

Piemēram, kvadrāts uz kura ir uzrakstīts 3 nozīmē, ka pārtikas krājumi ir paslēpti tieši zem trim kokiem, kas pieskaras šim kvadrātam.



Tomēr šī ideja neliekas pareiza viņa draudzenei Bellai un viņai ir taisnība.

Viņa saka, ka Niks varēs noteikt tikai 7 no 9 kokiem, zem kuriem ir paslēpts ēdiens.

Uzdevums

Izvēlies šos 7 kokus

Skaidrojums

7 koki zem kuriem noteikti ir paslēpts ēdiens ir šie:



2 koki un viens tukšais (x) var būt identificēti, sākot analizēt no apakšas:

| | | | | | | |
|---|----|----|---|---|---|---|
| 🌳 | 🌳 | 🌳 | 🌳 | 🌳 | 🌳 | 🌳 |
| 🌳 | 2 | 2 | 1 | 1 | 2 | 🌳 |
| 🌳 | 2 | | | | 2 | 🌳 |
| 🌳 | 3 | 2 | 1 | | 2 | 🌳 |
| ✖ | ★🌳 | ★🌳 | 1 | | 1 | 🌳 |

Skatoties no apakšējā labā stūra uz kreiso, zem pirmā koka noteikti ir jābūt ēdienam, lai sakristu ar zemāko skaitli 1. Arī zem otrā koka ir jābūt ēdienam, lai sakristu ar skaitli 2. Tomēr, ja ēdiens būtu zem apakšējā kreisā koka, tad nebūtu iespējams apmierināt skaitli 2, ko aptver tik mazs lokš.

Tādā veidā ir iespējams noteikt visus kokus kreisajā pusē, zem kuriem atrodas ēdiens:

| | | | |
|----|----|----|----|
| 🌳 | ★🌳 | 🌳 | ★🌳 |
| ★🌳 | 2 | 2 | 1 |
| 🌳 | 2 | | |
| ★🌳 | 3 | 2 | 1 |
| ✖ | ★🌳 | ★🌳 | 1 |

Labajā dīķa pusē ir iespējami divi varianti:

| | | | |
|----|---|---|----|
| ★🌳 | 🌳 | 🌳 | ?🌳 |
| 1 | 1 | 2 | 🌳 |
| | | 2 | ★🌳 |
| | | 2 | ?🌳 |
| | | 1 | 🌳 |

| | | | |
|----|---|---|----|
| ★🌳 | 🌳 | 🌳 | 🌳 |
| 1 | 1 | 2 | ?🌳 |
| | | 2 | ★🌳 |
| | | 2 | 🌳 |
| | | 1 | ?🌳 |

Tieši tāpēc ir iespējams pievienot tikai vienu sarkano zvaigzni.

Kā vēl labāk noformulēt šo problēmu? Kad mēs esam noteikuši apakšējos kvadrātus, zem kuriem atrodas ēdiens, mēs varam iedot nosakumus burtu veidā arī pārējiem kvadrātiem:

| E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|
| D | 2 | 2 | 1 | 1 | 2 | L |
| C | 2 | | | | 2 | M |
| B | 3 | 2 | 1 | | 2 | N |
| A |   |   | 1 | | 1 | O |

Cipari zilajos kvadrātos satur "ierobežotājus". Skatoties pulksteņrādītāja virzienā no apakšas, pirmie 3 noteikumi (1,1,2) ir izpildīti, bet, lai izpildītu 3. ierobežojumu, ir jāizvēlas vēl tieši viens elements no kopas {A, B, C}, un tā tālāk. Tad šo uzdevumu var pārveidot šādi:

No {A, B, C} jāizvēlas tieši viens elements,
No {B, C, D} izvēlēties tieši 2 elementus,
No {C, D, E, F, G} izvēlēties tieši 2 elementus,
Nav {F, G, H} izvēlas tieši 2 elementus,
Nē {G, H, I} izvēlas tieši vienu elementu,
Nav {H, I, J} izvēlas tieši vienu elementu,
Nav {I, J, K, L, M} izvēlas tieši 2 elementus,
Nē {L, M, N} izvēlas tieši 2 elementus,
Nē {M, N, O} izvēlas tieši 2 elementus,
Nē {N, O} izvēlas tieši vienu elementu.

Apskatot visus variantus (no pirmās kopas nevar izvēlēties A, jo tad no otrās kopas būtu jāizvēlas B vai C; ja no pirmās kopas izvēlas B, tad no otrās kopas jāizvēlas D utt.), var pārbaudīt, ka ir divi risinājumi:

{B, D, F, H, K, M, N}, {B, D, F, H, L, M, O}.

Elementi, kurus noteikti var atzīmēt ar sarkano zvaigzni, ir B, D, F, H, M (kas veido abu iegūto kopu krustpunktu), kas saskan ar iepriekš iegūto.

Šī ir informātika

Šis uzdevums ir iedvesmots no slavenās loģikas puzzles - videospēles, kura ir pazīstama ar dažādiem nosaukumiem (Minesweeper, Flower Field). Pat tad, kad ir atklātas plaši laukumi, var rasties situācijas, kad nav pieejams pietiekami daudz informācijas, lai droši turpinātu spēli, tāpēc dažreiz ir jāpaļaujas uz veiksmi. Tieši šāds gadījums rodas mūsu uzdevumā: dīķa labo pusi nevar atrisināt tikai balstoties uz doto informāciju!

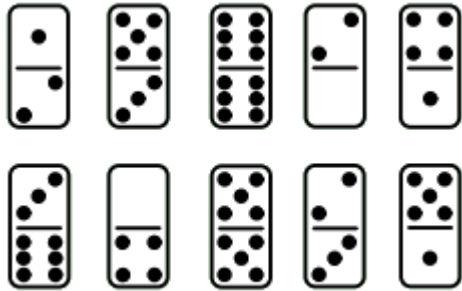
Ievērojiet, ka kopumā, ja ap dīķi ir n koku, tie atbilst 2^n iespējamām konfigurācijām, un tad parasti ir daudz vieglāk pārbaudīt, vai dotais risinājums vispār atbilst uzdevuma ierobežojumiem, nekā to atrast praktiski!

Skatoties uz šo uzdevumu mēs varam arī redzēt, cik bieži uzdevumus var risināt pakāpeniski t.i. atrisināt pa atsevišķām daļām. Nešaubīgi interesantākais aspekts no informātikas viedokļa ir problēmas formalizācija, izmantojot atbilstošu datu struktūru (kā mēs to darījām šim uzdevumam skaidrojuma pēdējā daļā), kas norāda, ka šī problēma ir skaitliski sarežģīta, kā arī tai ir vairāki risinājumi.

Uzmini domino kauliņu

Somija

Alise un Bobs spēlē spēli. Viņiem uz galda ir 10 domino kauliņi:



Bobs izvēlas slepenu domino figūru, kas zināma tikai viņam. Pēc tam Alise var uzdot Bobam jautājumus "jā" vai "nē", lai noskaidrotu, kuru domino kauliņu viņš izvēlējies. Katram jautājumam jābūt ar atbildi jā vai nē.

Alisei jautājumi jāveido tā, lai neatkarīgi no Boba atbildes viņai būtu iespēja pēc iespējas precīzāk noteikt, kurš ir slapenais domino kauliņš

Jautājums

Kuru jautājumu Alisei vajadzētu uzdot kā pirmo?

A. Vai punktiņu summa uz figūras ir lielāka vai vienāda ar 7?

B. Vai punktiņu skaits figūras galā, kurā ir lielāks punktu skaits, ir lielāks vai vienāds ar 4?

C. Vai punktiņu skaits figūras galā, kurā ir mazāks punktu skaits, ir lielāks vai vienāds ar 2?

D. Vai abos figūras galos ir vienāds punktu skaits?

Skaidrojums

Pareizā atbilde ir C, jo šis jautājums sadala kauliņu komplektu tieši divās grupās pa pieciem domino kauliņiem (pieciem domino kauliņiem viena puse ir tāda, kas lielāka vai vienāda ar 2, tāpēc pārējiem pieciem ir viena puse tāda, kas ir ar mazāk nekā 2 punktiem.), tāpēc

neatkarīgi no Boba atbildes viņam paliks piecas vēl derīgas domino kauliņu kandidāti. Šādi rīkojoties, Alise katru reizi izslēdz pusi no iespējam neatkarīgi no Boba atbildes.

Citos variantos kauliņu grupas nav sadalītas tik vienmērīgi, un viena no iespējamām atbildēm atstās vairāk nekā piecus domino kauliņus kā atlikušos kandidātus.

Nākamajā attēlā katram iespējamajam jautājumam parādīts, uz kuru domino kauliņu kopu norādītu "jā" (pelēkā krāsā) vai "nē" atbildi.

Tā ir informātika:

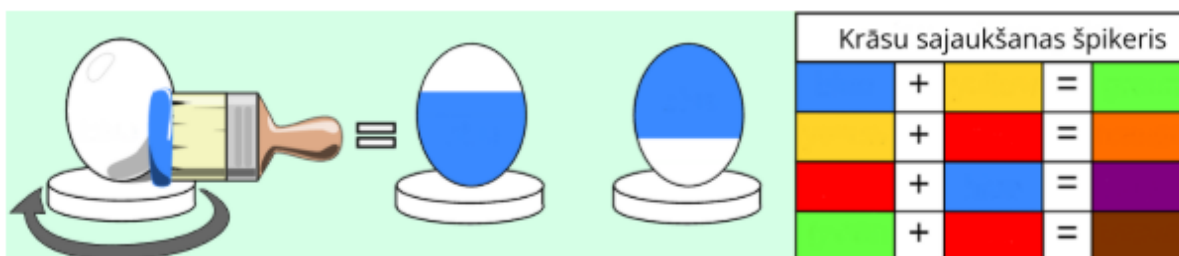
Datori glabā daudz datu, un tiem ir vajadzīgi ātri paņēmieni, kā atrast konkrētu informāciju. Tā vietā, lai aplūkotu katru datu vienību atsevišķi, mēs varam izmantot gudrākas metodes, ja datus labi sakārtojam. Piemēram, ja dati ir sakārtoti, mēs varam izmantot metodi, ko sauc par bināro meklēšanu. Tā darbojas šādi: mēs sākam ar vidējā elementa pārbaudi. Ja tas, ko meklējam, ir lielāks vai mazāks par meklēto, mēs varam ignorēt pusi datu un pievērsties tikai pārējiem. Mēs turpinām to darīt, katru reizi samazinot datu apjomu uz pusi. Tātad, ja mums ir 1 miljons elementu, mēs varam atrast meklēto tikai 20 soļos!

Šajā uzdevumā mums ir tikai jā/nē tipa jautājumi, tāpēc labākais pirmais uzdotais jautājums sadalītu iespējas divās daļās, līdzīgi kā pirmais solis binārajā meklēšanā. Ja mums būtu nepieciešams turpināt uzdot jautājumus, lai atrastu slepeno domino, mēs varētu izveidot visu iespējamo jautājumu plānu, kas līdzinās lēmumu koka izveidei.

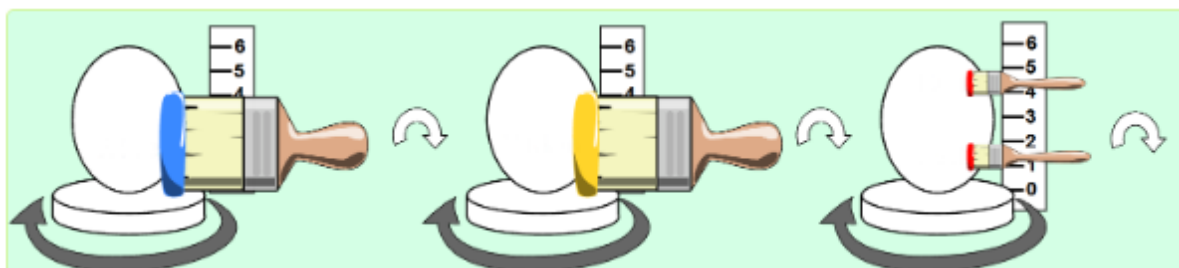
Olu krāsošana

Īrija

Aija grib nokrāsot olas Lieldienām. Kad viņa uzliek baltu olu uz krāsojamā aparāta, kas griežas ap savu asi un tur otas galu pie olas, olas tiek nokrāsotas ļoti ātri. Viņa parasti neizkustina otu, kamēr ola griežas aparātā, taču vienmēr apgriež olu otrādi pēc katras krāsas uzklāšanas, kā parādīts zemāk esošajā attēlā. Krāsas sajaucas kopā, kad tās pārklājas. Zemāk esošajā tabulā ir parādīti krāsu saikumi pēc divu dažādu krāsu sajaukšanas.

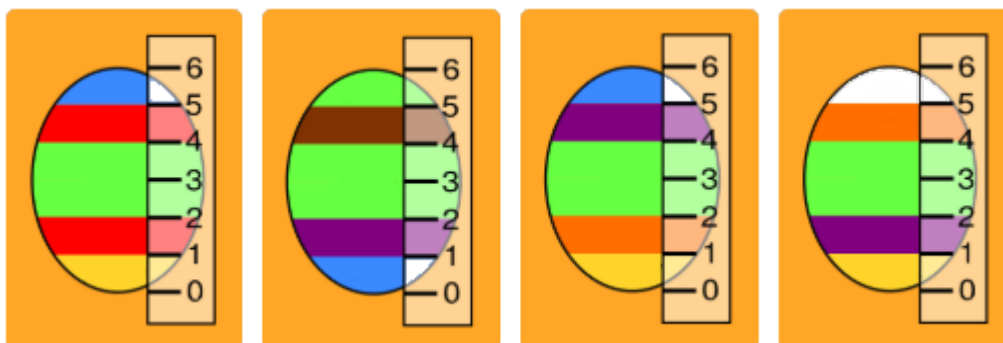


Aija nokrāso savu balto olu, izmantojot krāsu secību un otas platumu, kā parādīts nākamajā attēlā.



Jautājums

Kāda būs Aijas ola pēc krāsošanas?

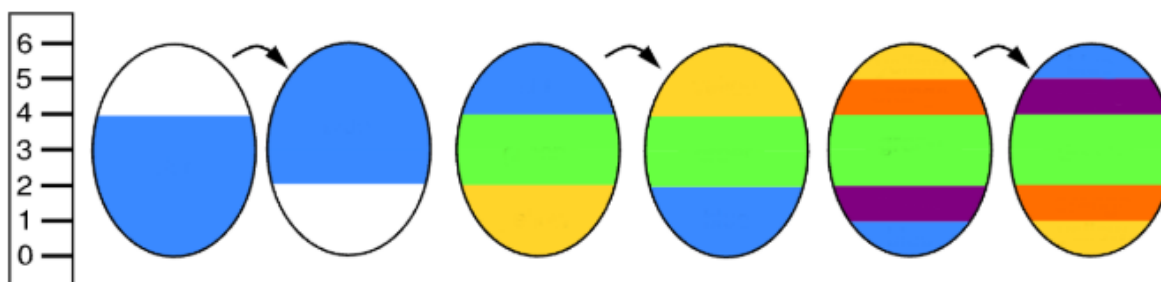


Skaidrojums

Pareizā atbilde ir C.

Šī uzdevuma risināšanā ir svarīgi pamanīt, kuras krāsas pārklāsies. Lielākas olas nokrāsos 4/6 olas, bet mazākas olas - 1/6 olas.

Zemāk ir ilustrācija, kurā redzamas olas krāsas pēc katras krāsošanas un apgriešanas darbības.



C ir pareizā atbilde, jo vidējā 1/3 ir zaļa krāsas josla, kurā sajaucas zilā un dzeltenā krāsa. Abas mazākās olas ir iestatītas dažādos augstumos: tur, kur sajaucas sarkanā un zilā krāsa, parādās violeta josla, bet tur, kur sajaucas dzeltenā un sarkanā krāsa, parādās oranža josla.

A nav pareiza, jo sarkanā krāsa nesajaucas ar zem tās esošo krāsu.

B nav pareiza, jo visa ola būtu jānokrāso zilā krāsā, lai vienā galā būtu zila, bet otrā - zaļa, bet patiesībā tikai 2/3 olas ir nokrāsota zilā krāsā.

D nav pareiza, jo vienīgais veids, kā vienā galā iegūt baltu krāsu, ir neapgriezt olu starp zilo un dzelteni otru, bet mēs zinām, ka olu apgriež pēc katras krāsošanas darbības.

Šī ir informātika

Šajā uzdevumā ola iziet cauri krāsošanas konveieram. To var uzskatīt par iteratīvu algoritmu, kas apstrādā savus "datus", proti, olas. Katrā iterācijā olu a) krāsošanas laikā pagriež par 360 grādiem un b) apgriež otrādi. Turklāt būtu vajadzīga datu struktūra, lai uzglabātu apstrādājamās olas strīpiņu pašreizējo stāvokli. Šajā gadījumā piemērota datu struktūra varētu būt viendimensiju masīvs, kas sastāv no sešām šūnām: pa vienai katrai joslai. Krāsošanas operācijas laikā šūnu saturs būtu jāpārbauda un jā sajauc ar otiņas krāsu, un iegūtā krāsa aizvietotu veco krāsu šūnā. Iegūto krāsu aprēķina saskaņā ar sniegtajiem rādītājiem. Ja katru olas joslu uzskata par patstāvīgi krāsojamu olas zonu, tad laba datu struktūra būtu sistoliskais masīvs: šādā sistoliskajā masīvā dati iziet cauri viens otram līdzīgu skaitļošanas soļu montāžas līnijai, katram no tiem nedaudz pārveidojot datus, līdz beigās parādās pilnībā apstrādāti dati.

Lielākais virknes punktu skaits

Slovākija

Spēlē ir jāredīgē secība, kas izveidotas no kartītēm ar burtiem A, B un C. Spēlētāji cenšas iegūt pēc iespējas augstāku punktu skaitu, nomainot dažus no burtiem.

Par katru 2 vienādu burtu grupu rindā spēlētājs saņem 2 punktus. Par katru 3 vienādu burtu grupu spēlētājs saņem 3 punktus. Un tā tālāk. Piemēram, piecu pelēku kāršu virkne iegūst 0 punktu, jo blakus nav neviena vienāda burta. Taču ir pieejamas divas sarkanas kārtis, lai uzlabotu rezultātu.



Pirmo C var aizklāt ar sarkanu A karti, bet otro C - ar sarkanu B karti, kā parādīts attēlā.



Uzdevums

Tad divu "A" un divu "B" punktu grupai rezultāts ir $2 + 2 = 4$ punkti. Pirmā B ir pati par sevi, tāpēc tā neieskaita punktu skaitu.

Velciet sarkanās kārtis B, B un C, lai nosegtu jebkurus trīs burtus un izveidotu secību ar visaugstāko iespējamo punktu skaitu.



Skaidrojums

Zemāk ir parādīts risinājums:



Šajā rindā jūs iegūstat $2 + 2 + 4 + 3 = 11$ punktus; no kreisās uz labo 2 B, 2 C, 4 B and 3 A vienus pēc otra. Vairak punktus iegūt jūs nevarat.

- Šajā rindā ir 12 burti, tāpēc vairāk par par 12 punktiem jūs iegūt nevarat;
- Tomēr, burts A kreisajā malā nevar piederēt grupai, tapēc ka jums nav neviena A, lai samainītu uz B
- Visi pārējie 11 burti pieder grupai un katrs sastāda pa 1 punktam no kopejā rezultāta.
- Katrs burts var dot tikai maksimums 1 punktu.

Vai ir kādi citi veidi kā iegūt 11 punktus? Lai iegūtu maksimāli lielu punktu skaitu, jums ir jāizkārto burti B, B un C vienā līnijā, kurā izveidojas jauna burtu grupa, bet neviena veco burtu grupa netiek samazināta:

- Ja burts tiek pievienots vienai grupai, jums jāpieskaita 1 punkts.
- Tomēr, ja burts ir novietots blakus vienam burtam un izveido jaunu grupu vai šis burts savienojas ar esošu burtu grupu, jūs varat pieskaitīt 2 punktus.
- Ja burts ir novietots starp diviem citiem brīviem burtiem, tad veidojas 3 burtu grupa un tiek pieskaitīti 3 punkti.

Piemērs jums sākotnēji dod 5 punktus. Nav iespējams iegūt 3 punktus ar B, B vai C. Lai iegūtu 11 punktus, B, B un C jānovieto tādā veidā, lai start tiem katru reizi veidojas jauna grupa un to var izdarīt tikai ar s B un diviem C:



Ir tikai 1 veids kā izveidot 3 jaunas grupas ar B, B un C, nepazaudējot esošo grupu lielumu. Tikai pēdējā gadījumā jūs iegūstat 11 punktus, bet par pārējiem 10 punktu. Šeit ir šie 4 gadījumi:



Šī ir informātika

Šis piemērs skaidri parāda, kā datorspeciālisti rīkojas, meklējot optimālako risinājumu vai metodi risinājuma atrašanai.

Ideja ir atrast tādu B, B un C burtu izvietojumu, kurā iegūst visvairāk punktu. Pirmā pieeja varētu būt vienkārši ģenerēt visas iespējas, kurās trīs burti sērijā tiek aizstāti ar B, B un C. Divpadsmit burtu sērijai ir $12 * 11 * 10 = 1320$ iespējas, un katrai no tām jāaprēķina un jāsalīdzina punkti, lai noteiktu labāko risinājumu.

Risinājumu skaitu var ievērojami samazināt, ja ņem vērā, ka B, B un C saņem punktus tikai tad, ja tie atrodas blakus vienādiem burtiem. Ja ņem vērā arī to, ka esošās grupas nedrīkst samazināt, risinājumu skaits vēl vairāk samazinās. Papildus, ja ņem vērā arī to, ka atsevišķu burtu savienošana dod vairāk punktu nekā esošo grupu palielināšana, tad paliek tikai viens iespējamais izvietojums.

Iepriekš minētās 1320 iespējas nav tik daudz, lai dators tās varētu aprēķināt, Tomēr datorzinātnē vienmēr tiek uzdots jautājums, kā mainīsies problēmas atrisināšanai nepieciešamās pūles, pieaugot problēmas lielumam. Ar garākām rindām un vairāk mainīgajiem burtiem dažādu risinājuma iespējas būtu attiecīgi vairāk. Ir vērts rūpīgi analizēt problēmu un izstrādāt efektīvāko metodi.

No informātikas viedokļa tā ir optimizācijas problēma. Informātikas jomā optimizācijai ir būtiska nozīme. Runa ir par to, lai kaut ko padarītu pēc iespējas efektīvāku vai funkcionālāku ņemot vērā noteiktus ierobežojumus. Optimizācijas mērķis ir atrast labāko no visiem iespējamajiem risinājumiem. Piemēram, kad mēs runājam par koda optimizāciju kompilatorā, mēs runājam par koda pārveidošanas procesu, lai padarītu to efektīvāku un mazāku, nemainot tā rezultātus un neveidojot blakusparādības. Tas ir svarīgi, jo palīdz programmām darboties ātrāk un izmantot mazāk resursu. Piemēram, datu saspiešana ir viens no optimizācijas optimizācijas piemēriem. Tas ietver informācijas kodēšanu, izmantojot mazāk bitu nekā sākotnēji, lai ietaupītu glabāšanas vietu vai samazinātu pārsūtāmo datu apjomu. Šajā uzdevumā, mēģinot optimāli sakārtot burtus "BBC", mēs būtībā darām to pašu - cenšamies atrast visefektīvāko sakārtojumu, kas ļaus sasniegt mūsu mērķi, t. i., izveidot secību, kas dod mums visvairāk punktu. Lai to izdarītu, mēs saglabājam burtus "BBC", lai

palielinātu to pašu burtu secības garumu vai lai izveidotu jaunu apakšvirskni. Visām šādā veidā izveidotajām sekvencēm aprēķinām to punktu skaitu. Pēc tam varam izvēlamies virskni ar vislabāko rezultātu.

Draugi

Lietuva

Klasē ir septiņi bebri: Aleks, Baiba, Kira, Dāvis, Ēriks, Freds un Džeina. Daži no bebbriem viens otru pazīst, bet daži nē.

Tabulā zemāk starp diviem bebbriem ir atzīme tad, ja viņi viens otru pazīst.

| | Aleks | Baiba | Kira | Dāvis | Ēriks | Freds | Džeina |
|--------|-------|-------|------|-------|-------|-------|--------|
| Aleks | ■ | ✓ | ✓ | | | | |
| Baiba | ✓ | ■ | | | | | ✓ |
| Kira | ✓ | | ■ | | ✓ | | |
| Dāvis | | | | ■ | | ✓ | |
| Ēriks | | | ✓ | | ■ | | |
| Freds | | | | ✓ | | ■ | |
| Džeina | | ✓ | | | | | ■ |

Katrs bebris saņēma no citiem atšķirīgu ziņu no kāda bebra ārpus klases un dalījās ar to ar visiem savas klases bebbriem, kurus pazina. Ikreiz, kad bebris saņem jaunu ziņu no kāda ārpus klases, viņš nekavējoties dalās tajā ar visiem klasē pazīstamajiem bebbriem.

Jautājums

Kurš(-i) šīs klases bebris(-iem) kopumā saņēma vismazāk ziņojumu?

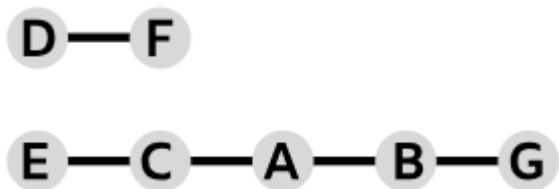


Skaidrojums

Pareizā atbilde ir Dāvis un Freds.

Mēs varam izmantot diagrammu, lai parādītu savstarpējās attiecības starp šīs klases bebbriem. Šajā diagrammā katrs bebris ir attēlots ar apli, un, ja divi bebbri viens otru pazīst, tad viņu apļus savieno līnija. Dāvis pazīst tikai Fredu, un Freds pazīst tikai Dāvi, tāpēc mēs

savienojam viņu apļus kopā un nesavienojam tos ar citiem apļiem. Ēriks pazīst tikai Kiru, tāpēc mēs savienojam viņu apļus. Baiba pazīst arī Aleku, tāpēc mēs savienojam viņu apļus. Aleks pazīst arī Baibu, tāpēc mēs savienojam viņu apļus. Visbeidzot, Baiba pazīst arī Džeinu, tāpēc mēs savienojam viņu apļus.



Tagad no diagrammas redzams, ka katrs no Dāvja un Freda var saņemt tikai 2 ziņas, bet katrs no Ērika, Kiras, Aleksa, Baibas un Džeinas saņems 5 ziņas. Tāpēc Dāvis un Freds saņems vismazāk ziņu.

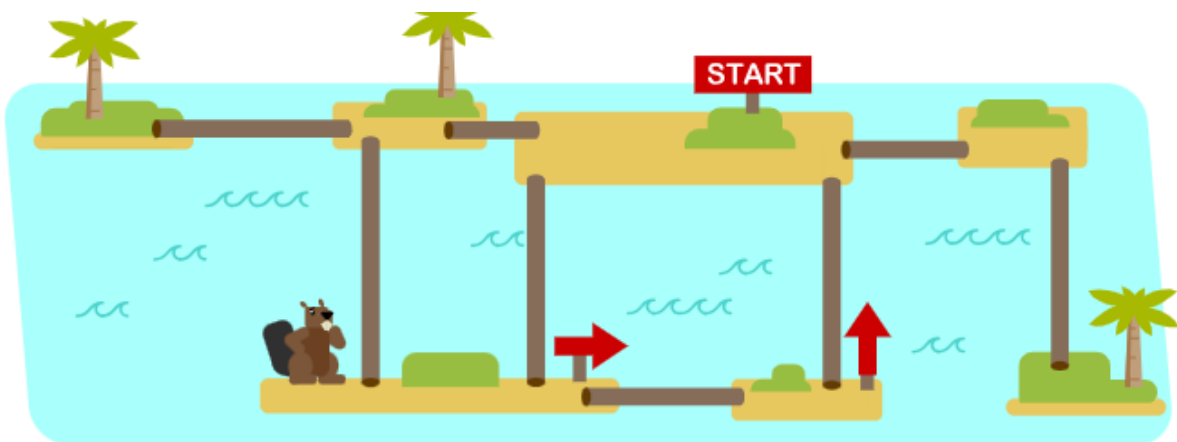
Šī ir informātika

Grafi ir svarīgs abstrakcijas rīks datorikā. Šajā uzdevumā grafs tiek attēlots kā saistību matrica. Bebru draudzības problēma ir saistīta ar grafu sakarības problēmu. Grafu teorijā un datorikā saistību matrica ir kvadrātveida matrica, ko izmanto, lai attēlotu grafu. Matricas elementi norāda, vai virsotņu pāri grafā ir savienoti ar šķauni vai nē. Šis grafs ir neorientēts (t.i., visas šķautnes ir divvirzienu), jo saistību matrica ir simetriska. Saistību matrica var tikt izmantota kā datu struktūra, lai attēlotu grafus datorprogrammās, kas paredzētas darbībām ar tiem. Galvenā alternatīva šai datu struktūrai, ko izmanto arī šajā pielietojumā, ir kaimiņu saraksti.

Izpēte

Čehija

Bebrs Harijs pēta ar balķiem savienotu salu tīklu, kāds parādīts zemāk attēlā. Viņš vēlas apmeklēt visas parādītās salas. No jebkuras salas viņš var redzēt blakus esošo salu un noteikt, vai uz tās ir zīme.



Iepriekš redzamajā attēlā viņš ir sācis no patvaļīgi izvēlētas salas un sasniedzis vēl divas salas, izmantojot šādu stratēģiju:

Uzdevums

Aizpildiet tukšās vietas ar pareizajiem norādījumiem, nodrošinot ka Harijs apmeklē visas salas.

levieto pareizos teksta blokus tukšajās vietās, kas sakrīt ar teksta bloku formu. Nospiediet "saglabāt", kad esat pabiedzis

Ejiet uz pirmo salu un novietojiet START zīmi.

Katru reizi, kad apmeklējat jaunu salu:

- Ja uz salas nav zīme, novietojat bultiņu, kura norāda uz salu, no kurienes atnācāt
- Ja salai ir blakus cit sala, uz kuras , tad dodaties uz to salu
- Ja salai ir blakus cit sala, uz kuras
- Ja uz salas ir bultiņa, dodaties uz salu
- Vai paliekat šeit. Jūs esat bijuši uz visām salām vismaz vienu reizi

nav zīme
ir bultiņas zīme
ir START zīme

no kuras jūs atnācāt
uz kuru norāda bultiņa

Skaidrojums

Pareizā atbilde ir:

Ejiet uz pirmo salu un novietojiet START zīmi.

Katru reizi, kad apmeklējat jaunu salu:

- Ja uz salas nav zīme, novietojat bultiņu, kura norāda uz salu, no kurienes atnācāt
- Ja salai ir blakus cit sala, uz kuras **nav zīme**, tad dodaties uz to salu
- Ja salai ir blakus cit sala, uz kuras **nav zīme**, tad dodaties uz to salu
- Ja uz salas ir bultiņa, dodaties uz salu **no kuras jūs atnācāt**
- Vai paliekat šeit. Jūs esat bijuši uz visām salām vismaz vienu reizi

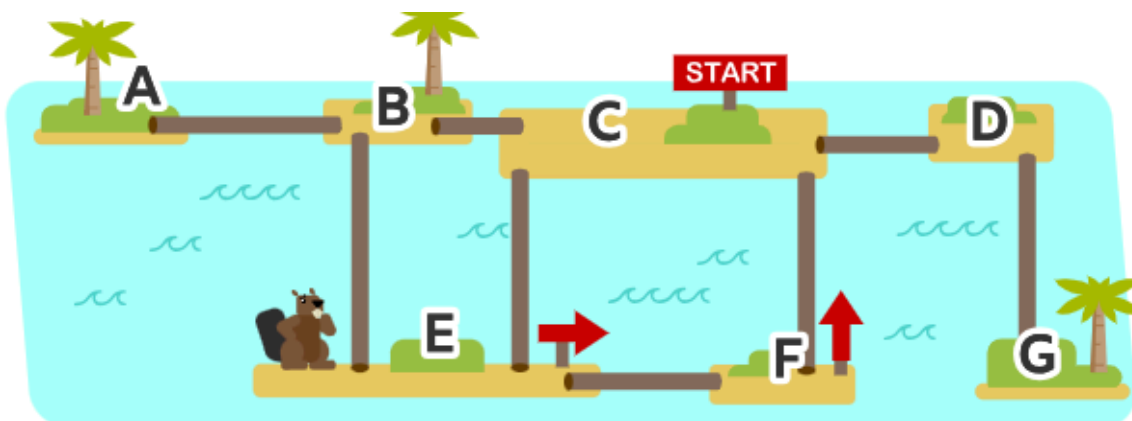
nav zīme
ir bultiņas zīme
ir START zīme

no kuras jūs atnācāt
uz kuru norāda bultiņa

Pirmais tukšais laukums:

Uz salām, kurās Harijs jau ir bijis, ir zīme, bet uz salām, kurās viņš vēl nav bijis, zīmes nav (jo viņš uzstāda zīmi, kad dodas uz salu, uz kuras tās vēl nav).

Pareizi: a) nav zīmes - dodieties uz salām, kurās viņš vēl nav bijis. Piemēram, turpmāk parādītajā situācijā viņam būtu jādodas uz blakus esošo salu bez norādes, B. Ja viņš priekšroku dotu salām, kurām jau ir norādes, viņš pārvietotos starp salām C, E un F, bet nekad nedotos uz salām, uz kurās vēl nav bijis.



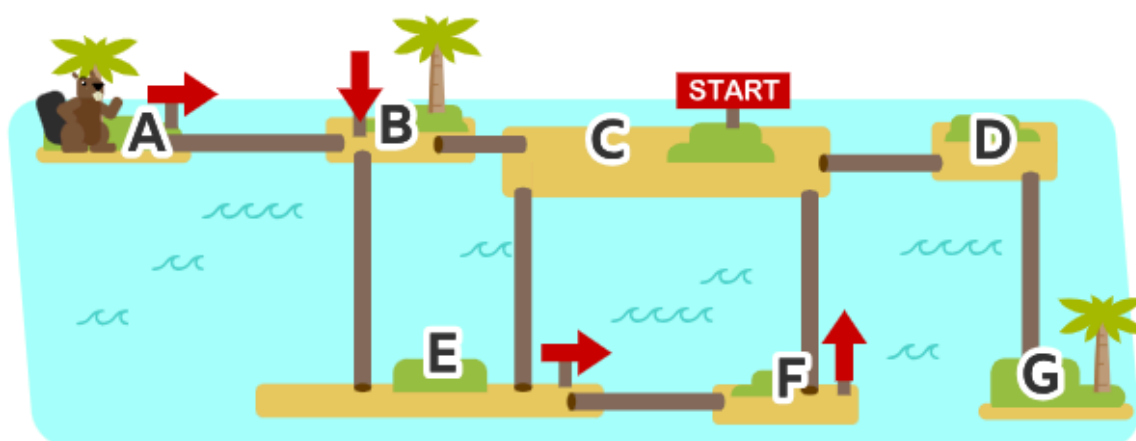
Nepareizi: c) sākuma zīme - algoritma otrās iterācijas laikā Harijs atkal atradīsies sākuma salā, un algoritms nepareizi izpildītu pēdējo "pretējā gadījumā" instrukciju.

Otrs tukšais laukums:

Pareizi: b) sala, uz kuru norāda bultiņa - atpakaļceļš uz salu, kurai vēl ir neizpētītas kaimiņu salas.

Nepareizi: a) sala, no kuras nupat nācis, - var sanākt visu laiku pārvietoties starp divām salām, kurām nav kaimiņu salas bez zīmēm.

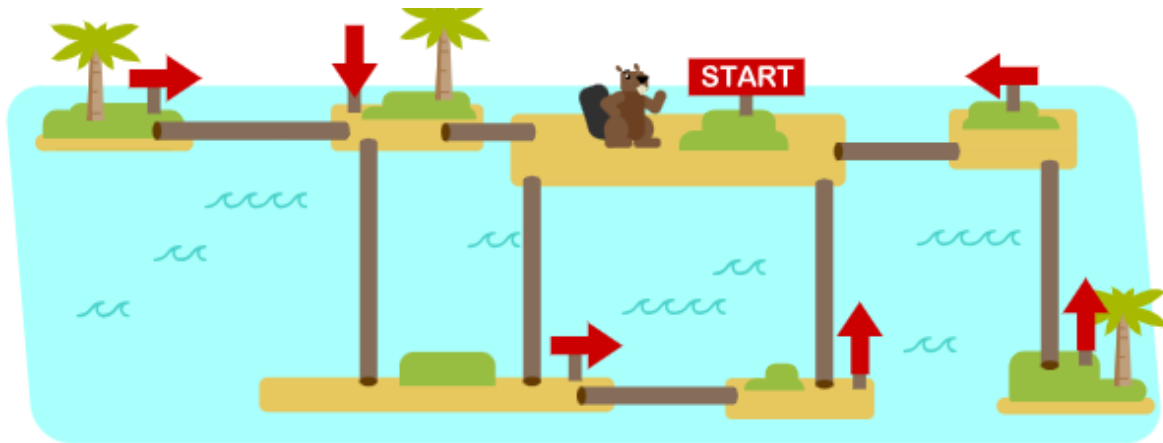
Piemēram, tālāk redzamajā situācijā viņš atrodas uz salas A, kurai vairs nav neizpētītu kaimiņu salu. Viņam nav citas izvēles, kā vien doties uz salu B. Nākamajā solī, ja viņš dotos atpakaļ uz salu, no kuras tikko bija nācis, viņš atgrieztos uz A, tad uz B, tad uz A un nekad neatstātu šo salu pāri. Tādējādi viņam jāseko bultiņām uz zīmēm, lai dotos uz E, F, C un no turienes uz salu D, kurā viņš vēl nav bijis.



Šī ir informātika

Savstarpēji savienotu salu tīklu var modelēt ar grafu palīdzību. Grafs ir datu struktūra. Tas sastāv no virsotnēm (šeit tās atbilst salām) un šķautnēm (šeit tās atbilst koku stumbriem), kas savieno virsotnes. Daudzas sistēmas realitātē var attēlot ar grafu palīdzību. Piemēram, cilvēkus saista draudzība, dzelzceļa stacijas savieno dzelzceļa līnijas, bet iespējamās pozīcijas šaha spēlē - derīgi gājieni. Dažreiz ir svarīgi pilnībā izpētīt grafu, piemēram, ja vēlaties atklāt noteiktas īpašības. Īpaši tas attiecas uz lieliem grafikiem ar daudz virsotnēm un šķautnēm.

Šajā uzdevumā izmantoto stratēģiju datorzinātnē sauc par meklēšanu dziļumā jeb dziļkursiju. Kad algoritms ir pilnībā izpildīts, bebrs ir atgriezies uz salas, no kuras viņš sāka, un šajā piemērā salu grupa izskatās šādi.



Algoritms, kas bija jāizpilda, tika pierakstīts pseidokodā. Pseidokods ir dabiskās valodas un programmēšanas valodas sajaukums. Instrukcijas ir formulētas tā, lai tās būtu vispārēji saprotamas. Tiek izmantotas atkāpes (līdzīgi kā programmēšanas valodā Python), lai parādītu, kuras instrukcijas pieder cita citai.

Ķieģeļu siena

Somija

Bobs uzcēla sienu, izmantojot trīs veidu ķieģeļus: A, B un C.

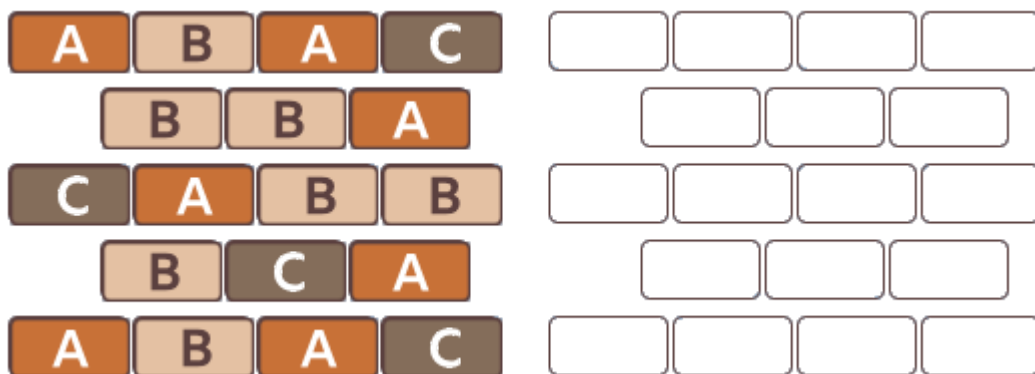
Diemžēl Bobs uzcēla sienu nepareizā vietā, un viņam tā ir jāpārceļ citur. Viņš to dara pa vienam ķieģelim pēc kārtas šādi:

- No sākotnējās sienas noņem jebkuru ķieģeli, tieši virs kura nav neviena ķieģeļa.
- Novieto šo ķieģeli jaunajā sienā vai nu uz zemes, vai uz cita ķieģeļa, bet ne zemāk par jau esošajiem ķieģeļiem.

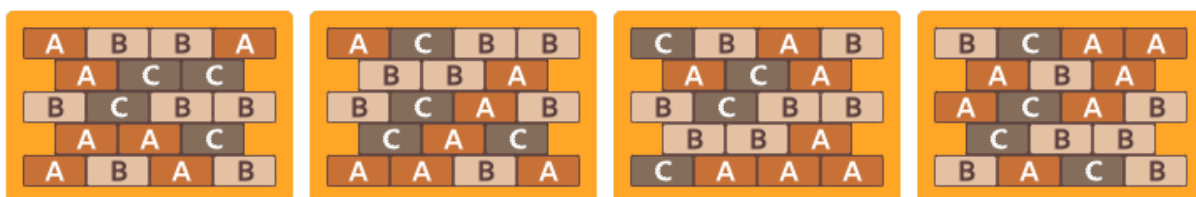
Kamēr viņš ievēro šos noteikumus, Bobam ir vienalga, kādā secībā dažādu veidu ķieģeļi tiek pārnesti no sākotnējās sienas uz jauno. Nākamajā attēlā pa kreisi redzama Boba sākotnējā siena. Jūs varat klikšķināt un vilkt ķieģeļus, lai uzbūvētu jaunu sienu labajā pusē, taču ņemiet vērā, ka netiks pārbaudīts, vai jūs esat pareizi ievērojis noteikumus. Lai sāktu no jauna, noklikšķiniet uz pogas RESET.

Jautājums

Izvēlies visas sienas, ko Bobs var uzbūvēt, sekojot šiem noteikumiem!



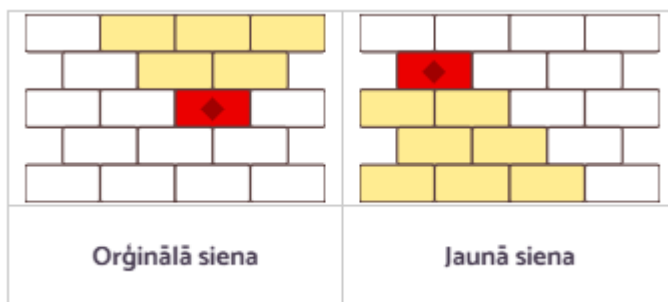
Atbilžu varianti



Skaidrojums

Ir iespējamas tikai sienas A) un C).

Aplūkojiet diagrammu kreisajā pusē, kurā redzams ķieģelis, kas sākotnējā sienā ir iezīmēts sarkanā krāsā. Tas ir otrādi apgriezta trīsstūra gals, kas sniedzas līdz augšējai rindai - dzeltenie izceltie ķieģeļi ir tie, kas noteikti jānoņem, pirms var noņemt sarkano ķieģeli. Līdzīgi, diagrammā pa labi ir attēlots sarkans ķieģelis, kas ir uz augšu vērsta trīsstūra gals (nogriezts šajā gadījumā) - pirms sarkanā ķieģeļa novietošanas ir jānovieto visi dzeltenie ķieģeļi.



Šī ir informātika

Uzdevums ir saistīts ar savstarpējo atkarības (jeb prioritātes ierobežojumu) analīzi starp objektiem. Tā ir klasiska datorzinātnes problēma. Attiecības starp objektiem - šajā Bebras uzdevumā - ķieģeļiem - parasti tiek izteiktas izmantojot attiecību grafiku. Derīga datu apstrādes secība - šajā uzdevumā tā ir secība, kādā ķieģeļus var noņemt un novietot - atbilst objektu topoloģiskai sakārtošanai.

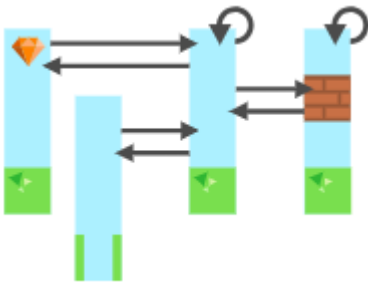
Superbebras

Vācija

Datorspēlē Superbebras spēles fons sastāv no daudzām vertikāliem blokiem. Dators pastāvīgi pievieno jaunu bloku rindas labajā pusē un vienlaikus noņem bloku kreisajā pusē. Tādējādi dators rada ilūziju, ka attēls kustās.



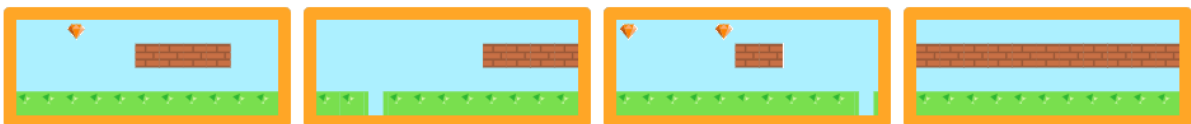
Dators izvēlas jaunu bloku, ko pievienot, izmantojot zemāk redzamo shēmu. Tas aplūko iepriekšējo plāksnīti, pārbauda bultiņas, kas nāk no šīs plāksnītes, un pēc nejaušības principa izvēlas vienu no blokiem, uz kuru norāda bultiņa.



Piemēram, pēc bloka  dators var izvēlēties nākamo bloku vai nu  vai šo  bloku.

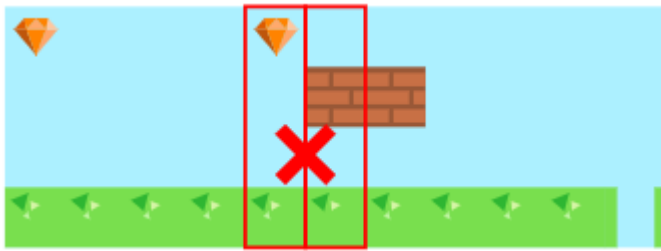
Jautājums

Viens no šiem variantiem nevar būt Superbebras fons. Kurš ir tas?



Skaidrojums

Atbildē C parādītais fons nav pilnībā nesaskan ar pirmstam parādīto shēmu.



Blokam ar dimantu var sekot tikai, tikai viens specifisks bloks, kas šajā gadījumā nav tā noticis

Ir vairāki veidi, kā atrast šī uzdevuma risinājumu. Vienkāršākā stratēģija ir izmēģināt katru doto fona attēlu un pārbaudīt katru bloku pēc shēmas, lai pārlicinātos, vai tas ir pareizs. Ātrākais veids ir šāds: Jūs pārbaudāt shēmu pēc blokiem, sākot ar tiem, no kuriem iziet tikai viena bulta. Šīs bultas uzskatiet par ierobežojumiem. $A \rightarrow B$ nozīmē, ka "blokam A jāseko blokam B". Tad pārbaudiet, vai šis ierobežojums ir spēkā katram atbilstošam variantam.

Šī ir informātika

Dažās datorspēlēs, piemēram, bezgalīgās skriešanas spēlēs, fons pārvietojas horizontāli, radot ilūziju, ka spēlētājs pārvietojas uz priekšu pa iedomātu pasauli. Dažkārt fons nav nemainīgs attēls, bet to automātiski izveido dators. To sauc par procedurālo ģenerēšanu. Parasti mazāki elementi netiek nejauši apvienoti, lai radītu visdažādākos fonus.

Uzdevumā šie noteikumi tiek attēloti ar flīžu diagrammu, un šādas diagrammas, kurās elementi ir savienoti ar bultiņām, sauc par virzītiem grafiem, tos sauc arī par grafiem. Elementi tiek saukti par mezgliem, bet bultas ir norādītās malas. Virzītos grafus izmanto visiem modelēšanas veidiem.

Paslēptās bildes

Austrālija

Leo ir izgudrojis jaunu attēlu atšifrēšanas metodi, izmantojot objektus (H) (horizontāli) un (V) (vertikāli).

Attēls būtībā ir taisnstūris, kas sadalīts kvadrātveida šūnu rindās un kolonnās. Šūnas sauc par pikseliem, un katrs pikselis ir iekrāsots kādā krāsā.

Katrā objekta (H) izmantošanas reizē:

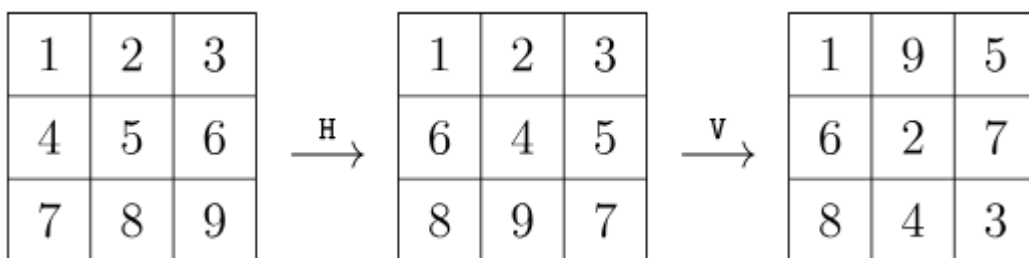
- Katrs 1. rindas pikselis paliek savā vietā (t. i., tas nepārvietojas).
- Katrs 2. rindas pikselis pārvietojas vienu vietu pa labi.
- Katrs 3. rindas pikselis pārvietojas divas vietas pa labi.
- Katrs pikselis n-tajā rindā pārvietojas (n - 1) vietu pa labi.

Ja kādas rindas pikseli tiek pārvietoti ārpus attēla labās malas, tie tiek saglabāti tajā pašā secībā un kā grupa pārvietoti uz brīvo vietu rindas kreisajā galā.

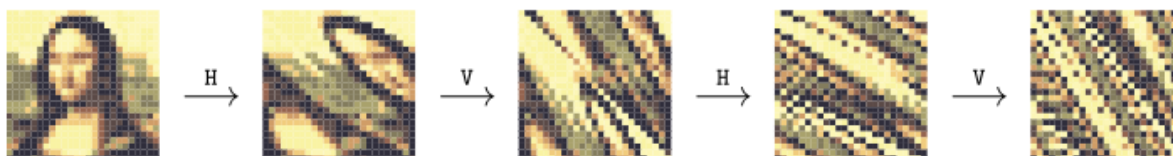
Līdzīgi katrā objekta (V) izmantošanas reizē:

- Katrs pikselis n-tajā kolonnā pārvietojas par (n - 1) vietu uz leju, un pikseli, kas tiek pārvietoti ārpus attēla apakšējās malas, tiek pārvietoti uz augšu.

Šeit ir piemērs ar 3*3 attēlu krāsām, kas apzīmētas ar 1 līdz 9:



Šeit ir redzams, kā secība HVHVHVHVHVHV var tikt izmantota, lai šifrētu 25 *25 Mona Lizas attēlu:



Jautājums

Leo šifrē šādu 1000×1000 attēlu, izmantojot (V) un pēc tam (H).



Kurš no šiem variantiem atbilst vēlamajam rezultātam?



Skaidrojums

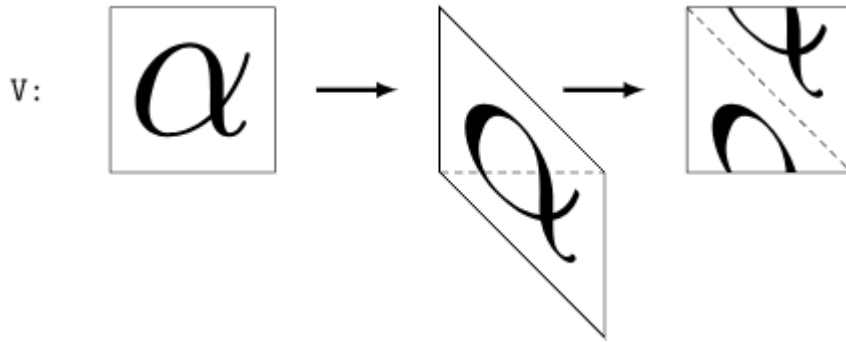
Pareizā atbilde ir:



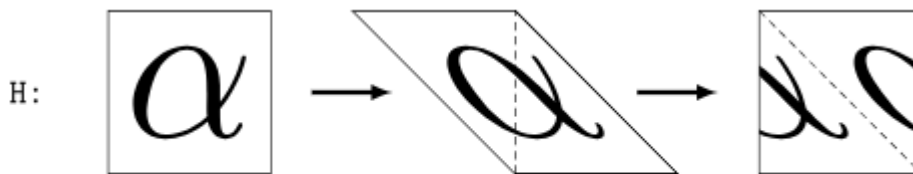
Attēlam ir augsta izšķirtspēja, tāpēc atsevišķu pikseļu kustība var būt diezgan labi pielīdzināta attēlā attēloto figūru ģeometriskajām transformācijām.

Piemērā 3×3 redzam, ka pirmā kolonna (1, 4, 7) kļūst par diagonāli no augšējā kreisā līdz apakšējā labajam stūrim, savukārt diagonāle no augšējā labā līdz apakšējā kreisajam stūrim (3, 5, 7) kļūst par pēdējo kolonnu. Tas liecina, ka notiek nobīde, ko apstiprina arī pirmais attēls Mona Lizas piemērā. Mēs sīkāk analizēsim divu darbību ģeometriskos efektus, sākot ar (V), jo tas tika pielietots pirmais.

(V) efekts var tikt sadalīts divos posmos. Vispirms taisnstūri sagriež vertikāli uz labo apakšējo pusi, lai izveidotu paralelogramu ar izkropļotu attēlu. Tad apakšā esošais trīsstūris tiek vertikāli pārvietots, lai izveidotu jaunu taisnstūrveida attēlu sākotnējā izmērā:



Līdzīgi (H) efekts ir horizontāla nobīde, kam seko daļēja horizontāla pārvietošana :



Piemērojot šos soļus dotajam attēlam, iegūstam šādu secību:



Šī ir informātika

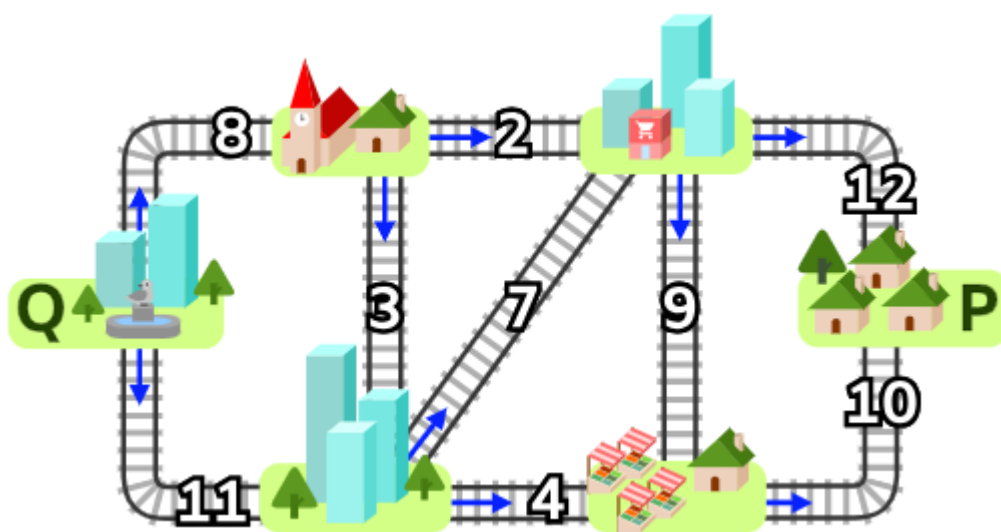
Pārsūtot informāciju, var būt svarīgi šifrēt datus, lai kāds, kas pārtver jūsu informāciju, nevarētu viegli uzzināt, kādi bija sākotnējie dati. Šajā uzdevumā jums ir dots šifrēšanas algoritms, kas izskatās pavisam vienkāršs, ir viegli apgriežams, tomēr joprojām ģenerē jaunu attēlu, no kura uzreiz nav skaidrs, kāds bija oriģināls. Patiesībā, ja jūs nezināt šifrēšanas mehānismu, varētu būt daudz sākotnējo attēlu. Taču ņemiet vērā, ka nav slepenas atslēgas, tāpēc faktiski viss, kas jums jāzina, ir šifrēšanas algoritms, un jūs varētu atšifrēt jebkuru šādā veidā šifrētu attēlu.

Dzelzceļa tīkls

Pakistāna

Bebravijas zemē blakus esošās apdzīvotās vietas savieno dzelzceļa sliežu tīkls.

Katram sliežu cejam ir noteikts ierobežots vilcienu skaits, kas pa to var braukt katru dienu, kā norādīts zemāk redzamajā diagrammā. Dažādi vilcieni var izmantot vienu sliežu ceju.



Kandrija(Q) piedāvāja nosūtīt materiālus uz Pixeltonu. Šajā virzienā braucot vilcieniem vienmēr jāseko bultiņām.

Jautājums

Kāds ir maksimālais vilcienu skaits, kas katru dienu var izbraukt no Qadrijas un ierasties Pixeltonā?

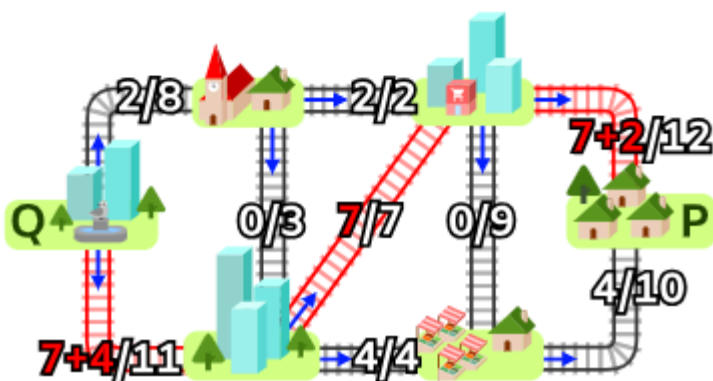
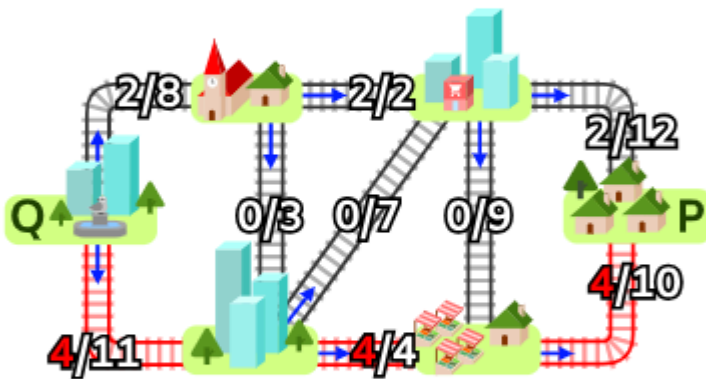
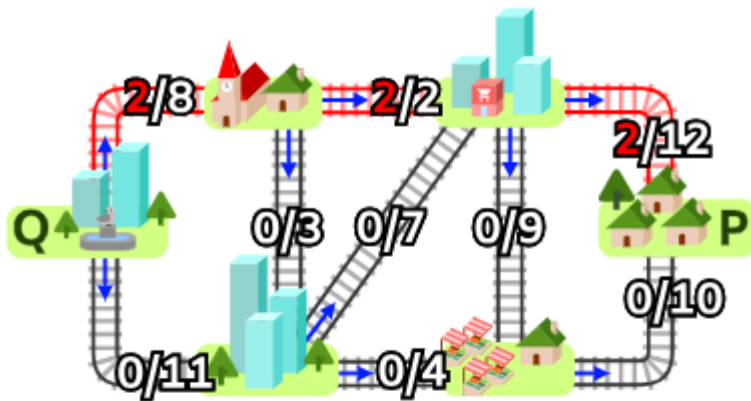
- a) 13
- b) 15
- c) 19
- d) 22

Skaidrojums

Pareizā atbilde ir a) 13. Pirmā pazīme ir tāda, ka Kandarija (Q) nevar nosūtīt uz Piksletonu (P) vairāk par $8+11=19$ vilcieniem, kas ir vilcienu skaits, kuri var izbraukt no Q. Tāpēc atbilde

c) 22 ir nepareiza.

Viens no veidiem, kā atrisināt šo uzdevumu, ir atrast īsāko iespējamo ceļu no Q uz P (gadījumā, ja garums ir 3) un noteikt lielāko vilcienu skaitu, ko tas var uzņemt. Pēc tam atkārtotam šo darbību pārējā tīklā, nosakot atlikušo vilcienu skaitu, kas pieejams katrā maršrutā pa šo sliežu ceļu. Turpmāk dotajās diagrammās parādīti trīs šādi ceļi, kuru kopējā ietilpība ir $2+4+7=13$ vilcienu. Ir arī citi veidi, kā no Q uz P nogādāt 13 vilcienu, tostarp daži, kuros izmanto garākus ceļus.



Lai pārbaudītu, ka 13 tiešām ir lielākais skaits, var pārbaudīt, ka 3 sliedes, kas atrodas pa vidu, var uzņemt tikai 2, 7 un 4 vilcienu. Tā kā visi ceļi iet caur šīm slīdēm, mēs nevaram atrast risinājumu, kas ir lielāks par 13.

Ši ir informātika

Šajā uzdevumā risināmā problēma tiek dēvēta par maksimālās plūsmas problēmu (*maximum flow problem*). Šāda veida uzdevumi ir kā puzzles spēle, kurā ūdens plūst caur caurulēm: jums ir cauruļu tīkls ar dažādu ietilpību, un jūs vēlaties noskaidrot, cik daudz ūdens var plūst no sākumpunkta līdz beigām.

Maksimālās plūsmas problēmu var atrisināt, izmantojot dažādus algoritmus: Forda-Fulkersona algoritmu, Edmonda-Karpa algoritmu, Dinika algoritmu un daudzus citus. Šajā uzdevumā tie ceji tīkla vidū, par kuriem ir obligāti jāizbrauc, veido tā saukto "minimālo griezumu"

Katram no iepriekš minētajiem algoritmiem ir savas priekšrocības un trūkumi attiecībā uz izpildes laika efektivitāti, īstenošanas vienkāršību un piemērojamību dažādiem tīklu tipiem. Algoritma izvēle ir atkarīga no tādiem faktoriem kā tīkla lielums, jaudu raksturs un vēlamais optimizācijas līmenis.

Maksimālās plūsmas algoritmus izmanto satiksmes optimizācijā, sadales sistēmās (piemēram, ūdens un elektrības), resursu sadalē, datortīklos un citur.

Palago

Ungārija

Palago ir divu spēlētāju spēle, kas tiek spēlēta ar vienādām sešstūrīgām figūrām. Katra figūra ir iekrāsota zilā vai baltā krāsā, kā parādīts attēlā.

Viens spēlētājs ir balts, otrs ir zils. Spēles mērķis ir izveidot noslēgtu (pabeigtu) figūru savā krāsā.



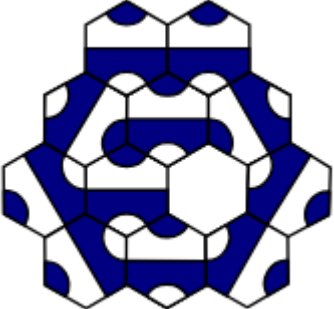
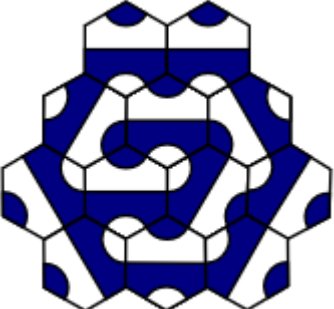
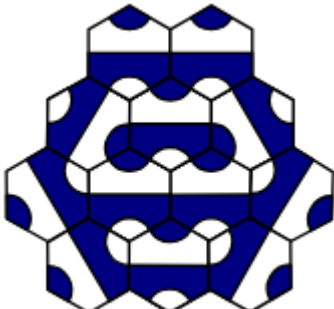
Pirmais spēlētājs novieto divas figūras blakus viena otrai, kuru malu krāsām jir jābūt vienādām. Pēc tam spēlētāji pārmaiņus novieto divas figūras katrā gājienā, ievērojot šādus noteikumus:

1. Vismaz vienai no divām figūrām ir jāatrodas blakus jau izvietotai figūrai.
2. Abām novietotajām detaļām jāatrodas blakus viena otrai, t.i., tām jāsasakaras ar kādu no malām.
3. Visām figūru malām, kas saskaras ir jābūt vienādās krāsās.

Spēlētājs uzvar, ja viņš var izveidot noslēgtu figūru sevis izvēlētajā krāsā ar vismaz vienu taisnu malu. Tomēr spēlētājs zaudē, ja viņš tajā pašā gājienā izveido noslēgtu figūru abās krāsās. Ja ir nepieciešama tikai viena detaļa uzvarai, otrā detaļa nav jānovieto.

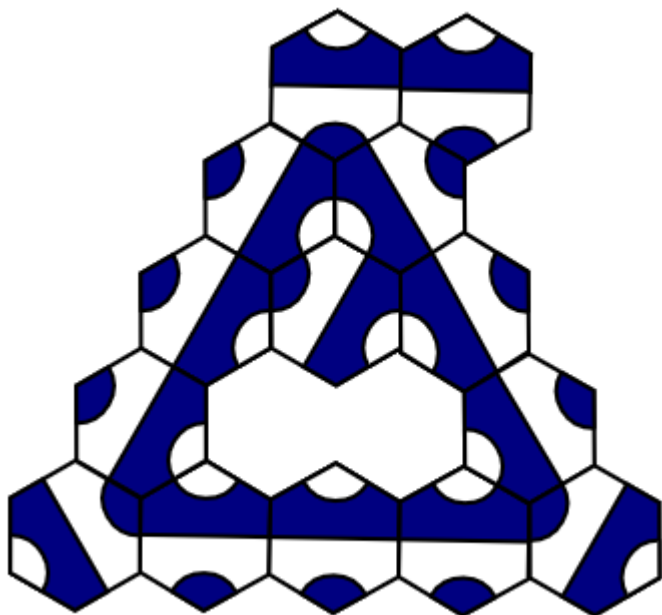
Situācijā pa labi, zilajam ir izveidojusies noslēgta forma, bet tā neietver taisnu līniju. Tāpēc zilais nav uzvarējis. (Pagaidām). Baltajam nav noslēgta forma, tāpēc, baltais nav uzvarējis.



| Pašreizējā spēles situācija | Pēc nākamās figūras uzlikšanas | |
|---|--|---|
|  |  | <p>...Baltais uzvar, jo spēlētājs ir izveidojis noslēgtu baltu figūru ar taisnu līniju.</p> |
| |  | <p>... Baltais zaudē, jo spēlētājs ir izveidojis ne tikai baltu noslēgtu figūru, bet arī zilu noslēgtu figūru ar taisnu līniju.</p> |

Jautājums

Kurš uzvarēs šajā spēles situācijā, ja abi spēlētāji spēlē optimāli, lai uzvarētu?



Zilais uzvar

Baltais uzvar

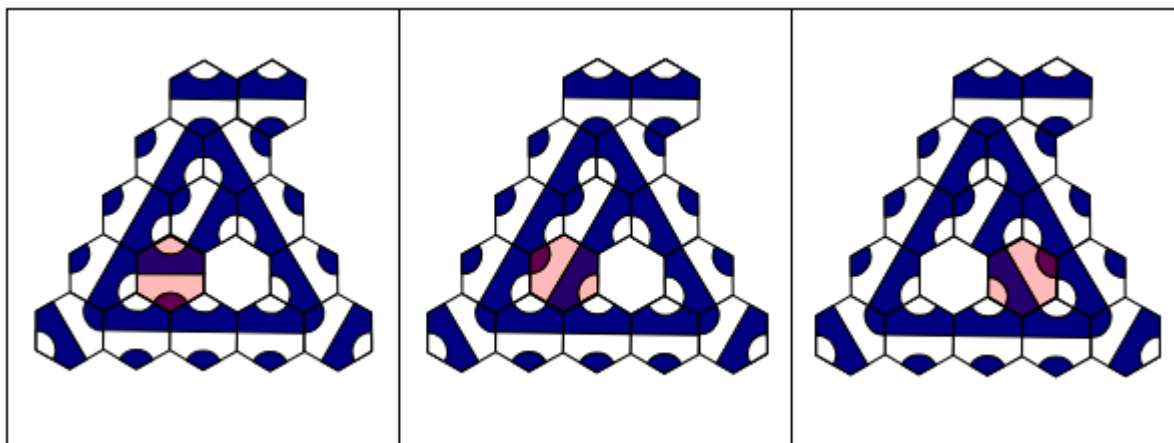
To nevar noteikt

Tas ir atkarīgs no tā, kuram ir nākamais gājiens

Skaidrojums

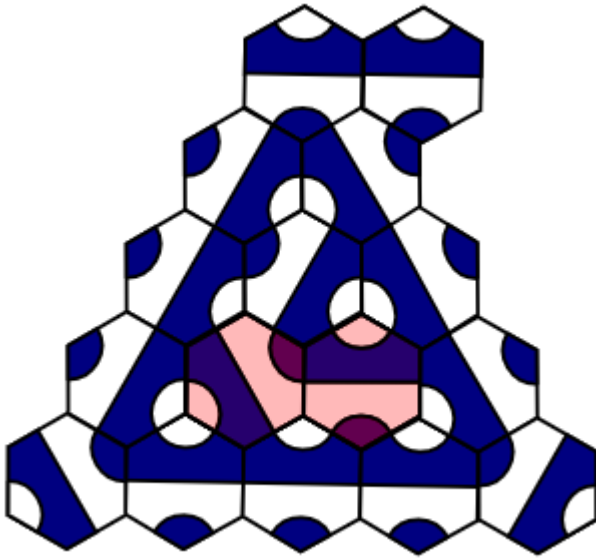
Pareizā atbilde: Baltais uzvar.

Baltais var novietot savu nākamo figūru tā, lai viņš ir pabeidzis baltu noslēgtu figūru ar vienu detaļu, neizveidojot zilu figūru. Viņš to var izdarīt trīs dažādos veidos:



Saskaņā ar noteikumiem, spēlētājam nav jānovieto arī otra detaļa.

Kad Zilais novieto savu nākamo figūru, viņš var aizpildīt caurumu un noslēgt zīmējumu ar abām detaļām, bet tad viņš arī noslēgtu figūru Baltajam, tādējādi zaudējot spēli.



Ja Zilais neizveido figūru, spēle turpinās, un Baltais var to izdarīt nākamajā gājienā. Zilajam nav iespējas izveidot figūru ap jau izspēlētajām figūrām, tāpēc Baltais var uzvarēt vēlākais nākamajā gājienā.

Tāpēc atbilde "baltais uzvar" ir pareiza.

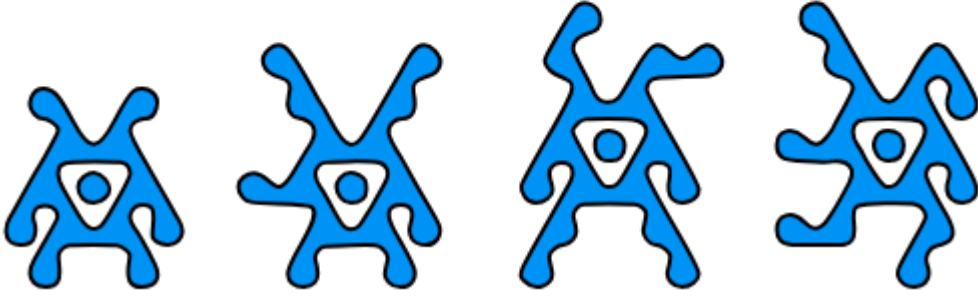
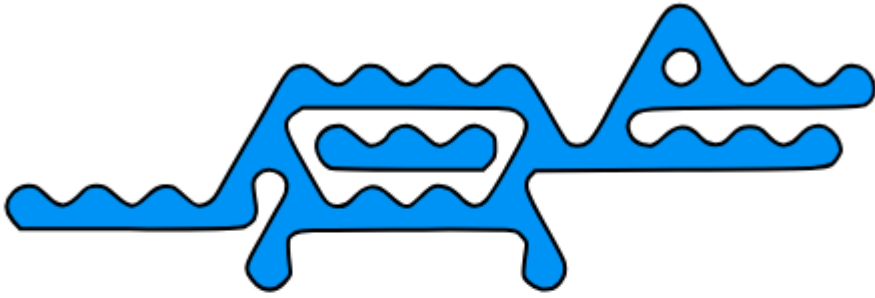
Ši ir informātika

Stratēģiskā vairāku spēlētāju spēlē, piemēram, Palago, ir nepieciešama stratēģija, lai uzvarētu vai izvairītos no zaudējuma nākamajos gājienos. Ir svarīgi paredzēt nākamās iespējamās gājienus un to rezultātus, lai pieņemtu lēmumu, ko darīt tālāk.

Šādas spēles var sistemātiski izpētīt, izmantojot "spēles koku". Spēles koka sakne ir sākuma pozīcija, un visi iespējamie gājieni no katras pozīcijas ir attēloti zaru veidā. Tādējādi tiek izveidots koks, kas attēlo visas iespējamās spēles situācijas šajā spēlē. Izveidojot spēles koku, jūs varat atrast ceļu, pa kuru varat "iet", lai uzvarētu, pirms pretinieks to izdarīs pirmais.

Dažām spēlēm (piemēram, šaham, Go un Palago) ir milzīgs vai pat bezgalīgi liels spēles koks, tāpēc ir jāstrādā ar daļēju spēles koku un jāanalizē tikai daļa no visa koka, piemēram, tikai nākamie pieci gājieni. Alternatīvi, jūs varat novērtēt spēles situāciju no malas un izpētīt tikai tās pozīcijas, kas sola panākumus.

Palago ir radošs mākslas galvlauzis, kuru izstrādājis Kamerons Brauns. Spēlē Palago nav jāspēlē tikai, lai uzvarētu. Daži spēlētāji mēģina veidot "palagoniešus" (Palagonijas iedzīvotājus) vai dzīvniekus no akmeņiem.



Turpini mainīt

Beļģija

Ar mašīnu var pārvietot dažāda augstuma blokus. Tā izmanto marķieri, kas parasti ir novietots starp diviem blokiem. No sākuma bloki un marķieris ir starta pozīcijā, piemēram, šādā:



Mašīna atkārtoti vienu no diviem soļiem:

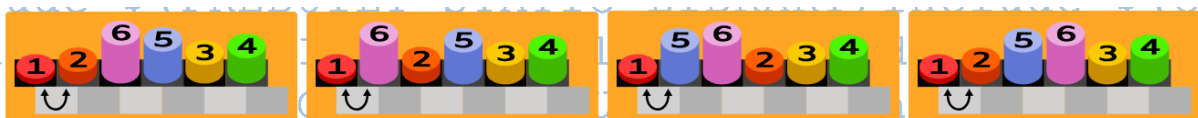
| | |
|---|--|
| Ja bloks no marķiera pa kreisi ir mazāks par bloku, kas atrodas pa labi, marķieris to pārvieto pa labi. | |
| Ja bloks pa kreisi ir lielāks par bloku pa labi, bloki tiek samainīti. Papildus, marķieris pārvietojas pa kreisi, bet tikai tad, ja turpina atrasties pa vidu diviem blokiem. | |

Mašīna apstājas, kad marķieris ir bloku labajā malā un tālāk tam nav kur iet.

Atkarībā no starta pozīcijas, mašīna var iet dažādu soļu skaitu

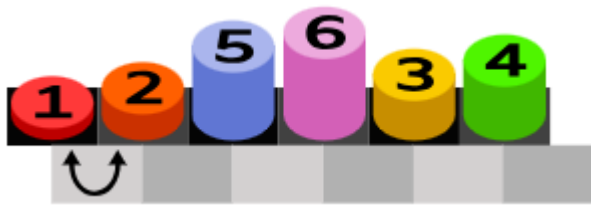
Jautājums

Kurā no šīm starta pozīcijām, mašīnai ir jāveic vismazākais soļu skaits?



Skaidrojums

Pareizā atbilde ir šī:



Lai noteiktu pareizo atbildi, ir iespējams simulēt mašīnas darbību, skaitot tās soļus, līdz tā apstājas.

Ir vieglāk salīdzināt šīs 4 starta pozīcijas, sāukumā neskaitot visus iespējamās versijas.

Lai atvieglotu risināšanu, ir nepieciešams reprezentēt klucīšus kā ciparus (pēc klucīšu augstuma) un novietot marķieri starp diviem klucīšiem. Šajā zīmējumā, sākuma pozīcija izskatās šādi:

1 ▲ 6 4 2 5 3

Mēs reprezentēsim katru albildi kā burtu:

| | | | |
|---|---------------|---|---------------|
| A | 1 ▲ 2 6 5 3 4 | C | 1 ▲ 5 6 2 3 4 |
| B | 1 ▲ 6 2 5 3 4 | D | 1 ▲ 2 5 6 3 4 |

Ja mašīna sāk no atbildes B, atbilde A ir sasniedzama 2 soļos.

1 ▲ 6 2 5 3 4 → 1 6 ▲ 2 5 3 4 → 1 ▲ 2 6 5 3 4

Priekš atbildes B, mašīnai ir nepieciešami par 2 soļiem vairāk nekā atbildei A. Tāpēc atbilde B nevar būt pareiza

Ja mašīna sāk kā norādīts atbildē A, viņa veic šos 3 soļus:

1 ▲ 2 6 5 3 4 → 1 2 ▲ 6 5 3 4 → 1 2 6 ▲ 5 3 4 → 1 2 ▲ 5 6 3 4

No atbildes A, atbilde D ir sasniedzama 1 solī:

1 2 5 6 3 4 → 1 2 5 6 3 4

Priekš atbildes A, mašīnai vajadzīgi par 2 soļiem vairāk nekā atbildei D. Tāpēc atbilde A nevar būt pareiza.

Ja mašīna sāk kā ir norādīts atbildē C, versija D no tās ir sasniedzama 4 soļos:

1 5 6 2 3 4 → 1 5 6 2 3 4 → 1 5 6 2 3 4 → 1 5 2 6 3 4 → 1 2 5 6 3 4

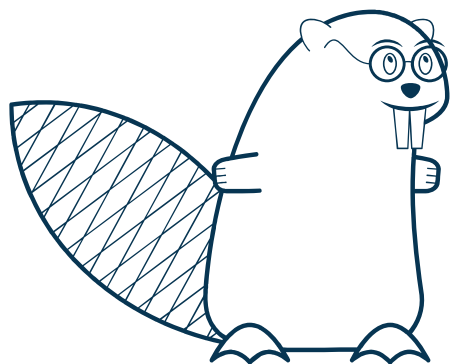
Tā kā mašīnai ir vajadzīgi par 4 soļiem vairāk priekš versijas C nekā priekš versijas D.

Mēs varam secināt, ka vismazāk soļi vajadzīgi versijai D.

Šī ir informātika

Datu kārtošana gan pēc lieluma, cenas un alfabētiski ir standarta process datorzinātnēs. Tāpēc datu zinātnieka darbs ir atrast visefektīvāko datu kārtošanas metodi.

Šī izmantotā datu kārtošanas metode ir "Rūķu metode", tomēr tā nav pārāk efektīva. Datu zinātnieki šo datu kārtošanas veidu vērtētu kā vidēju, ņemot vērā tā kārtošanas laiku un sliktāku nekā tādas efektīvas datu kārtošanas metodes kā "ātrā kārtošana" (quicksort). Tomēr šī metode ir vieglāk uztverama un saprotama.



copyright Bebras