



Bebr[a]s'20

2020. gada
1. kārtas uzdevumi

9.-10. klasei



SATURS

<u>Akmeni</u>	2
<u>Klucīšu tornis</u>	4
<u>Lēkšana un bultas</u>	7
<u>Lēkājošais ķengurs</u>	9
<u>Teātra izrāde</u>	11
<u>Skaitļu veidošana</u>	13
<u>Puku kaste (interaktīvs)</u>	15
<u>Koku sudoku (interaktīvs)</u>	18
<u>Maģiskā dzērienu pagatavošanas mašīna 2</u>	21
<u>Bebru tikšanās</u>	24
<u>Bebru torņu pulkstenis (interaktīvs)</u>	28
<u>Bebri pret ķenguriem</u>	31
<u>Zinājumu izplatīšana (interaktīvs)</u>	34
<u>Robotu labirints (interaktīvs)</u>	36
<u>Epidēmiskā krīze</u>	39

Akmeņi

Ungārija

Uzdevuma apraksts

Ansītis un Grietiņa spēlē spēli.

Laukumā ir 3 melni un 7 balti akmeņi. Katrā gājienā spēlētājs var noņemt 1 vai 2 melnus akmeņus vai 1, 2 vai 3 baltus akmeņus. Spēli uzvar tas, kurš noņem pēdējo akmeni no jebkuras krāsas.



Jautājums

Spēli sāk Grietiņa. Cik akmeņi viņai jānoņem pirmajā gājienā, lai viņa uzvarētu?

- A) 1 balts akmens
- B) 2 melni akmeņi
- c) 3 balti akmeņi
- D) Tam nav nozīmes

Pareizā atbilde: C) 3 balti akmeņi

Atbildes izskaidrojums

Par trim melniem akmeņiem: Ja Grietiņa noņems vienu vai divus melnus akmeņus, tad Ansītis noņems atlikušos akmeņus. Tātad melno akmeņu ņemšana nav uzvaroša stratēģija.

Par baltajiem akmeņiem: Jebkurā četrus akmeņu grupā otrais spēlētājs var noņemt atlikušos akmeņus neatkarīgi no tā, cik akmeņus būs noņēmis pirmais

spēlētājs. Ja pirmais noņems vienu, tad otrais - trīs, ja pirmais divus, tad otrais arī divus, ja pirmais trīs, tad otrais - vienu.

Tas nozīmē, ka uzvarošā stratēģija ir atstāt pretiniekam četrus baltus akmeņus. Tātad Grietiņai pēc sava gājiena jāatstāj trīs melni un četri balti akmeņi. **Tātad Grietiņai pirmajā gājienā jānoņem trīs balti akmeņi.**

Izdarot jebkuru citu gājienu, Ansītim ir iespēja "nozagt" uzvarošo stratēģiju un, tātad, citi gājieni nav uzvaroši.

Tā ir informātika!

Spēles ir svarīga datorzinātņu tēma. Spēles var izmantot reālu procesu modelēšanai un raksturīga algoritmiskā problēma ir atrast uzvarošu stratēģiju vienam no spēlētājiem.

Šajā uzdevumā risinājuma meklēšanu var sākt no pozīcijām (akmeņu komplektiem), kur Grietiņai var uzvarēt ar pēdējo gājienu, un atzīmēt šīs pozīcijas kā Grietiņai uzvarošas un, attiecīgi, Ansītim zaudējošas. No tā var atrast likumsakarības, kas Grietiņai jāuztur, lai vienmēr pēc sava gājiena nonāktu uzvarošā pozīcijā. To sauc par invariantu, kam jā saglabājas pēc katra izdarītā gājiena un kuru Ansītis ar saviem gājieniem izjaukt nevar. Ja Grietiņa var saglabāt šo invariantu, tad viņai ir uzvarošā stratēģija.

Šāda veida analīzi var attiecināt arī uz tādām sarežģītām spēlēm kā šahs un Go, bet šajās spēlēs iespējamo pozīciju skaits kļūst pārāk liels, un ir jāizmanto citas pieejas, piemēram, mašīnmācīšanās, lai "iemācītos" labu stratēģiju.

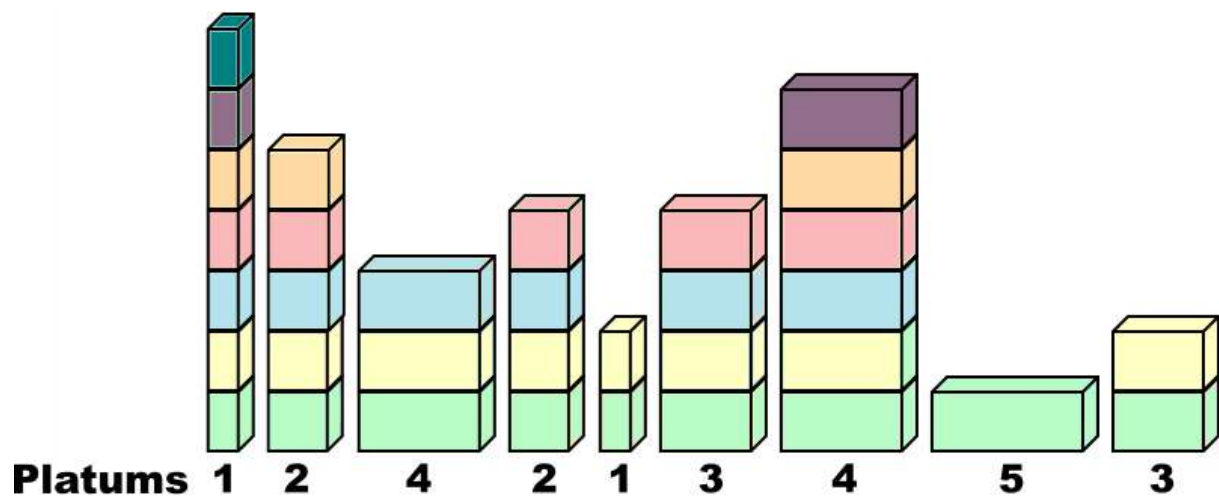
Nim bija viena no pirmajām spēlēm, kas tika realizēta elektroniskajā ierīcē. Šobrīd ar kombinatoriskās spēļu teorijas palīdzību Nim ir atrisināta jebkuram skaitam kaudzīšu un objektu skaitam tajās. Tāpēc ir relatīvi viegli šo spēli ieprogrammēt un noteikt, kurš spēlētājs uzvarēs un kādi uzvarošie gājieni ir šim spēlētājam iespējami.

Klucīšu tornis

Portugāle

Uzdevuma apraksts

Mazais bebrs Sems spēlējas ar rotaļu klucīšiem. Visiem blokiem ir vienāds augstums, bet var būt atšķirīgs platums, kas izteikts platuma vienībās (naturāls skaitlis). Sems ir uzbūvējis deviņus skaistus torņus, katru torni būvējot no vienāda platuma klucīšiem.



Ir divi veidi, kā iespējams izmainīt torņa augstumu - vai nu torņa augšā uzlikt klucīšus klāt, vai arī noņemt nost. Jebkurā gadījumā uzskatīsim, ka enerģija, kas jāpatērē lai izmainītu torņa augstumu, ir proporcionāla torņa platumam un nomainīto (pievienoto vai noņemto) klucīšu skaitam. Piemēram, noņemot divus klucīšus tornim, kura platums ir viena vienība, būs nepieciešamas $2 \cdot 1 = 2$ enerģijas vienības. Savukārt, pievienojot četrus klucīšus tornim, kura platums ir trīs vienības, būs nepieciešamas $4 \cdot 3 = 12$ enerģijas vienības.

Sems vēlas panākt, ka visi torņi ir vienā augstumā, tam patērējot iespējami mazāku enerģijas daudzumu.

Jautājums

Kāds ir mazākais enerģijas daudzums ar kādu Sems visus torņus var pārveidot par vienāda augstuma torņiem?

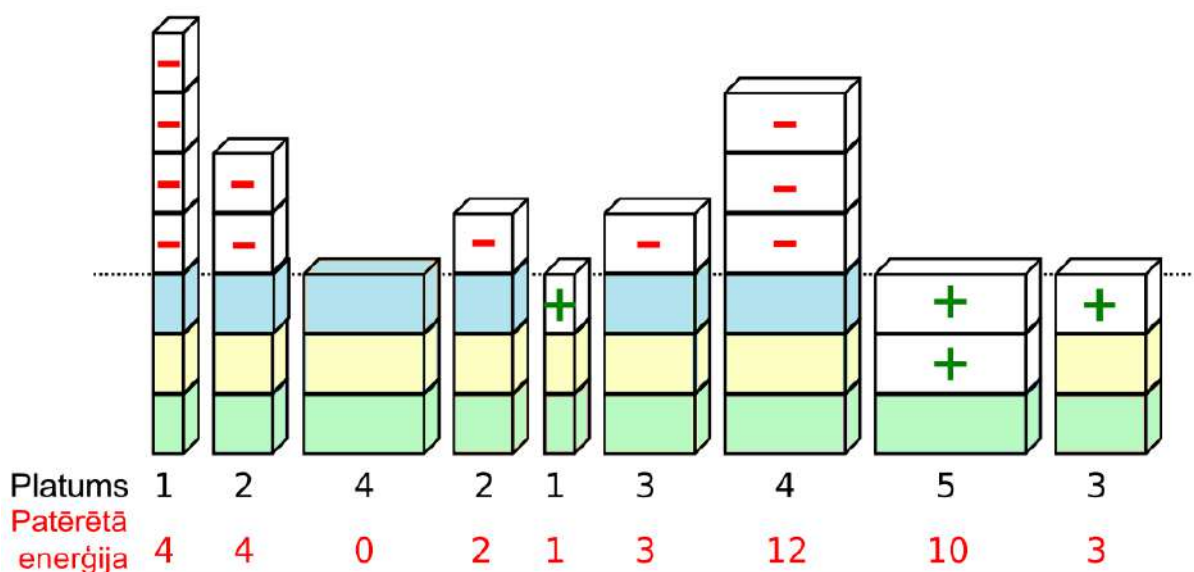
Atbildes forma

Naturāls skaitlis, kas norāda nepieciešamo enerģijas vienību skaitu.

Pareizā atbilde: 39

Atbildes izskaidrojums

Ja pievienojamie klucīši ir attēloti kā , bet noņemamie - kā , tad zīmējumā parādīts pareizais atrisinājums, visu torņu augstumu novienādojot uz augstumu 3:



Visu torņu vidējais augstums ir 4 (1 2 2 3 4 4 5 6 7) un var iedomāties, ka šim arī būtu jābūt beigu torņu augstumam, bet tā nav. Tabulā parādīts, ka visiem pārējiem torņu augstumiem ir nepieciešams lielāks enerģijas daudzums:

	Originālais augstums	7	5	3	4	2	4	6	1	2	Kopējais enerģijas patēriņš
	Platums	1	2	4	2	1	3	4	5	3	
Enerģijas patēriņš	Augstums 1	6	8	8	6	1	9	20	0	3	61
	Augstums 2	5	6	4	4	0	6	16	5	0	46
	Augstums 3	4	4	0	2	1	3	12	10	3	39
	Augstums 4	3	2	4	0	2	0	8	15	6	40
	Augstums 5	2	0	8	2	3	3	4	20	9	51
	Augstums 6	1	2	12	4	4	6	0	25	12	66
	Augstums 7	0	4	16	6	5	9	4	30	15	89

Tas tāpēc, ka torņu platums nav vienāds. Tomēr funkcijai, kas nosaka nepieciešamo enerģijas daudzumu, ir tikai viens minimums. Šī minimuma atrašanai var lietot gudrāku stratēģiju, uzreiz atmetot variantus ar tikai nojaukšanu (gala augstums vienāds ar 1) vai tikai piebūvēšanu klāt (gala augstums 7). Tikai mazliet grūtāk ir pārlicināties, ka arī augstumi 2 un 6 nebūs optimāli. Atliek pārbaudīt gala augstumus 3, 4 un 5 un izvēlēties to, kur enerģijas daudzums ir mazākais.

Tā ir informātika!

Katru klucīšu torni datorprogrammā var modelēt ar datu struktūru, ko sauc par "steku" ar divām iespējamām darbībām: "pievieno" (jauna bloka pievienošana kaudzes augšpusē) un "noņem" (augšējā bloka noņemšana). Šajā uzdevumā jāņem vērā katra jau uzceltā torņa pašreizējais augstums un platums. Stekus izmanto daudzos algoritmos, kas ir krietni sarežģītāki nekā šajā uzdevumā. Piemēram, tos var izmantot, lai aprēķinātu aritmētisku izteiksmju vērtības.

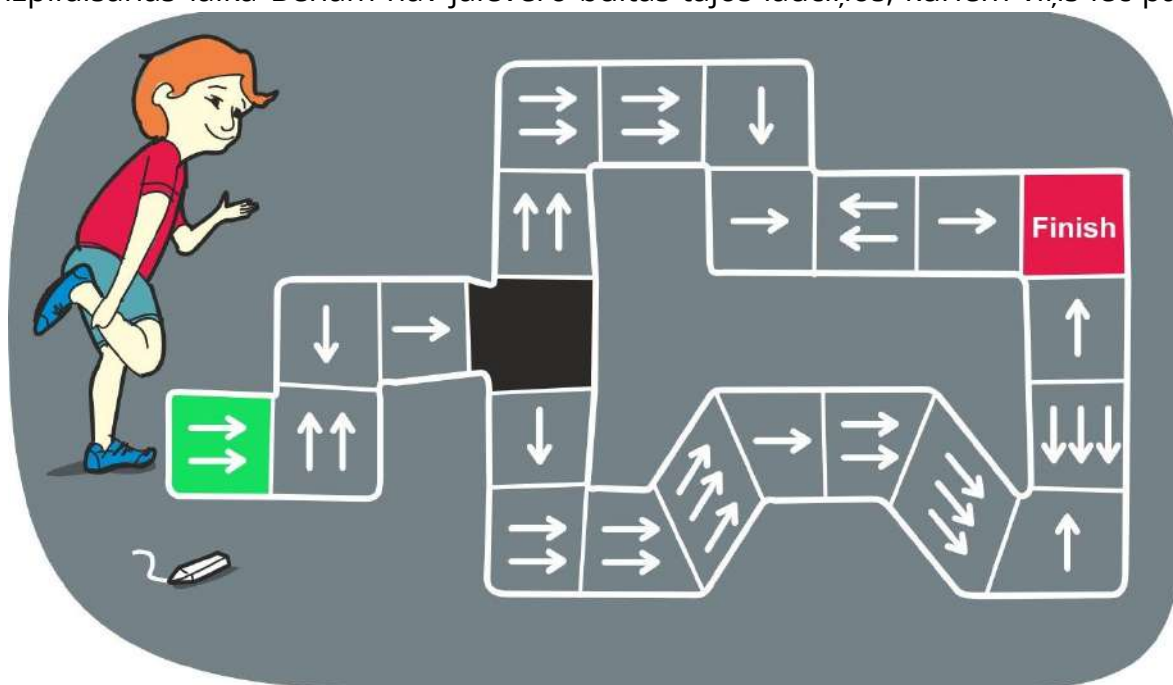
Datorzinātnēs ļoti bieži tiek meklēta labākā iespēja starp visām iespējamajām (piemēram, kādas ir vismazākās izmaksas). To sauc par optimizācijas uzdevumu. Šajā gadījumā izmaksām ir unimodāls sadalījums, tāpēc var izmantot ternāro meklēšanu, kas nosaka, ka minimums vai maksimums nevar būt ne domēna pirmajā, ne pēdējā trešdaļā, un pēc tam meklēšana jāatkārto atlikušajās divās trešdaļās. Ternārā meklēšana ir "skaldi un valdi" algoritma piemērs.

Lēkšana un bultas

Lietuva

Uzdevuma apraksts

Bens ir uzzīmējis klasīšu lēkāšanas spēlei līdzīgas spēles laukumu. Spēles sākumā Bens atrodas pirmajā (zaļajā) laukuma lauciņā un atkārtoti izdara gājienus, kurus apraksta tajā lauciņā, kurā Bens tobrīd atrodas, iezīmētās bultas. Bultu skaits nosaka, cik lauciņus uz priekšu Benam jāpalec, bet bultu virziens norāda, kurā virzienā atrodas nākamais lauciņš, uz kuru jālec. Gājiena izpildīšanas laikā Benam nav jāievēro bultas tajos lauciņos, kuriem viņš lec pāri.



Jautājums

Kuram no dotajiem zīmējumiem jābūt melnā laukuma vietā, lai Bens varētu sasniegt sarkano beigu laucinu?

A	B	C	D
			

Pareizā atbilde: C

Visos citos gadījumos uzzīmētās bultiņas noved Benu mūžīgajā ciklā.

Tā ir informātika!

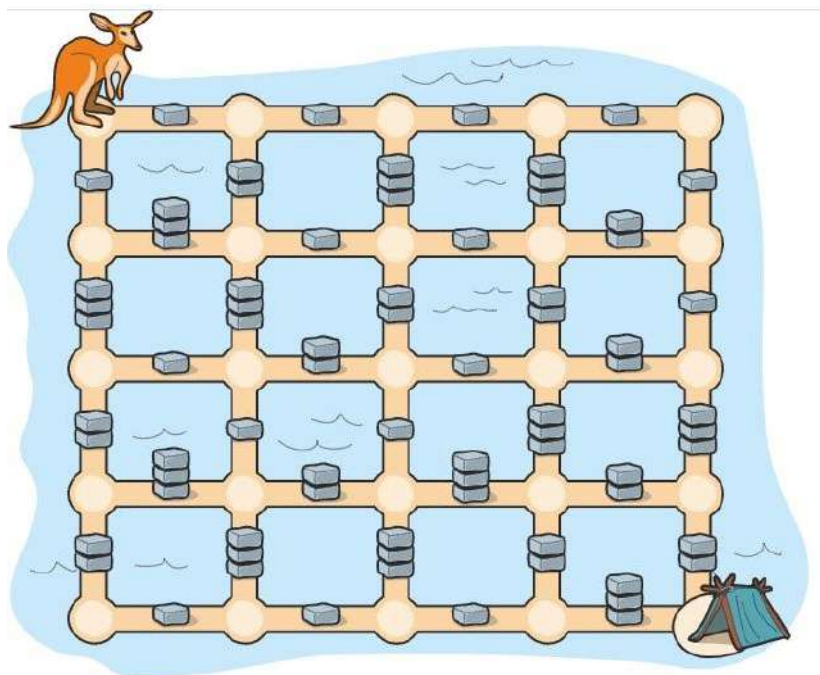
Algoritms ir instrukciju vai noteikumu kopums, lai atrisinātu noteiktu uzdevumu. Instrukciju pierakstam var izmantot dažnedažādus izteiksmes līdzekļus un apzīmējumus. To starpā ir dabiskās valodas, pseidokodi, blokshēmas, programmēšanas valodas utt. Tā kā algoritms ir precīzs skaidri definētu darbību apraksts, algoritma darbībai vienmēr ir svarīga arī darbību un aprēķinu secība. Algoritmos pamatā tiek izmantoti trīs darbību izpildes veidi: secīga izpilde, izvēle (nosacījuma operācija) un atkārtošana (cikls). Šajā Bebru uzdevumā vissvarīgākā ir atkārtošana. Bens visu laiku atkārtoti vienu un to pašu pamatinstrukciju - līdz viņš sasniedz finišu. Tomēr atbilde A, B un D gadījumā izrādās, ka Bens iekļūst bezgalīgā ciklā - nebeidzamu instrukciju virknē, kurai nav beigu vai izejas nosacījuma. Lai gan programmētājam dažkārt ir izdevīgi izmantot bezgalīgā cikla konstrukciju (piemēram, nav zināms, cik reizes cikla ķermeni nāksies izpildīt, bet izešanu no cikla nosaka kāda nosacījuma izpilde), jaunie programmētāji mēdz kļūdīties un netīšām izveidot bezgalīgu ciklu.

Lēkājošais ķengurs

Lietuva

Uzdevuma apraksts

Ķengurs lēkā mājup pa zīmējumā attēlotajiem horizontālajiem un vertikālajiem ceļu posmiem kuros katrā atrodas viens vai vairāki ķieģeļi, kas sakrauti tornīšos. Katru ceļa posmu ķengurs veic ar vienu lēcieni, bet var pārlēkt ķieģeļu tornītim tikai tad, ja tajā ir ne vairāk kā divi ķieģeļi. Ķengurs vēlas nokļūt mājās pēc iespējas agrāk - veicot pēc iespējas mazāk lēcienus.



Jautājums

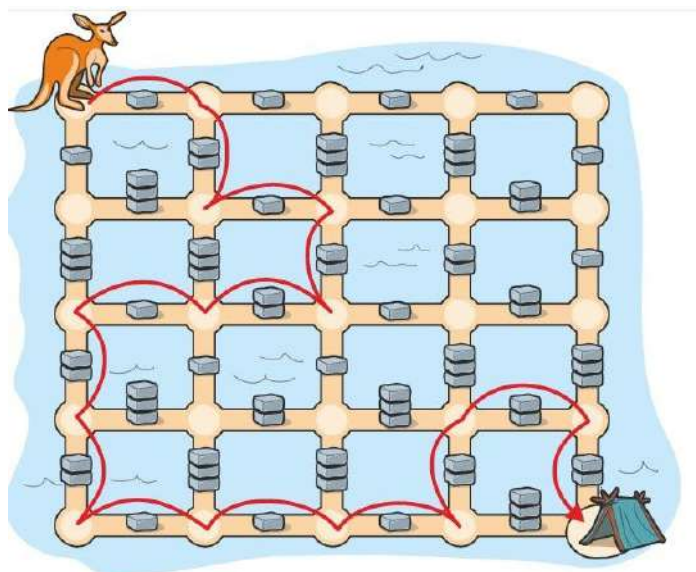
Ar kādu mazāko lēcienu skaitu ķengurs var nokļūt mājās?

Atbildes forma

Patvaļīgs naturāls skaitlis.

Pareizā atbilde: 14 lēcieni.

Atrisinājums parādīts zemāk dotajā zīmējumā.



Tā ir informātika!

Lai atrastu risinājumu, var rīkoties šādi: sākt meklēšanu, iet soli pa solim un, ja nonākam strupceļā, kur visiem iespējamajiem ceļiem ir trīs akmeņi, doties atpakaļ (iespējams, vairākus soļus) un kā nākamo izmēģināt citu iespējamu soli vai soļu virkni.

Šī pieeja informātikā ir zināma kā atgriezmeklēšana, un to izmanto daudzos algoritmos. Piemēram, lai atrisinātu loģiskas mīklas, sudoku vai kombinatoriskās optimizācijas problēmas.

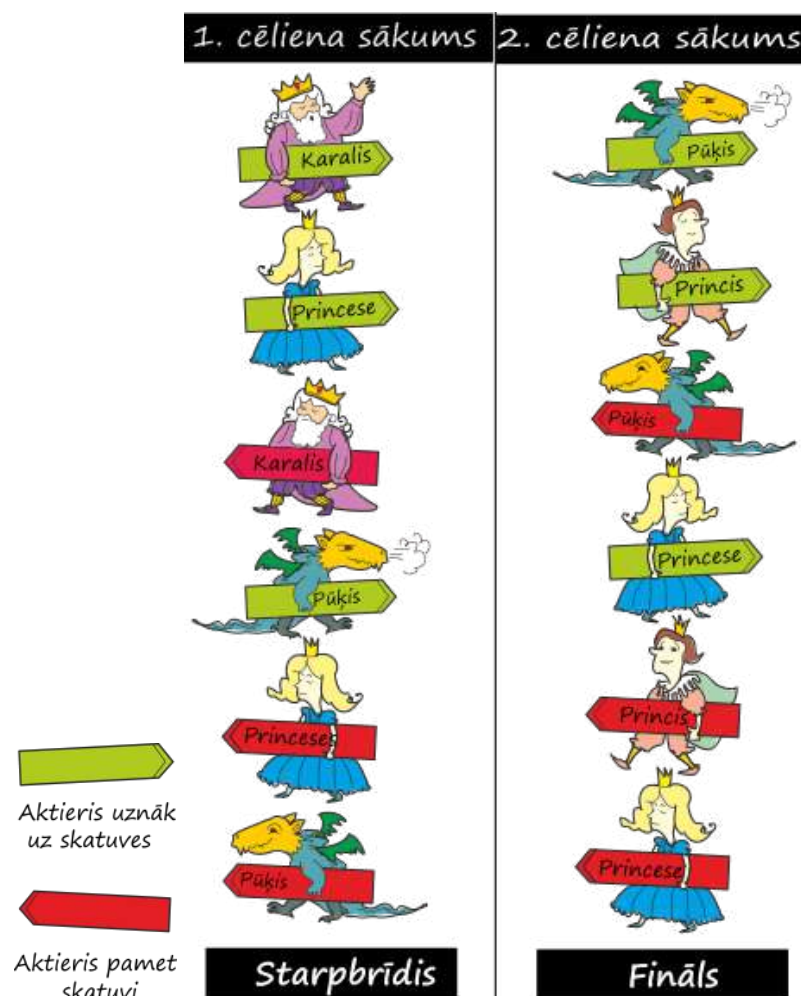
Šis uzdevums parāda, ka dažreiz efektīvāk ir sākt risinājumu meklēt pretējā virzienā - no beigām, šeit sākot no ķenguru mājas. Šajā uzdevumā, šādi risinot, mums ir nepieciešams atgriezties mazāku skaitu reižu, un risinājumu ir vieglāk atrast. Bet bez uzdevuma iepriekšējas izpētes nav iespējams pateikt, vai labāk sākt risinājuma meklēšanu no sākuma vai no beigām

Teātra izrāde

Slovākija

Uzdevuma apraksts

Aktieri pasaku lugas uzvedumā uznāk uz skatuves un noiet no tās saskaņā ar zīmējumā attēloto secību virzienā no augšas uz leju. Izrādei ir divi cēlieni ar starpbrīdi starp tiem.



Jautājums

Kurš no dotajiem apgalvojumiem **nav patiess**?

Atbilžu varianti

- A) Princis un princese uz skatuves ir atradušies vienlaikus.
- B) Karalis un pūķis uz skatuves ir atradušies vienlaikus.

C) Princis uz skatuves uznāca tikai pēc starpbrīža (2. cēlienā).

D) Princis un pūķis uz skatuves ir atradušies vienlaikus

Pareizā atbilde: B) Karalis un pūķis uz skatuves ir atradušies vienlaikus.

Atbildes izskaidrojums

A) Apgalvojums "Princis un princese uz skatuves ir atradušies vienlaikus" ir patiess. Otrajā cēlienā princis skatuvi vēl nebija pametis, kad uz tās uznāca princese.

B) Apgalvojums "Karalis un pūķis uz skatuves ir atradušies vienlaikus" **nav patiess**, jo brīdī, kad 1. cēlienā uz skatuves paādījās pūķis, karalis skatuvi jau bija pametis, bet 2. cēlienā karalis uz skatuves neparādījās.

C) Apgalvojums "Princis uz skatuves uznāca tikai pēc starpbrīža (2. cēlienā)" ir patiess, jo 1. cēlienā princis uz skatuves neparādās, bet 2. cēlienā - parādās.

D) Apgalvojums "Princis un pūķis uz skatuves ir atradušies vienlaikus" ir patiess, jo viņi abi pēc kārtas uznāk uz skatuves 2. cēliena sākumā.

Tā ir informātika!

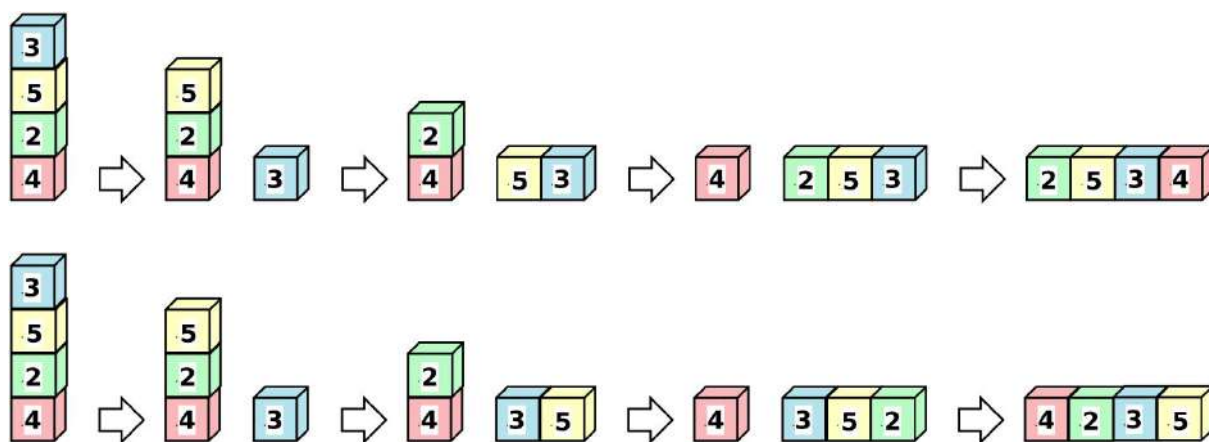
Informātikā bieži grafiski attēlo procesus (laika gaitā secīgus notikumus). Informācijas, datu vai zināšanu grafiskie attēlojumi ir paredzēti, lai ātri un skaidri attēlotu informāciju, kas citādi būtu grūti uztverama. Tie var uzlabot izpratni, izmantojot cilvēka redzes sistēmas spēju ieraudzīt modeļus un tendences. Vizuālo attēlojumu (vai diagrammu) izpratne un adekvātu secinājumu izdarīšana ir svarīga prasme, kuras attīstīšanā datorzinātne var būtiski palīdzēt.

Skaitļu veidošana

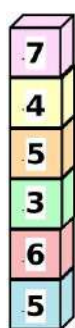
Portugāle

Uzdevuma apraksts

Bebres Olīvija rotaļājas ar klucīšiem. Uz katra klucīša ir uzrakstīts kāds cipars. Olīvijai patīk salikt klucīšus vienā augstā tornī un tad ņemt klucīšus no šī torņa pa vienam pēc kārtas un veidot naturāla skaitļa pierakstu. Katru noņemto klucīti Olīvija var likt iepriekš izveidotās klucīšu virknes labajā vai kreisajā galā. Zīmējumā parādīts piemērs, kā no viena un tā paša četru klucīšu torņa iespējams izveidot divu atšķirīgu skaitļu (2534 un 4235) pierakstu. Turklāt, šie nav vienīgie iespējamie varianti!



Tikko Olīvija ir uzbūvējusi zīmējumā redzamo sešu klucīšu torni.



Jautājums

Kāds ir mazākais naturālais skaitlis, kura pierakstu Olīvija var izveidot no šī torņa iepriekš aprakstītajā veidā?

Atbildes forma

Naturāls sešciparu skaitlis.

Pareizā atbilde: 347565.

Atbildes izskaidrojums

Skaitlim jāsākas ar mazāko tornī pieejamo ciparu, kas ir 3. Visi turpmākie klucīši būs jāliek klucīšu virknes labajā galā. Tātad skaitļa pieraksta struktūra būs $3^{***}65$. Zvaigznīšu vietā jābūt mazākā skaitļa pierakstam, ko iespējams izveidot no pirmajiem trim klucīšiem. Tam jāsākas ar 4 un tātad šis skaitlis ir 475. Iegūstam beigu rezultātu - 347565.

Tā ir informātika!

Labākā atrisinājuma (šajā uzdevumā - mazākā skaitļa) atrašana starp visiem iespējamajiem variantiem ir vispārzināms uzdevumu veids, ko sauc par *optimizāciju*. Bieži tiek pielietoti paņēmieni, kas ļauj sašaurināt derīgo risinājumu apgabalu un tādejādi paātrināt risinājuma atrašanu. Šajā gadījumā *rijīgi* tiek atrasts mazākais skaitlis, kurš tiks izmantots skaitlī kā pirmais cipars (atradīsies visvairāk pa kreisi) un tad pēc *skaldi-un-valdi* pieejas šis pats uzdevums tiek risināts klucīšiem, kas atrodas virs klucīša ar mazāko skaitli

Puķu kaste (interaktīvs)

Vācija

Uzdevuma apraksts

Benam patīk puķes un viņš istabā ir izveidojis puķu kasti, kurā var ievietot 3x3 puķes. Viņam ir trīs krāsu puķes: sarkanas, dzeltenas un oranžas. Tagad Bens cenšas kastē izveidot visskaistāko puķu izvietojumu.

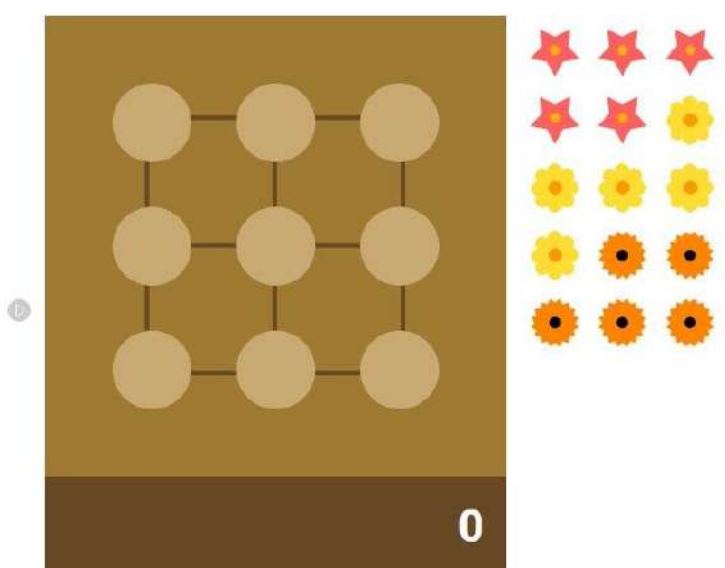
Katram puķu izvietojumam Bens dod skaitlisku novērtējumu - punktus. Lai to aprēķinātu, katrai puķei viņš apskata puķes, kas no tās atrodas pa labi, pa kresi, uz augšu un uz leju.

- Ja sarkana un dzeltena puķe atrodas blakus, tad Bens punktu kopsummai pieskaita trīs punktus.
- Ja dzeltena un oranža puķe atrodas blakus, tad Bens punktu kopsummai pieskaita vienu punktu.
- Citi gadījumi punktu skaitu neietekmē.

Puķu izvietojumā ir jābūt visu trīs krāsu puķēm.

Uzdevums

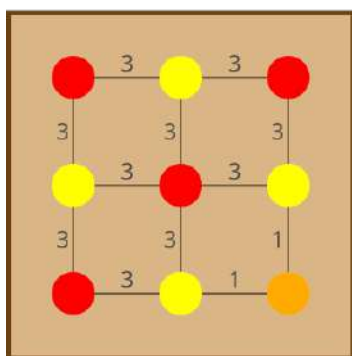
Izveidojiet puķu izvietojumu ar lielāko iespējamo punktu kopsummu. Pārvietojiet puķes no labās puses uz kastes tukšajām vietām. Jūsu izvietojuma punktu skaits būs redzams puķu kastes apakšā.



Atbildes izskaidrojums

Ievērojiet, ka Bens punktus dod par attiecīgās krāsas puķu pāriem. Savienosim blakus esošās puķes ar nogriežņiem, iegūstot 12 nogriežņus. Vislielāko punktu skaitu var iegūt, ja katrs nogrieznis savienos sarkanu un dzeltenu puķi un, tātad, būs trīs punktu vērtībā. To iespējams panākt, ja sarkanās un dzeltenās puķes izvieto pamīšus (šaha veidā). Bet šajā gadījumā laukumā nebūs neviena oranža puķe, kas ir pretrunā ar uzdevuma nosacījumu, ka laukumā jābūt visu trīs krāsu puķēm.

Aplūkojot puķu pārus savienojošos nogriežņus, redzams, ka vismazākais piesaistīto nogriežņu skaits ir stūra puķām - divi vienā stūrī sarkano puķi aizvietosim ar oranžo, tad tiks zaudēti četri punkti, bet visi nogriežņi joprojām dos punktu pienesumu - 10 nogriežņi pa trim, bet divi pa vienam punktam - kopā 32 punkti, kas arī ir lielākais iegūstamo punktu skaits:



Tā ir informātika!

Tā ir tik cilvēcīgi - atrast nevis tikai kaut kādu uzdevuma risinājumu, bet labāko iespējamo. To sauc par optimizāciju, kas ir noteikusi cilvēku uzvedību. Optimizācija ir ikdienas sastāvdaļa: ja vēlaties iegādāties konkrētu datoru modeli, jūs meklējat tādu pārdevēju, kurš piedāvā viszemāko cenu. Ja jums jābrauc no A līdz B, jūs vēlaties izmantot vislētāko maršrutu? Bet kā tiks aprēķinātas izmaksas: nauda, kas tiks iztērēta braucienam, garums kilometros, ceļojumam nepieciešamais laiks?

Lai veiktu optimizāciju, jums nepietiek zināt, kā izskatās jūsu problēmas risinājums. Nepieciešama arī precīza funkcija, kas var noteikt katra risinājuma izmaksas vai vērtību. Tad risinājums ar minimālu vai maksimālu vērtību (kas ir atkarīgs no funkcijas) ir optimāls. Šajā Bebru uzdevumā Bena metode, kā aprēķināt ziedu kompozīciju punktu skaitu, ir optimizācijas funkcija.

Matemātika gadsimtiem ilgi ir pētījusi optimizāciju. Informātikā liela uzmanība ir pievērsta matemātisko algoritmu ieviešanai, jo datorus var izmantot, lai veiktu

optimizāciju tik liela mēroga uzdevumos, kas cilvēkiem vairs nav pa spēkam. Datorzinātnieki ir atklājuši arī, ka daudzas optimizācijas problēmas ir patiešām grūti atrisināt. Īpaši šādām problēmām ir izgudrotas metodes (tā sauktās heuristikas), kas ne vienmēr atradīs optimālu atrisinājumu, bet tādu, kas ir tuvu tam.

Koku sudoku (interaktīvs)

Šveice

Uzdevuma apraksts

Bebri lauks ir sadalīts 9 kvadrātos, kurus veido 3 x 3 rūtiņu režģi. Katrā rūtiņā nepieciešams iestādīt vienu koku, kura augstums var būt 1(), 2() un 3 ().

Katrā līnijā (horizontālā un vertikālā) ir tieši viens no katra garuma kokiem (nevienā līnijā nav 2 vienādu augstumu koku).

Bebri novēroja, cik kokus viņi var redzēt no katras pozīcijas, kurai pievienota zīme. Tad viņi uzrakstīja šo skaitli uz zīmes.

( zīmes vertikālajām līnijām un  zīmes horizontālajām līnijām).

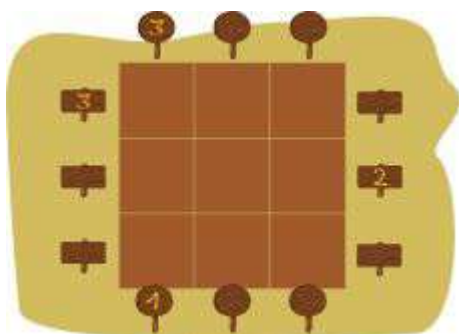
Skatoties uz koku līniju, bebbri nevar redzēt īsākus kokus, kas sastādīti aiz garākiem.



Uzdevums

Diemžēl lietuss noskalojis skaitļus no dažām zīmēm.

Ievietojiet kokus laukumā, par rādītāju izmantojot uz zīmēm atlikušos skaitļus.



Atbildes izskaidrojums

Ja viens no bebbiem, raugoties nosavas vietas malā, redz visus kokus, koki no šī punkta jāsakārto augošā secībā. Tas ļauj uzreiz novietot kokus pirmajā rindā un pirmajā kolonnā.



Pārējos kokus var savietot, atbilstoši "Sudoku" principam. Tā kā katrā no rindām un kolonnām drīkst būt pa vienam no katra augstuma koka, ir tikai viens iespējamais variants pārējo koku izvietojuma. Otrās rindas labajā pusē jānovieto zemākā veida koks 1 (🌲), jo trešajā rindā jau ir 2 (🌲), bet trešajā kolonnā jau novietots 3 (🌲). Tādejādi apakšējā labajā pusē jāievieto 2 (🌲), jo trešajā kolonnā jau ir koki 1 (🌲) un 3 (🌲).

Tātad gala risinājums izskatās šādi:



Tā ir informātika!

Šis uzdevums atsaucas uz divām datorzinātniekiem būtiskām spējām. Pirmā ir spēja atrast tādu uzdevuma atrisinājumu, kas apmierina dotos nosacījumus un ierobežojumus. Otrā ir spēja atjaunot objekta vai situācijas struktūru no daļējas informācijas par objekta vai situācijas īpašībām. Šīs spējas var pielietot, ja informācija par objektiem ir dota saspiestā formā. Papildus ir jāvar ievērot dotos noteikumus un atklāt kļūdas un pretrunas datu reprezentācijā.

Kļūdu korekcija ir vispārzināma datorzinātnē izmantojama tehnika, kas ļauj pārliecināties par datu, kas pārsūtīti pa vairākiem informācijas kanāliem, kuros iespējama datu nozaudēšana vai izkropļošana, uzticamību. Kļūdu labošanas tehnika iekļauj nepareizo datu labošanu, lai atjaunotu tos oriģinālajā un pareizajā izskatā.

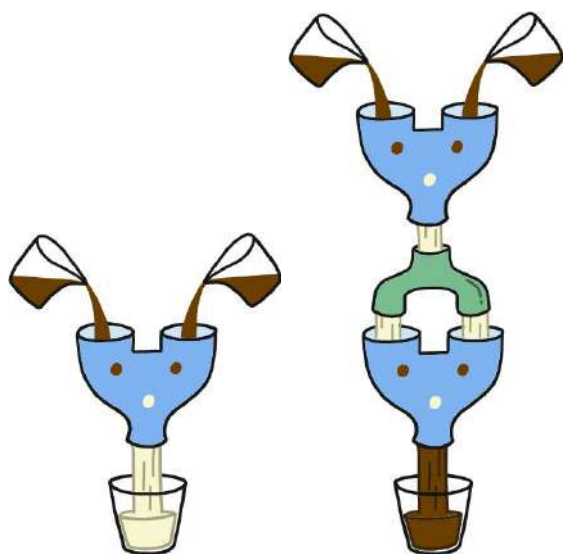
Maģiskā dzērienu pagatavošanas mašīna 2

Dienvīdkoreja

Uzdevuma apraksts

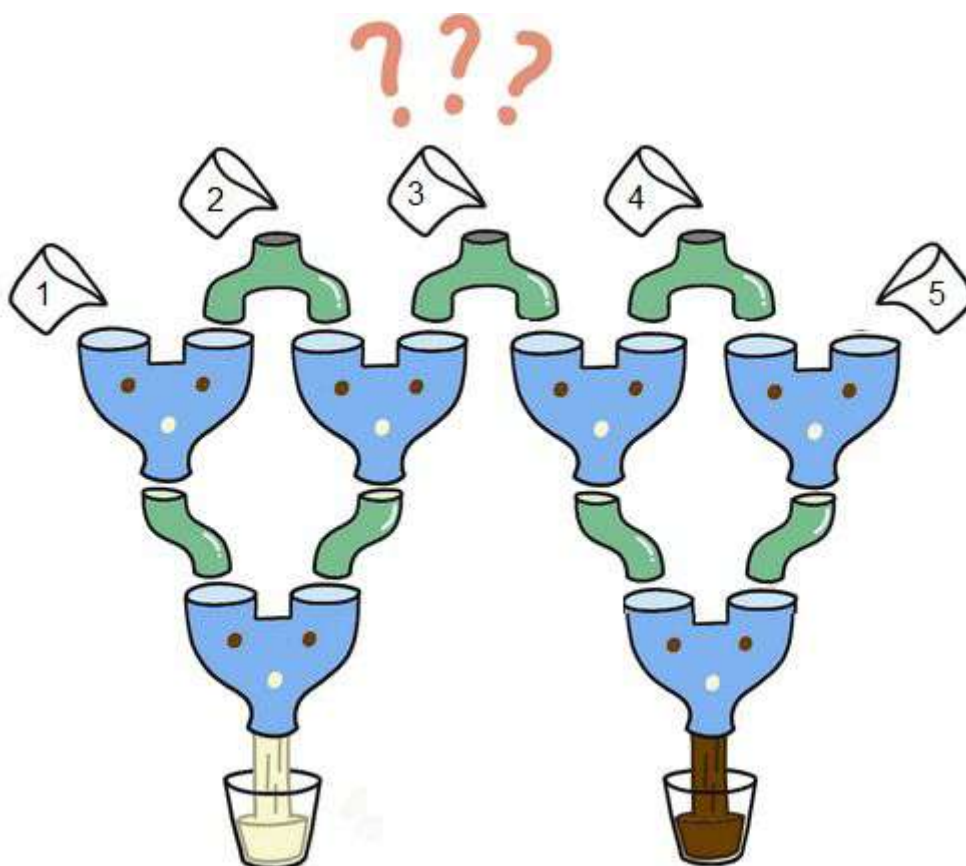
Bebram Kobi pieder 1. zīmējumā redzamā maģiskā dzērienu pagatavošanas mašīna. Mašīna ir zilā krāsā un tai ir divas piltuves. Mašīna darbojas šādi:

- Ja bebrs abās piltuvēs ielej šokolādes pienu, no mašīnas laukā izlīst piens.
- Ja bebrs kādā no piltuvēm ielej pienu, no mašīnas laukā izlīst šokolādes piens.
- Ja bebrs savieno divas mašīnas un abās augšējās mašīnas piltuvēs ielej šokolādes pienu kā parādīts 2. zīmējumā, tad no apakšējās mašīnas izlīst šokolādes piens. Ievērojiet, ka zaļie savienotājelementi neietekmē šķidruma veidu.



1. zīm.

2. zīm.



3. zīm.

Jautājums

Kobi no piecām krūzītēm ielēja 3. zīmējumā attēlotās augšējo mašīnu piltuvēs dažāda veida pienu un ieguva, ka no kreisās apakšējās mašīnas izlīst piens, bet no labās - šokolādes piens. Izrādījās, ka tikai vienā krūzītē bija piens, bet četrās pārējās - šokolādes piens. Kurā krūzītē bija piens?

Atbildes forma

Naturāls skaitlis no 1 līdz 5 - krūzītes numurs.

Pareizā atbilde: 2.

Atbildes izskaidrojums

Ja piens būtu trešajā krūzītē, tad, simetrijas dēļ, rezultāts abās apakšējās mašīnās būtu vienāds - pretruna. Tātad vienam "mašīnu kompleksam" (1-2-3 vai 3-4-5) augšā no visām trim krūzītēm tiek liets šokolādes piens. Ja no visām trim krūzītēm ielej šokolādes pienu, tad vidējā līmenī dabū vienīgi pienu, kas nozīmē, ka rezultātā iegūst šokolādes pienu. Tātad, šokolādes piens tiek liets no 3., 4. un

5. krūzītes. Ja no 1. krūzītes lietu pienu, tad no 2. būtu jālej šokolādes piens un kreisais mašīnu komplekss prasītā piena vietā ražotu šokolādes pienu.

Tāpēc vienīgā iespēja, ka šokolādes piens bija 2. krūzītē. Pārbaudām kreiso mašīnu kompleksu: ielej "šokolādes piens - piens - šokolādes piens" -> nākamajā līmenī iegūst "šokolādes piens - šokolādes piens" un apakšējā "piens", kas arī bija nepieciešams.

Tā ir informātika!

Loģiskā shēmā notiek darbošanās ar loģiskām vērtībām (paties vai aplams, vai, analogiski, 1 vai 0) un tiek izmantoti loģiskie operatori. Klasiskie loģiskie operatori ir "un" (AND), "vai" (OR), "nē" (NOT), "izslēdzošais vai" (XOR).

Šajā uzdevumā tiek izmantota mazāk zināms loģiskais operators "ne un" (NAND), kuru definē kā

- $A \text{ NAND } B = \text{NOT } (A \text{ AND } B)$

Interesanti, ka klasiskos operatorus var izteikt ar NAND palīdzību:

- $\text{NOT } A = A \text{ NAND } A$
- $A \text{ AND } B = (A \text{ NAND } B) \text{ NAND } (A \text{ NAND } B)$
- $A \text{ OR } B = (A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B)$
- $A \text{ XOR } B = (A \text{ NAND } (A \text{ NAND } B)) \text{ NAND } (B \text{ NAND } (A \text{ NAND } B))$

Šajā uzdevumā piens atbilst vērtībai 0, šokolādes piens - vērtībai 1, bet mašīna realizē operatora NAND izpildi. 2. zīmējumā ir realizēta AND izpilde.

3. zīmējumā ir attēlota sarežģītāka operatora izpilde izmantojot vienīgi operatoru NAND (izmantota divreiz ar dažādiem mainīgajiem):

- $(A \text{ OR } C) \text{ AND } B = (A \text{ NAND } B) \text{ NAND } (B \text{ NAND } C)$
- $(C \text{ OR } E) \text{ AND } D = (C \text{ NAND } D) \text{ NAND } (D \text{ NAND } E)$

Bebru tikšanās

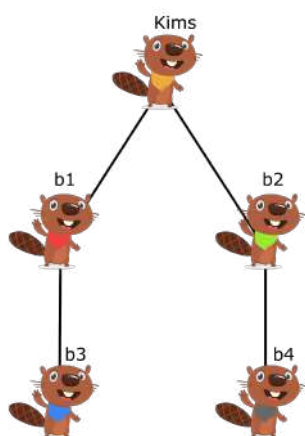
Taizeme

Uzdevuma apraksts

Bebri Kims regulāri apmeklē iknedēļas tirdziņu. Iepirkšanās laikā viņš satiek divus citus bebrus - b1 un b2. Bebri b1 un b2 katrs arī tirdziņā satiek divus citus bebrus. Cik bebbiem bija jābūt tirdziņā, lai Kims, b1 un b2 katrs būtu varējuši satikt pa diviem bebbiem?

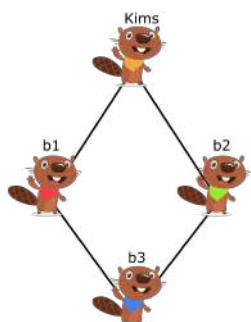
Attēlosim kā bebbi viens otru ir satikuši ar zīmējumu palīdzību. Ir iespējami vairāki varianti:

1. Zīmējums

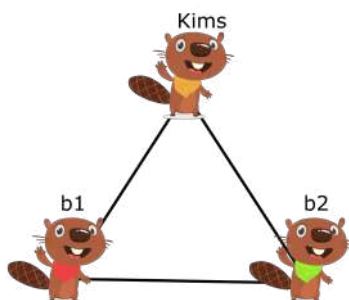


Kims satika b1 un b2, b1 satika Kimu un b3, b2 satika Kimu un b4. Šajā gadījumā tirdziņā bija jābūt pieciem bebbiem.

Divi citi iespējami varianti:



2. zīmējums



3. zīmējums

Tirdziņā jābūt četriem (2. zīm.) vai trim (3. zīm.) bebbiem.

Jautājums

Nākamajā nedēļā Kims atkal devās uz tirdziņu un tur satika trīs citus bebrus. To bebru skaits, ko bija satikuši šie trīs bebri, bija attiecīgi 2, 5 un 5.

Kāds **mazākais** bebru skaits varēja būt tirdziņā?

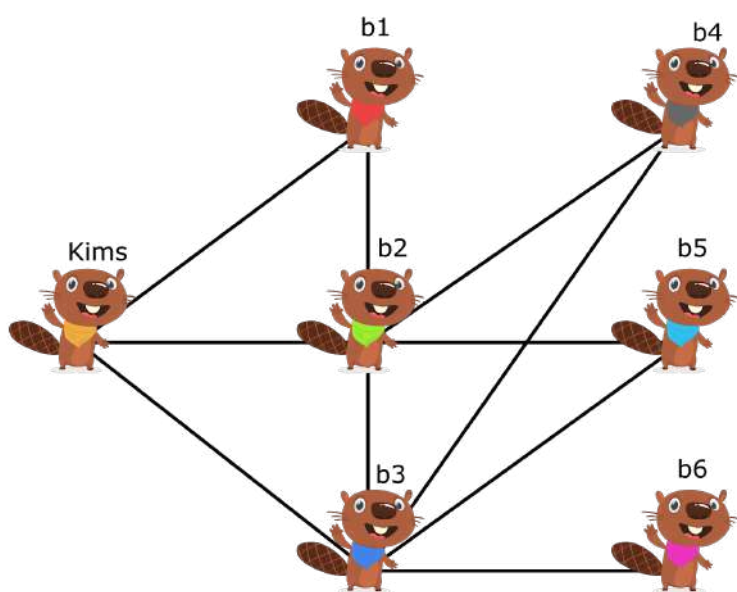
Atbilžu varianti

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

Pareizā atbilde: D)7

Atbildes izskaidrojums

Saskaitot tikai Kimu un tos bebrus, kurus Kims ir saticis (b1, b2 un b3) mēs jau esam saskaitījuši četrus bebrus.



Tagad aplūkosim bebru b1, kurš ir saticis divus bebrus. Pieņemsim, ka b1 ir saticis Kimu un kādu no bebbiem b2 vai b3. Joprojām mazākais bebru skaits tirdziņā ir 4.

Bebrs b2 ir saticis piecus bebrus. Mēs zinām, ka b2 satika Kimu.

Aplūkosim iespējamus gadījumus:

A. b2 satika b1.

- a. b2 satika b3 - ir nepieciešami vēl divi beбри (b4 un b5), kurus b2 būtu saticis. Tātad mazākais bebru skaits tirdziņā ir vismaz 6. Bet bebram b3 bija jāsatiek pieci beбри - Kims, b2 un vēl trīs. b1 neder, jo viņš jau divus ir saticis. Ja b3 satiek b4 un b5, tad ir nepieciešams vēl viens bebris (b6), lai izpildītu visus nosacījumus. Kopējais bebru skaits tirdziņā ir 7.
- b. b2 nesatika b3 - ir nepieciešami trīs beбри (b4, b5 un b6), lai b2 būtu saticis nepieciešamo bebru skaitu. b3 bija jāsatiek pieci beбри - Kims un vēl četri, starp kuriem nav b1 un b2. Tātad kopējais bebru skaits tirdziņā šajā gadījumā ir 8.

B. b2 nesatika b1.

- a. b2 satika b3 - ir nepieciešami vēl trīs beбри (b4, b5 un b6), kurus b2 būtu saticis. b3 var būt saticis Kimu, b1, b2, b4 un b5. Visi nosacījumi ir izpildīti un kopējais bebru skaits tirdziņā ir 7.
- b. b2 nesatika b3 - ir nepieciešami vēl četri beбри (b4, b5, b6 un b7), kurus b2 būtu saticis. b3 var būt saticis Kimu, b1, b4, b5 un b6. Kopējais bebru skaits tirdziņā ir 8.

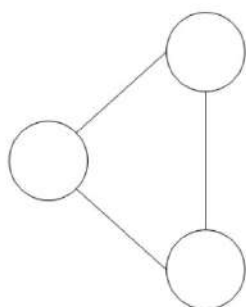
Kā redzams pēc visu gadījumu apskatīšanas, mazākais bebru skaits tirdziņā ir 7.

Zīmējumā ir attēlots viens no derīgiem satikšanās variantiem ar septiņiem bebrim.

Tā ir informātika!

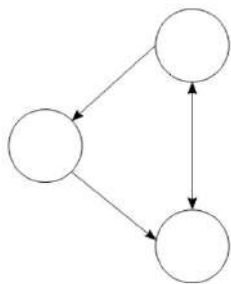
Šī uzdevuma risināšanā mēs izmantojam datu struktūru "grafs", kas sastāv no "mezgliem" (tos sauc arī par virsotnēm) un "šķautnēm" (sauktas arī par saitēm). Ir divu veidu grafi:

(1) neorientēti grafi, kur šķautnes savieno mezglus simetriski, t.i., starp mezgliem nav hierarhijas un savienojumam nav virziena;



un

(2) orientēti grafi, kur šķautnes mezglus saista asimetriski, t.i., starp mezgliem pastāv potenciāla hierarhija un savienojums ir vērsts noteiktā virzienā.



Šī uzdevuma risināšanā mēs varam attēlot tirgus ainu kā neorientētu grafu, kur katru bebru attēlo mezgls un viņa / viņas tikšanos ar citu bebru - šķautne. Grafam jābūt neorientētam, jo mēs pieņemam, ka divu bebru satikšanās ir simetriska (ja A ir saticis B, tad arī B ir saticis A).

Orientētu grafu varētu izmantot citā tirgus scenārijā, kur bebri nesatiekas, bet citus bebrus redz tikai no tālienes: Šajā gadījumā bebrs b1 var redzēt citu bebru b2, kurš savukārt neredz b1, bet redz citus bebrus b3 un b4 .

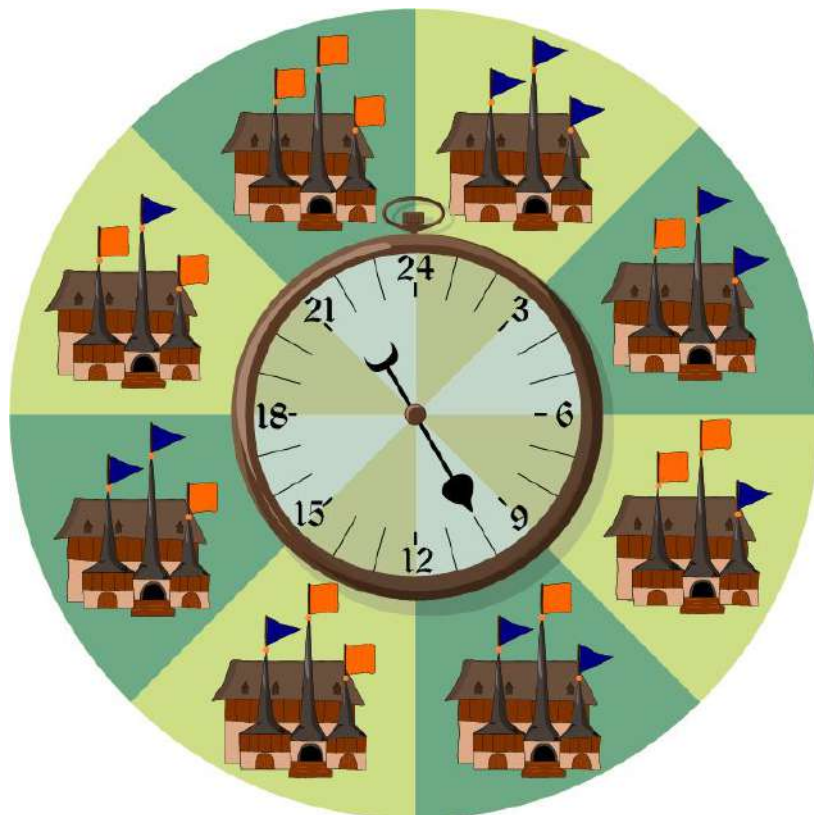
Grafi ļauj attēlot sarežģītas parādības vienkāršā un abstraktā veidā, atspoguļot vairākas attiecības starp dažādiem elementiem, tādējādi palīdz atrisināt daudzas reālās dzīves problēmas. Piemēram, grafi tiek izmantoti, lai attēlotu dažādus tīklus. Tīkli var ietvert ceļus pilsētā, telefonu tīklos vai elektrisko ķēžu tīklus. Grafus izmanto arī sociālajos tīklos, piemēram, LinkedIn vai Facebook. Piemēram, Facebook katru personu attēlo kā mezglu ar savu struktūra un satur informāciju par, piemēram, personas ID, vārdu, dzimumu, atrašanās vietu utt.

Bebru torņu pulkstenis (interaktīvs)

Šveice

Uzdevuma apraksts

Mazā, dīvainā bebru ciematā diena ir sadalīta astoņos trīs stundu garos intervālos, un bebrus katrā brīdī interesē tikai tas, kurš laika intervāls ir šobrīd. Lai noteiktu laiku, viņiem pietiek paraudzīties uz rātsnama torņiem, kuros trīs uzvilktie karogi norāda šībrīža laika intervālu. Ir divu veidu karogi - oranži taisnstūrveida un zili trīsstūrveida. Katram laika intervālam atbilst atšķirīgs karogu komplekts (kā parādīts zīmējumā).



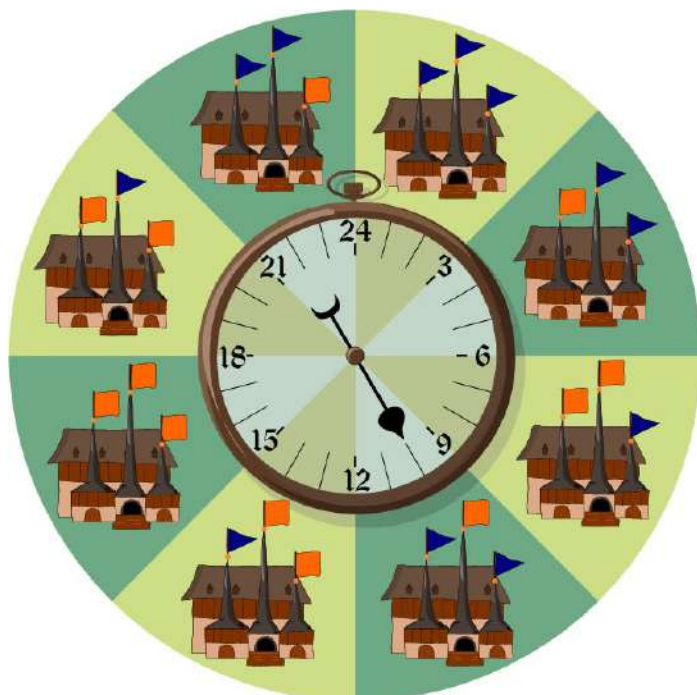
Šī sistēma ir nepilnīga - pirmajām četrām pārejām no viena laika intervāla uz nākamo pietiek nomainīt karogu vienā tornī, bet pārejām pārejām ir nepieciešama vairāku karogu nomaiņa.

Uzdevums

Izveidojiet vienu tādu intervālu attēlošanas sistēmu ar trim dotā veida karogiem, kurā **visas pārejas** starp laika intervāliem būtu iespējams veikt nomainot tikai vienu karogu. Šādas sistēmas ir vairākas - izveidojiet jebkuru no tām.

Atbildes izskaidrojums

Šis ir viens no daudziem iespējamiem atrisinājumiem:

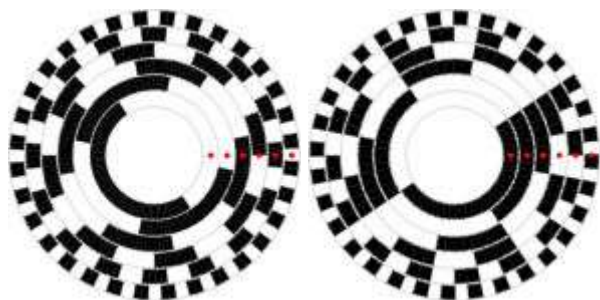


Intervālu kodēšanai izmantosim bināros skaitļus, kur ar 0 apzīmēsim zilu, bet ar 1 - sarkanu karogu.

Tad astoņiem intervāliem atbilstošie trīsciparu binārie skaitļi ir šādi: 000, 001, 010, 011, 100, 101, 110, 111. Šos skaitļus nepieciešams izvietot aplī tā, lai katri divi blakus skaitļi atšķirtos par tieši vienu ciparu. Varam rīkoties šādi. Vispirms aplūkosim divus viencipara skaitļus 0 un 1. Tiem prasītā īpašība ir spēkā. No šiem skaitļiem izveidosim divus divciparu skaitļus, abiem pierakstot priekšā ciparu 0: 00 un 01. Prasītā īpašība (nomainīts tikai viens cipars) joprojām ir spēkā. Tagad nākamās skaitļus veidojam no jau izveidotajiem, nomainot vecāko ciparu uz 1 un skaitļus ņemot pretējā secībā: 00, 01, 11, 10. Esam ieguvuši divciparu skaitļus ar nepieciešamo īpašību. Tāpat turpinām, lai iegūtu trīsciparu skaitļus: Vispirms vienkārši pierakstām priekšā 0, bet pēc tam 1 un ņemam pretējā secībā: 000, 001, 011, 010, 110, 111, 101, 100. Tā kā numerāciju var sākt no jebkura intervāla un pozīciju kārtību var izvēlēties patvaļīgi, tad derīgu sistēmu ir daudz un šī ir tikai viena no tām.

Tā ir informātika!

Šādu kodējumu sauc par Greja kodu, un tam ir daudz praktisku pielietojumu. Tas, ka starp secīgiem koda vārdiem mainās tikai viens bits, var, piemēram, samazināt enerģijas patēriņu. Pārslēdzot vairāk nekā vienu bitu, noteikti tiks patērēts vairāk enerģijas, un parastajā binārajā kodā vienubrīd visu bitu vērtības tiek nomainītas vienlaicīgi.



Zināms Greja koda pielietojums inženierzinātnēs ir rotējoša diska leņķa mērīšana. Uz diska tiek uzzīmēts Greja kods, kā parādīts attēlā pa kreisi. Gaismas sensoru masīva uzstādīšana, kas fiksētā stāvoklī var atšķirt melnbaltās krāsas (parādīta sarkanā krāsā), ļauj mums nolasīt bināro skaitli, kas kodē diska, kas rotē zem sensoriem, pašreizējo leņķi.

Attēlā pa kreisi mēs varam redzēt, kā gaismas sensori nolasā koda vārdu 001010 vai 001110, jo ceturtais gaismas sensors atrodas tieši malā starp baltu un melnu joslu. Jebkurā gadījumā nav nodarīts būtisks kaitējums, jo patiesais leņķis atrodas tieši starp leņķiem, kurus kodē šie divi koda vārdi. Izmantojot citu kodu secību, situācija varētu būt krietni nepatīkamāka. Piemēram, parastajā binārā kodā, kas attēlots attēlā pa labi, koda vārds 111010 seko koda vārdam 111001. Ja disks atrastos starp šiem leņķiem, sensori varētu nolasīt koda vārdu 111011, kas kodē nedaudz atšķirīgu leņķi. Sliktākajā gadījumā sensori var nolasīt robežu starp pilnīgi baltu koda vārdu 000000 un pilnīgi melnu koda vārdu 111111. Šajā gadījumā visi sensori var jebkurā laikā pārslēgties starp 0 un 1 un novest pie ziņojuma par pilnīgi patvaļīgu leņķi.

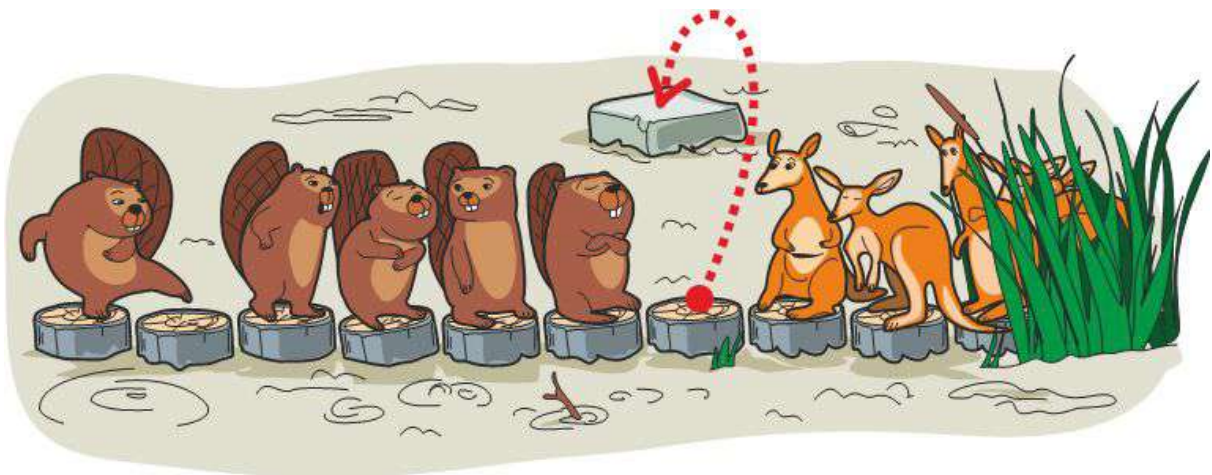
Bebri pret ķenguriem

Lietuva

Uzdevuma apraksts

Šķērsojot purvu pa garu taku, bebri ir satikuši uz tās ķenguru grupu, kas dodas pretējā virzienā. Neviens nevēlas kāpt nost no takas, riskējot palikt slapjš un netīrs. Ķenguri ir atraduši akmeni, kas atrodas netālu no takas un uz kuru jebkurš ķengurs var gan aizlēkt, gan no tā atlēkt atpakaļ. Diemžēl katrā brīdī uz akmens var atrasties ne vairāk kā viens ķengurs.

Ķenguriem un bebriem nav iebildumu arī pakāpties atpakaļ nepieciešamo soļu skaitu. Izņēmums ir Fredis - bebris, kas gāja bebru grupas priekšgalā un pirmais satika ķengurus. Fredis ir ar mieru spert soli atpakaļ ne vairāk kā 10 reizes.



Jautājums

Zinot Freža raksturu, cik ķenguru varēs tikt garām Fredim un nekāpties atpakaļ?

Atbilžu varianti

- A) Fredim garām var tikt neierobežots daudzums ķenguru,
- B) Tieši 10 ķenguri var tikt garām Fredim.
- C) Tieši 6 ķenguri var tikt garām Fredim.
- D) Tieši 5 ķenguri var tikt garām Fredim.
- E) Tieši 4 ķenguri var tikt garām Fredim.
- F) Mazāk nekā 4 ķenguri var tikt garām Fredim.
- G) Precīzu ķenguru skaitu noteikt nav iespējams

Pareizā atbilde: C) Tieši 6 ķenguri var tikt garām Fredim.

Atbildes izskaidrojums

Risinot uzdevumu, visus bebrus, izņemot Fredi, var ignorēt, jo viņiem nav iebildumu arī pakāpties atpakaļ.

Lai Fredis atļautu ķenguram tikt garām, jārīkojas šādi:

Ķengurs pārlec uz akmens:	
Fredis pāiet divus soļus uz priekšu:	
Ķengurs pārlec atpakaļ uz taku un var turpināt ceļu:	
Fredis pāiet divus soļus atpakaļ lai ļautu nākamajam ķenguram pārlēkt uz akmens:	

Izpildot minēto soļu secību piecas reizes, Fredis var palaist garām piecus ķengurus, kopā atkāpjoties 10 soļus. Tad papildus vēl viens ķengurs var tikt garām Fredim, jo viņš tobrīd atradīsies sākumpozīcijā, lai ļautu ķenguram tikt garām. Tātad pavisam Fredim, ja viņš atkāpsies 10 soļus, garām var tikt seši ķenguri.

Ķenguru skaitu var izteikt ar matemātiskas formulas palīdzību. Lai Fredis palaistu garām k ķengurus, viņam kopumā nāksies atkāpties $s = 2 \times (k - 1)$ soļus. Izsakot pretējā virzienā, formula ir $k = 0.5 \times s + 1$.

Tā ir informātika!

Algoritmi ir svarīgi, lai datori apstrādātu datus un izpildītu uzdevumā nepieciešamās darbības noteiktā secībā:

- Mainīgā satura maiņa: katrs bluķītis uz takas un akmens ir vieta, kur informāciju var glabāt tāpat kā programmu mainīgajos, un beбри un ķenguri ir dati, kas "jāuzglabā" šajās vietās.
- Pārvietošanos (pakāpšanos) secība: Lai ķenguri liktu garām beбriem, tiem jāpārvietojas noteiktā veidā. Tā kā viņi nevar uzreiz nokļūt vajadzīgajā vietā, nepieciešams izveidot noteiktu darbību virkni, kā rīkoties katrā situācijā. Un tad šo virkņu atkārtota pielietošana palīdzēs atrisināt sākotnējo uzdevumu.
- Nepieciešamo darbību atkārtošana tik reižu, cik nepieciešams: Šajā gadījumā viena un tā pati darbību virkne tiek atkārtota vairākas reizes, kas arī ir tipisks skaitļošanas jēdziens: atrisiniet nelielu problēmu vienreiz un pielietojiet risinājumu tik reižu, cik nepieciešams.

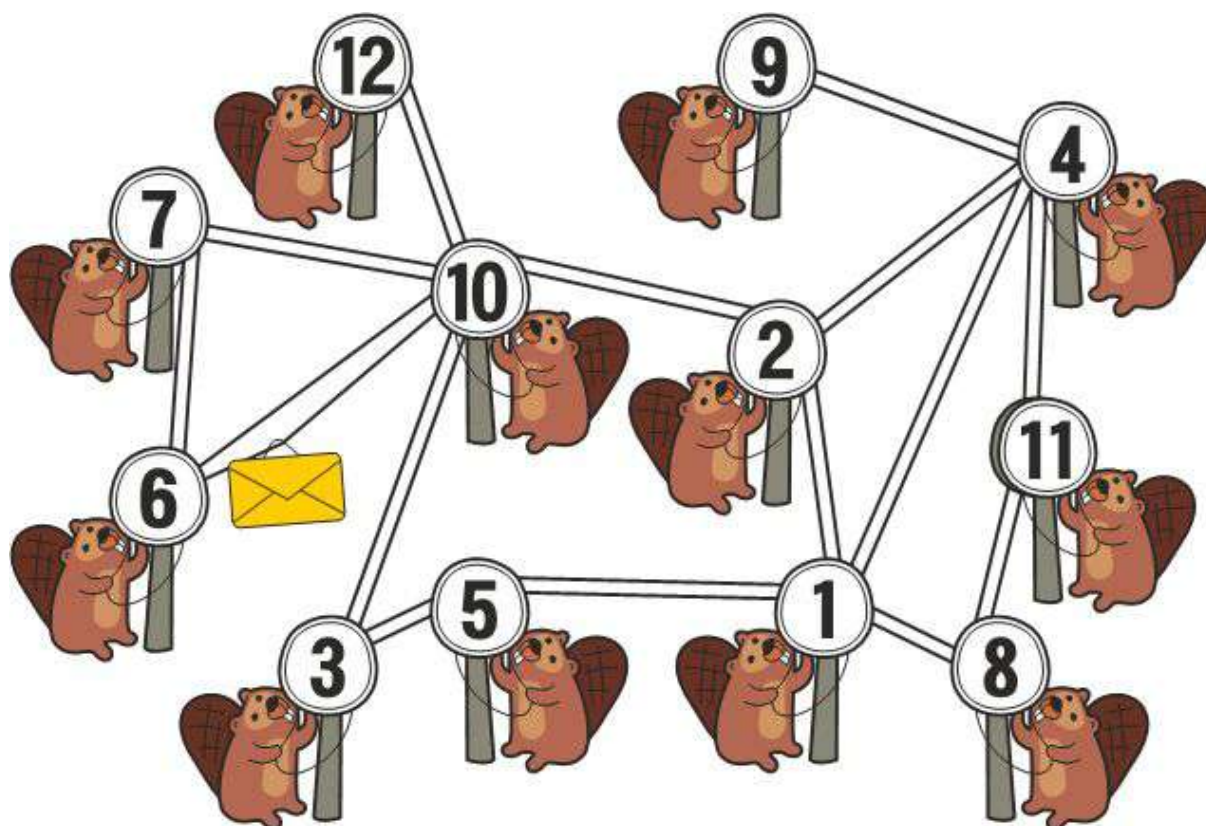
Tādējādi par algoritmu var uzskatīt jebkuru darbību secību, ieskaitot atkārtojumus, kurus datorsistēmas var izpildīt. Atpazītos algoritmu modeļus (līdzīgu darbību virknes, kas tiek atkārtotas) var pārvērst par atkārtoti lietojamu efektīvu uzdevuma automātiskas risināšanas kodu (kā šajā uzdevumā). Bluķīši uz takas un akmens ir kā reģistri datora procesorā, kurā var uzglabāt datus.

Ziņojumu izplatīšana (interaktīvs)

Ziemeļmaķedonija

Uzdevuma apraksts

Bebri kolonijā dzīvo 12 beбри. Katrs no beбriem dzīvo savā alā, bet alas ir savienotas ar virvēm kā parādīts zīmējumā. Virves tiek izmantotas ziņojumu nosūtīšanai.



Bebri vēlas pēc iespējas ātrāk uzzināt jaunumus. Tiklīdz kāds beбrs uzzin ko jaunu, viņš tūlīt izmanto savā alā pieejamās virves lai apziņotu citus beбrus. Piemēram, ja 8. beбrs uzzin ko jaunu, viņš par to paziņos 1. un 11. beбram (pirmais solis). Nākamie, kam tiks paziņots šis jaunums, būs 2., 4. un 5. beбrs (otrais solis). Tā šī jaunā ziņa tiks izplatīta tālāk, līdz kamēr visi beбri to zinās.

Jautājums

Kuram no beбriem jums jāpastāsta ziņa, lai visi beбri šo ziņu uzzinātu pēc iespējas ātrāk (pēc iespējas mazāk soļos)?

Atbildes forma

Naturāls skaitlis robežās no 1 līdz 12.

Pareizā atbilde: 2.

Atbildes izskaidrojums

Ja ziņa tiek pastāstīta 2. bebram, tad visi bebri šo ziņu uzzinās divos soļos: pirmajā ziņu uzzinās 4., 1. un 10. bebrs, bet otrajā - visi pārējie. Nevienai citai alai nav spēkā īpašība, ka tā būtu ne tālāk kā divus soļus no citām.

Tā ir informātika!

Daudzas reālās dzīves problēmas datorprogrammas var atrisināt pēc to pārveidošanas grafa formā. Grafs sastāv no virsotņu vai mezglu kopas (parasti attēlo kā punktus) un šķautņu kopas (parasti attēlo kā līnijas segmentus, iespējams, izliektu), kas šīs virsotnes savieno. Šajā uzdevumā galvenā problēma ir tā sauktā grafa centra atrašana.

Grafa centrs ir visu virsotņu u kopa, kur lielākais attālums $d(u, v)$ līdz citām virsotnēm v ir minimāls. Šeit bija tikai viena šāda virsotne - 2. pilsēta.

Grafa centra atrašana ir noderīga uzdevumos, kur mērķis ir samazināt sliktāko (tālāko iespējamo) attālumu līdz objektam. Piemēram, novietojot slimnīcu centrā, tiek samazināts lielākais iespējamais ātrās palīdzības brauciena attālums.

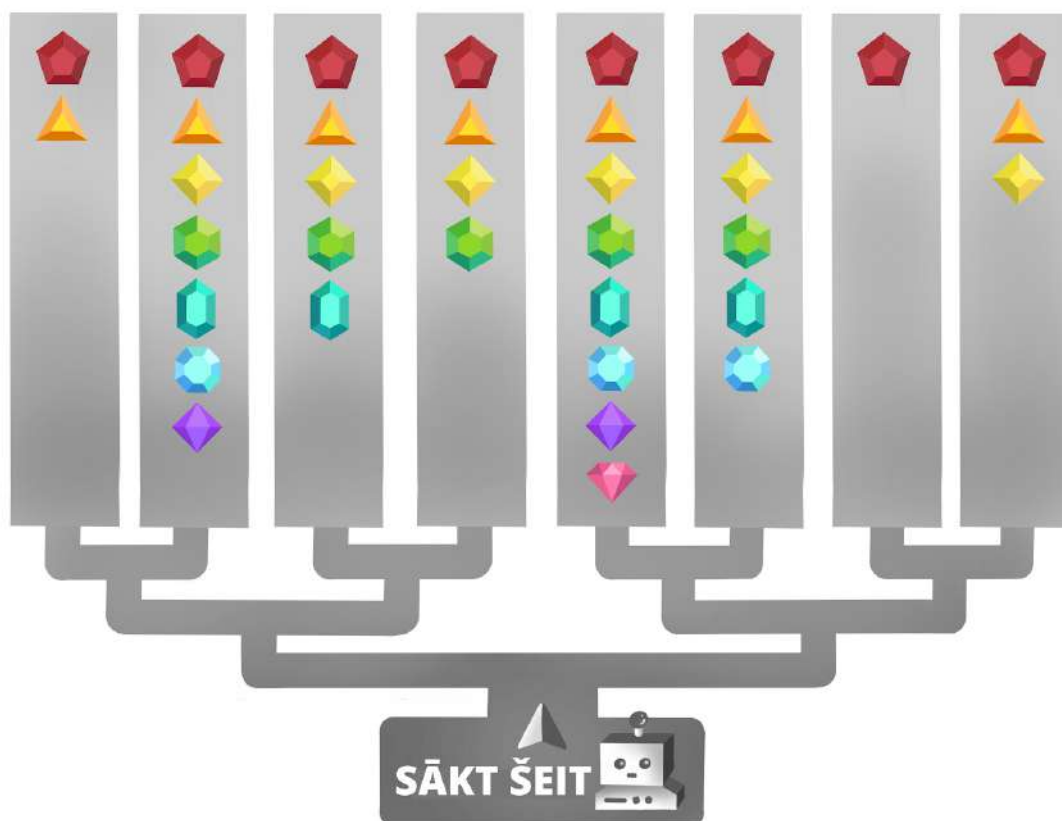
Ja mums ir tik mazs grafs kā šajā uzdevumā, var izmantot izmēģinājumu un kļūdu metodi, jo ir viegli aprēķināt attālumu starp visiem virsotņu pāriem. Lieliem grafiem centru var atrast izmantojot sarežģītākus algoritmus, piemēram, Floida-Varšala algoritmu.

Robotu labirints (interaktīvs)

Filipīnas

Uzdevuma apraksts

Alise un Bobs vada robotu, kurš atrodas dārgakmeņu labirintā. Robots savu kustību sāk vietā, kas zīmējumā attēlotajā labirinta plānā atrodas pašā apakšā. Robots virzās augšup līdz kamēr sastop ceļa sazarojumu un gaida komandu uz kuru pusi (pa labi vai pa kreisi) pagriezties. Kad komanda saņemta, robots turpina kustību norādītajā virzienā līdz sasniedz nākamo sazarojumu, gaida komandu, utt.



Alise un Bobs komandas robotam dod pēc kārtas. Pirmajā sazarojumā komandu robotam dod Alise.

Robots beidz kustību kad sasniegta kāda dārgakmeņu krātuve un savāc visus šajā krātuvē esošos dārgakmeņus. Alise vēlas, lai robots savāktu iespējami lielāku dārgakmeņu skaitu, bet Bobs - lai iespējami mazāku.

Alisei un Bobam abiem ir zināms otra spēlētāja mērķis un viņi cenšas pārspēt viens otru viltībā. Piemēram, ja Bobs novirza robotu uz sazarojumu, kurš ved uz krātuvēm ar trim un septiņiem dārgakmeņiem, tad viņš var būt pārliecināts, ka Alise liks robotam izvēlēties virzienu ar septiņiem dārgakmeņiem.

Uzdevums

Alise spēli sāks pirmā. Norādiet to dārgakmeņu komplektu, kur robots apstāsies.

Pareizā atbilde: Robots apstāsies rindā ar pieciem dārgakmeņiem.

Atbildes izskaidrojums

Ir zināms, ka Alise izvēlēsies robota kustības virzienu pirmajā un trešajā gājienā, bet Bobs - tikai otrajā.

Varam pieņemt, ka robots jau ir nonācis līdz trešajam gājienam un varam analizēt, kuru no divām izvēlēm katrā no variantiem izdarīs Alise. Skaidrs, ka Alise izvēlēsies lielāko iespējamo dārgakmeņu skaitu. Tas nozīmē, ka pirmajā pāri tiks izvēlēts 7 (pa labi), otrajā - 5 (pa kreisi), trešajā - 8 (pa kreisi), bet ceturtajā - 3 (pa labi).

Bobs otrajā gājienā izvēlētos mazāko no iespējamajiem dārgakmeņu skaitiem - no pirmajiem diviem pāriem tas būtu 5 (pa labi), bet no otrajiem diviem - 3 (pa labi).

Alisei atliek izvēlēties lielāko no šiem diviem skaitļiem, kas ir 5 un pirmā komanda liks robotam pagriezties pa kreisi.

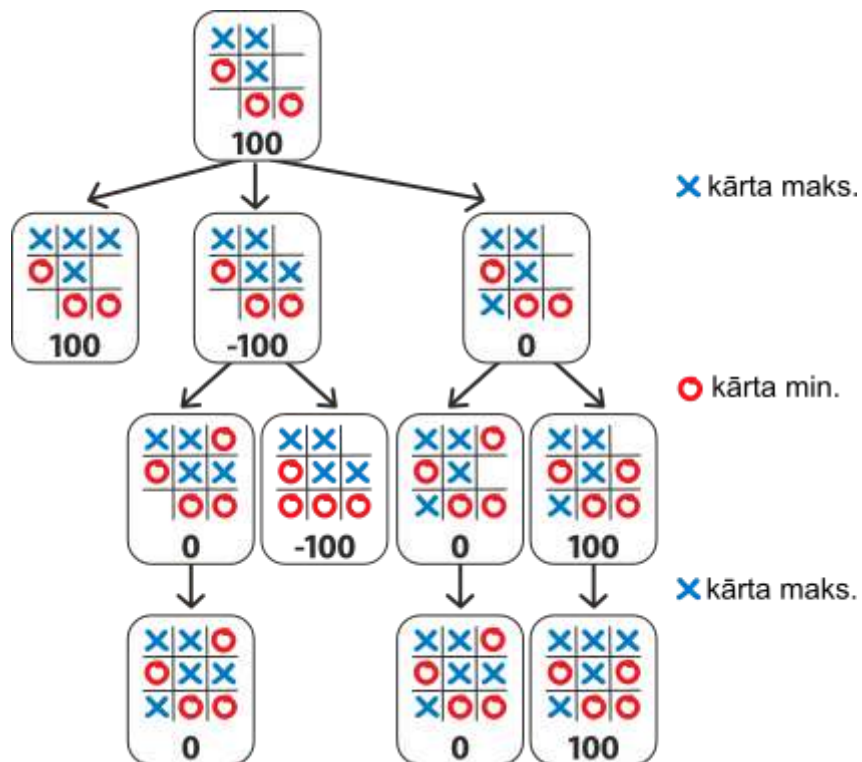
Tā ir informātika!

Ja labirintam ir tikai trīs līmeņi, uzdevuma atrisināšanai pietiek ar loģisku pamatojumu. Vairāk līmeņu gadījumā spēlējošajam algoritmam jāvar realizēt šī loģika lai izvēlētos labāko spēles turpinājumu. Pirms katras izvēles nepieciešams izanalizēt katra iespējamā gājiena turpmākās sekas un novērtēt tās skaitliskā izteiksmē (analogija šahā būtu uz šaha galdiņa katram spēlētājam atlikušo figūru skaits).

Pēc tam var izdarīt gājienu, kas maksimāli palielina rezultātu. Spēlējošais "mākslīgais intelekts" (MI) pieņem, ka pretspēlētājs ir "absolūti gudrs", un izdarīs tādu gājienu, kas visvairāk samazinās rezultātu (pat ja pretspēlētājs tik labi nespēlē). Tādējādi MI centīsies iegūt maksimālo punktu skaitu no minimālajiem. Vai arī, mainot lomas MI mēģinās iegūt minimālo no visiem maksimumu. Tāpēc

šādu algoritmu sauc par “minimaksa” algoritmu. Vairāku līmeņu gadījumā, atkarībā no tā, kam jāizdara gājiens, pārmaiņus tiks meklēts vai nu maksimums no minimumiem, vai arī minimums no maksimumiem.

Vienkāršā spēle “Desas” (jeb “krustiņi un nullītes”) ir lielisks šāda algoritma piemērs. Attēlā ir parādīta daļa no lēmuma koka aprēķināšanas, kas meklē vislabāko gājienu situācijā, kas attēlota zīmējuma pašā augšā.

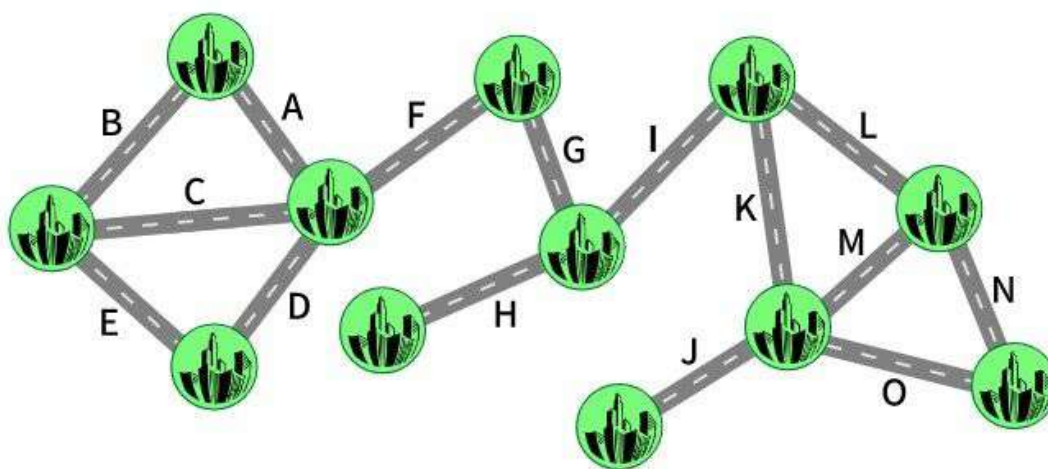


Epidēmiskā krīze

Taivāna

Uzdevuma apraksts

Bebruzemē ir 12 pilsētas, kuras savieno lielceļi, kas apzīmēti ar burtiem no A līdz O, tā, kā parādīts zīmējumā:



Pilsētas, kuras ceļi savieno tieši vai netieši (no vienas pilsētas ir iespējams aizbraukt uz otru, braucot cauri kādai citai), veido "ekonomisku savienību". Tāpēc šobrīd visas 12 pilsētas veido vienu savienību. Epidēmijas saasināšanās laikā, lai samazinātu ceļošanas iespējas, pilsētu mēri ir vienojušies slēgt divus ceļus. Šīs darbības mērķis ir izveidot trīs ekonomiskās savienības. Lai samazinātu lēmuma ietekmi uz pilsētu ekonomiku, mazākajām no šīm ekonomiskajām savienībām jābūt ar pēc iespējas lielāku pilsētu skaitu.

Jautājums

Kuri divi ceļi jāslēdz?

Atbildes forma

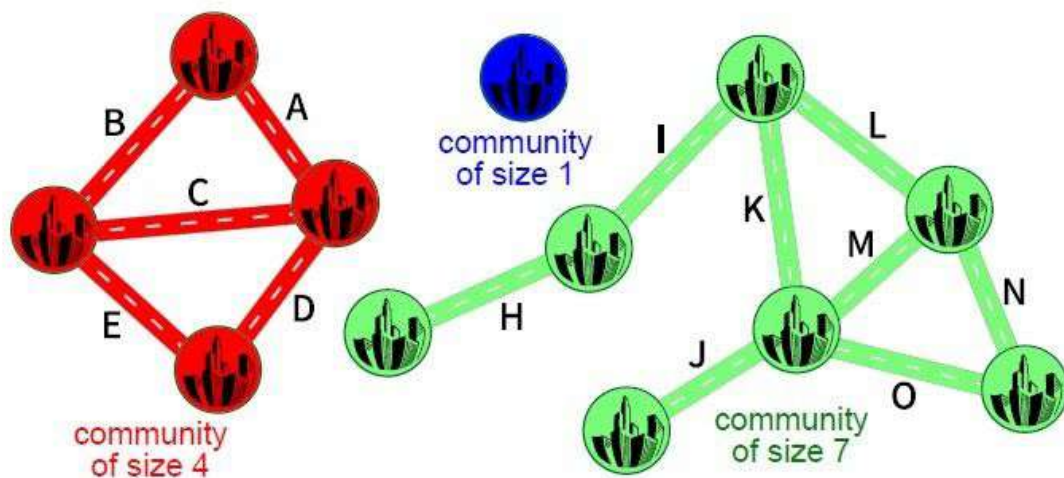
Atbildē jāsniedz divi burti alfabētiskā secībā bez atdalošām tukšumzīmēm.

Pareizā atbilde: FI.

Atbildes izskaidrojums

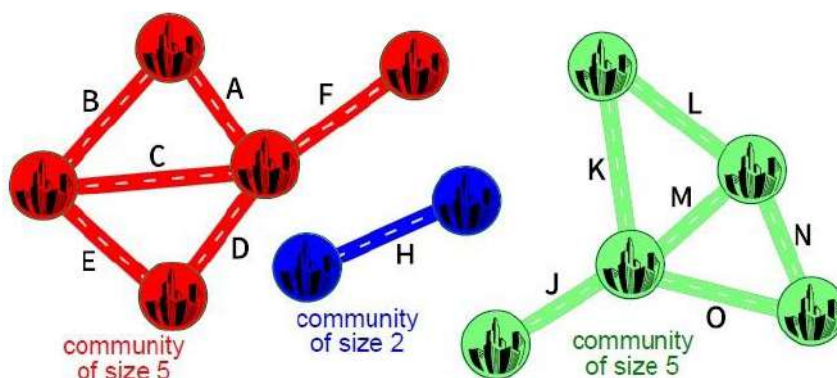
Ir vairāki ceļi, kurus var ignorēt tāpēc, ka to slēgšana nemainīs ekonomisko savienību. Piemēram, slēdzot ceļu C, ekonomiskā savienība netiks sadalīta, jo šī ceļa vietā varēs izmantot ceļus A un B. Ieskatoties vērīgāk, ir vērts slēgt tikai piecus ceļus: F, G, H, I vai J, kuru slēgšana sadala ekonomisko savienību mazākās savienībās. Jāatzīst, ka var slēgt arī ceļus A un B, vienā ekonomiskā savienībā atstājot vienu pilsētu. Bet šajā gadījumā sākotnējā ekonomiskā savienība būtu sadalīta tikai divās, bet ne prasītajās trīs daļās.

Tagad izmēģināsim dažādu ceļu pāru (no minētajiem pieciem) slēgšanu un katrā no variantiem aplūkosim pilsētu skaitu mazākajā ekonomiskajā savienībā. Piemēram, ja slēgs ceļus F un G, pilsētu skaits ekonomiskajās savienībās būs attiecīgi 4, 1 un 7. Mazākajā ekonomiskajā savienībā ir viena pilsēta.

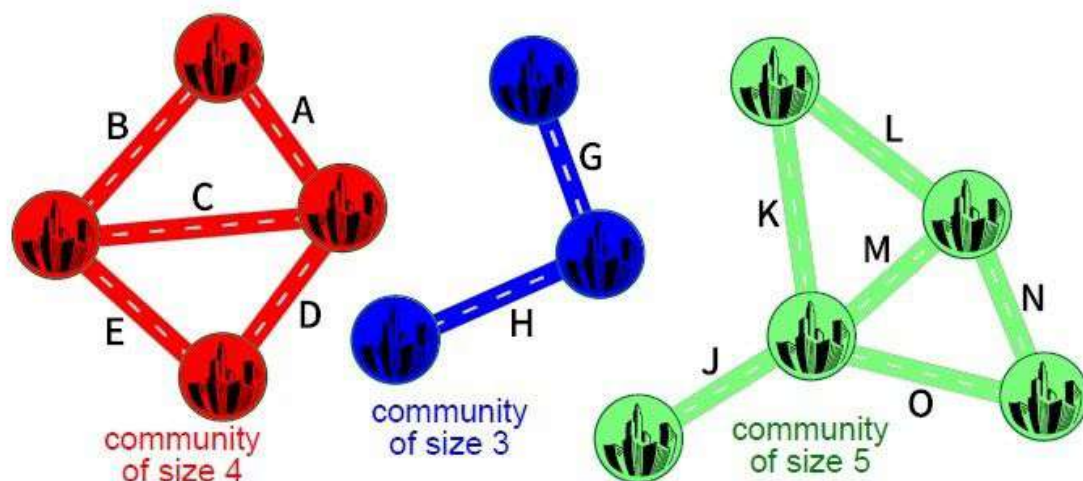


Mazākajā ekonomiskajā savienībā viena pilsēta ir arī tad, ja izvēlas slēgt ceļu pārus FH, FJ, GH, GJ, HI, HJ vai IJ.

Pēc ceļu G un I slēgšanas sākotnējā ekonomiskā savienība sadalās ekonomiskās savienībās, kuru izmēri ir attiecīgi 5, 2 (mazākā savienība) un 5 pilsētas:



Atlikušais ceļu pāris ir F un I. Šajā gadījumā izveidojušos ekonomisko savienību lielums ir 3, 4 un 5, kur mazākajā ir trīs pilsētas:



Tā kā tika meklēts variants, kurā mazākajā ekonomiskajā savienībā būtu pēc iespējas vairāk pilsētu, un visi varianti ir aplūkoti, tad lielākais pilsētu skaits mazākajā ekonomiskajā savienībā ir trīs pilsētas un variants ar ceļu F un I slēgšanu ir meklētais.

Tā ir informātika!

Šis uzdevums ir par saistītu komponentu atrašanu neorientētā grafā, kur jebkuru virsotņu pāri savieno vismaz viens ceļš.

Lai atrastu saistītu komponenti, jūs varat izmantot grafa apstaigāšanu sākot no avota virsotnes un sistemātiski pāriet uz nākamajām virsotnēm, kas savienotas ar avota virsotni, līdz kamēr vairs nevar sasniegt vēl neapstaigātas virsotnes.

Šajā uzdevumā mēs arī vēlamies atrast tiltiņus. Tiltiņš ir šķautne, kuras izmešana no grafa palielina saistīto komponentu skaitu tajā.

Saistīto komponentu analīzi izmanto vairākās datorredzes lietojumprogrammās, lai identificētu noteiktus reģionus. To izmanto, piemēram, rakstzīmju segmentēšanai automatiskās rakstzīmju atpazīšanas sistēmās, priekšplāna un fona atdalīšanai vizuālajā izsekošanā un medicīnisku attēlu apstrādē.

Bebr[a]s' 20



Copyright Bebras