

Bebr[a]s

'25

2025. gada 1. kārtas uzdevumi ar atbildēm

5.-6. klase

INFORMATĪVIE ATBALSTĪTĀJI



Izglītības un zinātnes
ministrija

start(it)

www.startit.lv

Satura rādītājs

Salas izpētes ekspedīcijas

Brīnišķīgā mašīna

Kreisais

Smilšu zīmējumi

Seulas apskate

Piepūšamie riņķi

Telpiskā domāšana

Emmas dārzs

Izvairoties no mākoņiem

Pārvadāšanas stratēģija

Autobuss

Trešais

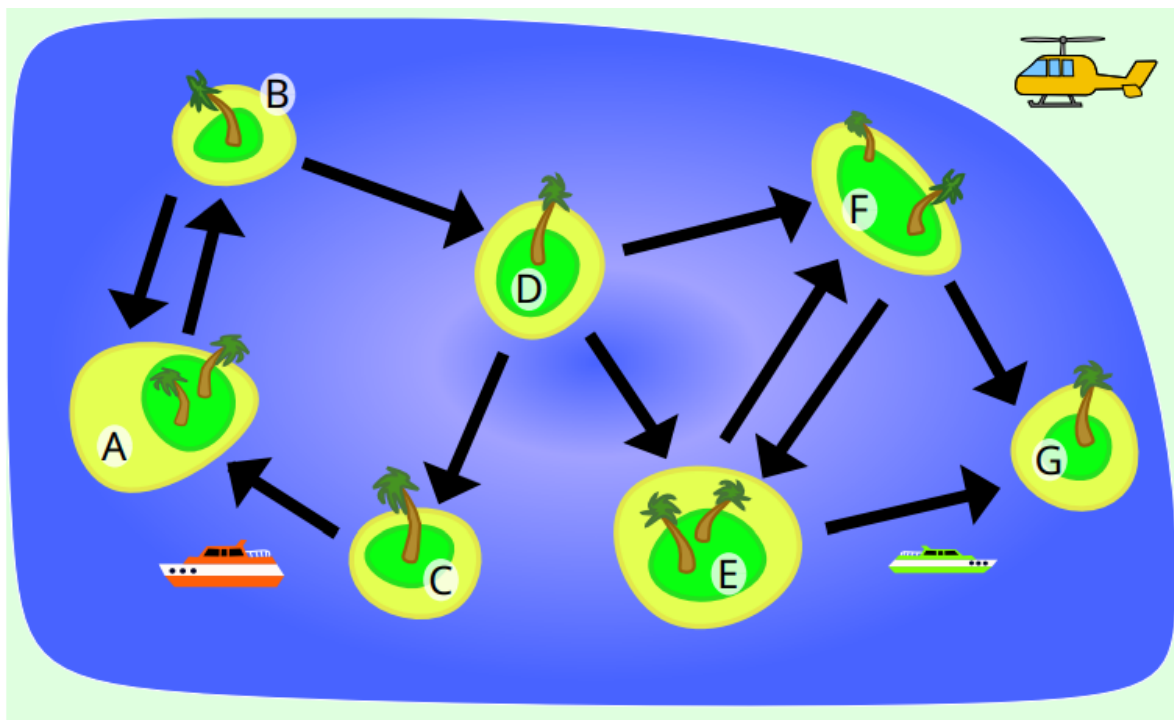
Stāvvietā

Aizraujošais nobrauciens

Salas izpētes ekspedīcijas

Austrija

Pētniecības komandai ir jāizpēta visas salas. Viņi var izmantot helikopteru, lai nolaistos uz jebkuras salas. Starp salām kursē arī prāmji, bet tikai kartē norādītajos virzienos. Kad komanda dodas izpētes braucienā, viņi nolaistas ar savu helikopteru uz kādas salas, var izmantot jebkuru prāmju skaitu un pēc tam atgriezties helikopterā.



Jautājums / Izaicinājums

Kāds ir minimālais izpētes braucienu skaits, kas nepieciešams, lai katra no salām būtu apmeklēta?

Atbilžu varianti / Interaktivitātes apraksts

Veseli skaitļi no [1, 7];

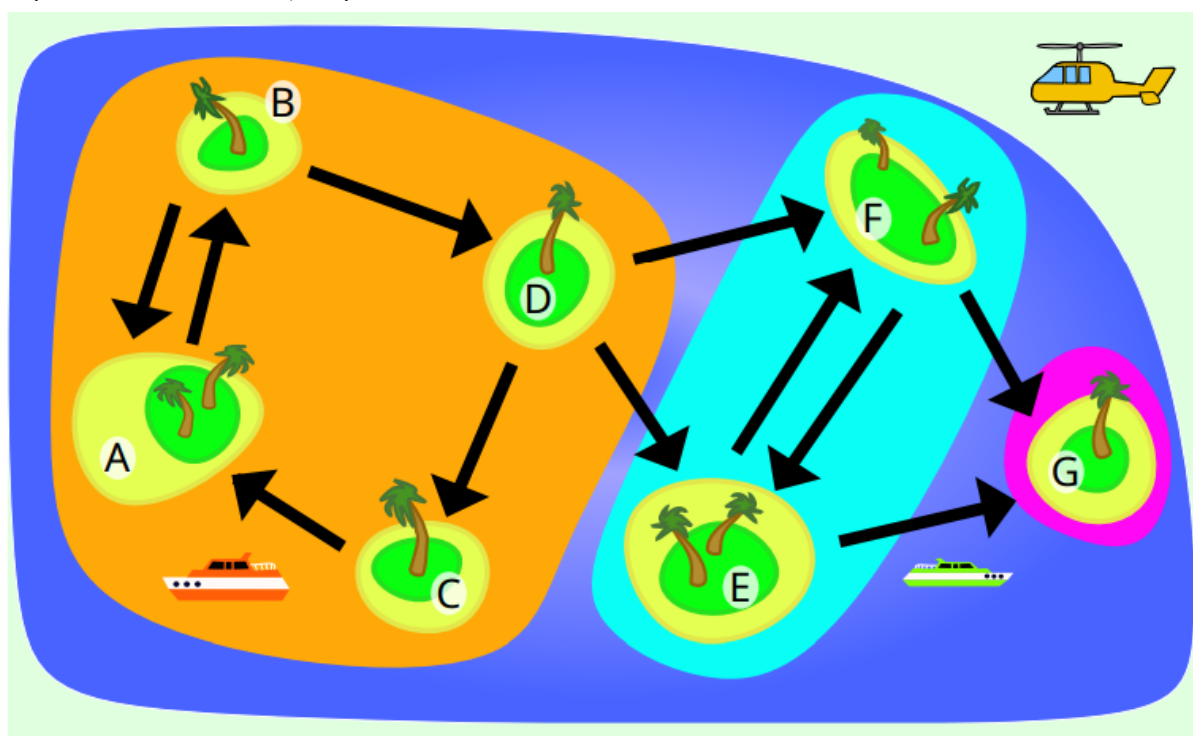
Atbildes skaidrojums

Minimālais pētniecības braucienu skaits, kas nepieciešams, ir 3.

Tā kā grupai pēc katra pētniecības brauciena ir jāatgriežas savā helikopterā, viņi var apmeklēt tikai tās salas, pēc kuru apmeklēšanas viņi var atgriezties sākuma salā. Lai to atrisinātu, mēs sadalām salas grupās, kur:

- No jebkuras salas grupā ir iespējams ar prāmi aizceļot uz katru citu salu grupā.
- Kad komanda pamet grupu, tā tajā vairs neatgriežas.

Lai atrastu katru šādu grupu, var sākt ar patvaļīgu salu, kas iepriekš nav piešķirta nevienai grupai, un pēc tam pievienot visas salas, no kurām var ceļot uz sākuma salu un atpakaļ. Attēlā salas ir sadalītas trīs grupās, kas apzīmētas ar dažādām krāsām. Tā kā ceļošana starp šīm grupām ir iespējama tikai vienā virzienā, katrai grupai ir nepieciešams atsevišķs izpētes brauciens.



Bieži pieļauta kļūda ir domāt, ka garš brauciens, piemēram, $D \rightarrow C \rightarrow A \rightarrow B \rightarrow D \rightarrow F \rightarrow E \rightarrow G$, darbosies. Lai gan tajā apmeklē katru salu, komanda nevarētu atgriezties savā helikopterā, padarot to par nederīgu risinājumu. Tādējādi komandai ir nepieciešami trīs atsevišķi izpētes braucieni - pa vienam katrai salu grupai.

Šī ir informātika

Iekrāsotās salu grupas atbildes skaidrojumā apzīmē salu kopas, kur katru grupas salu var sasniegt no jebkuras citas salas. Datorzinātnēs šo koncepciju sauc par cieši saistītu komponentu atrašanu. Cieši saistītas komponentes var izmantot daudzos uzdevumos. Piemēram:

- Tīmekļa navigācija: Attēlo tīmekļa vietņu grupas, kas visas ir saistītas viena ar otru.
- Sociālie mediji: Parāda lietotāju kopas, kas seko viens otram tīklā.
- Transporta tīkli: Parādīt reģionus, kuros ir iespējams ceļot starp pilsētām abos virzienos. Izpratne par cieši saistītām komponentēm palīdz daudzās jomās, tostarp tīkla analīzē, ieteikumu sistēmās un optimizācijas problēmu risināšanā.

Šī ir skaitļošanas domāšana.

Lai atrisinātu šo uzdevumu, skolēniem ir jāatpazīst, ka šis uzdevums ir par salu grupu atrašanu, lai varētu pārvietoties starp salām (abstrakcija). Turklāt viņiem ir jāizdomā sistemātisks veids, kā atrast šādas salu grupas (algoritmiskā domāšana).

Brīnišķīgā mašīna

Vācija

Bebriem ir lieliska mašīna.

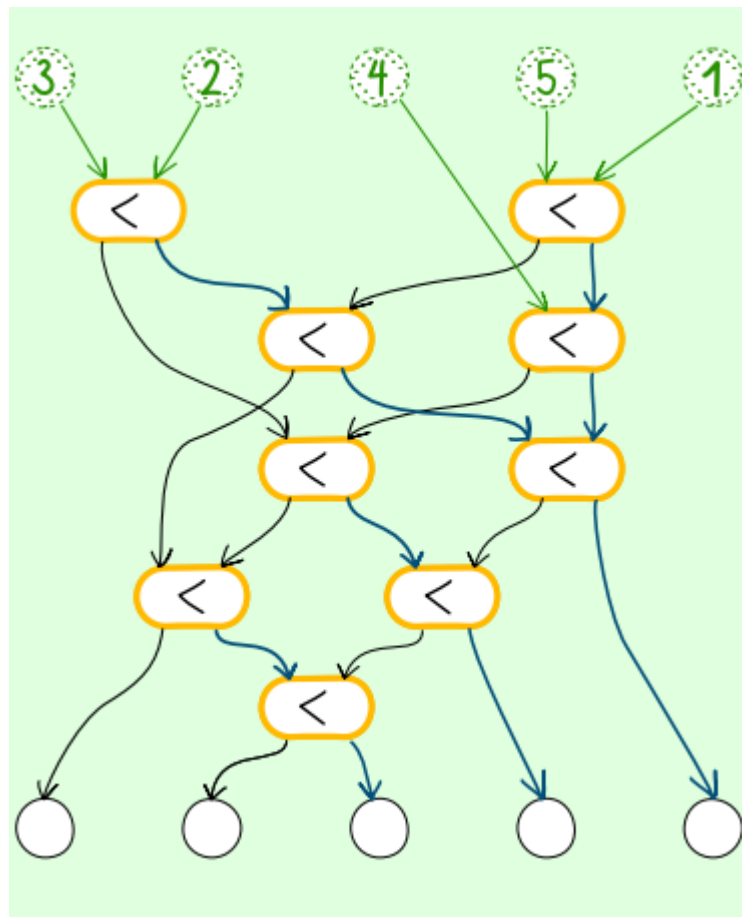
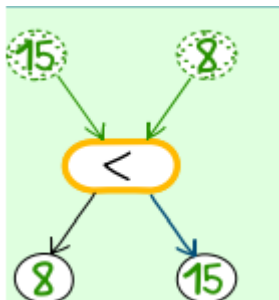
Sākumā pieci skaitļi ir ierakstīti ievades laukumos mašīnas augšpusē. Piemēram, 3, 2, 4, 5 un 1.

Pa bultiņām cauri slēdžiem skaitļi “virzās” caur mašīnu uz leju līdz izejas laukumiem apakšā.

Katrs no deviņiem slēdžiem salīdzina divus ienākošos skaitļus un “novirza”...

- ... mazāko skaitli pa kreisi un
- ... lielāko skaitli pa labi.

Piemēram:



Jautājums / Izaicinājums

Kādu uzdevumu veic lieliskā mašīna?

Atbilžu varianti / Interaktivitātes apraksts

A Izvada skaitļus apgrieztā secībā. Rezultāts: 1, 5, 4, 2, 3

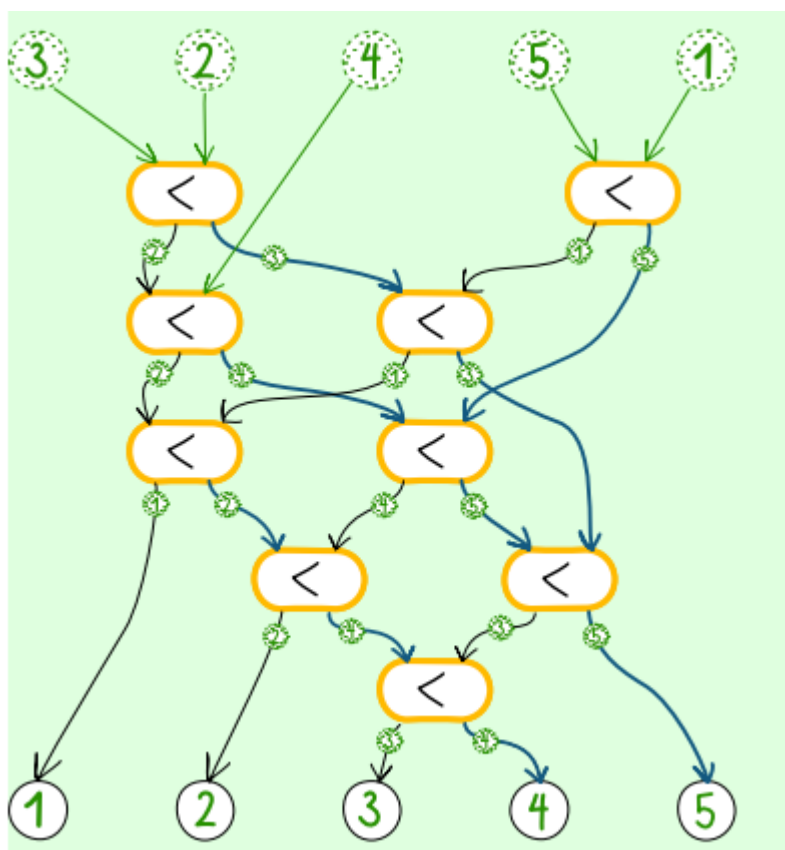
B Sakārto skaitļus augošā secībā. Rezultāts: 1, 2, 3, 4, 5

C Sakārto skaitļus dilstošā secībā. Rezultāts: 5, 4, 3, 2, 1

D Izvada skaitļus tādā pašā secībā. Rezultāts: 3, 2, 4, 5, 1

Atbildes skaidrojums

Pareizā atbilde: B



Nepareizās atbildes A, C, D ir ticami risinājumi, ko varētu izvēlēties dalībnieks, kurš neizmēģina mašīnu.

Šī ir informātika

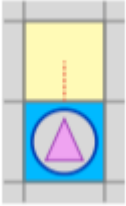
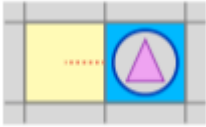
Informātikā šī Bebru lieliskā mašīna ir pazīstama kā "kārtotāšanas tīkls". Tā ir veidota no vairākām identiskām, ļoti vienkāršām sastāvdaļām – salīdzinātājiem. Katrs salīdzinātājs

saņem divas skaitliskās vērtības un salīdzina tās un pārsūta mazāko vērtību pa kreisi, bet lielāko - pa labi. Savienojot pietiekamu skaitu salīdzinātāju, jebkuru skaitļu secību var sakārtot pēc to vērtības. Lai kārtotu piecus skaitļus, nepieciešami vismaz deviņi salīdzinātāji. Sešiem skaitļiem nepieciešami divpadsmit salīdzinātāji. Kārtošanas tīkliem ir praktiska nozīme šādu iemeslu dēļ: (1) Kopumā ātra un efektīva kārtošana atvieglo meklēšanu, piemēram, meklētājprogrammu rezultātu sarakstos. (2) To kārtošanas algoritms ir piemērots paralēlai izpildei, kas var būt ļoti ātra. (3) Tie ir ļoti piemēroti aparatūras ieviešanai, piemēram, grafikas procesoros, jo salīdzinātājus var veidot kā ļoti vienkāršas un tāpēc lētas elektroniskas ierīces. Trūkumi ir: (1) Šķirošanas tīkli ir paredzēti tikai fiksēta garuma ievadei. (2) Salīdzinātāju skaits strauji pieaug līdz ar ievades garumu.

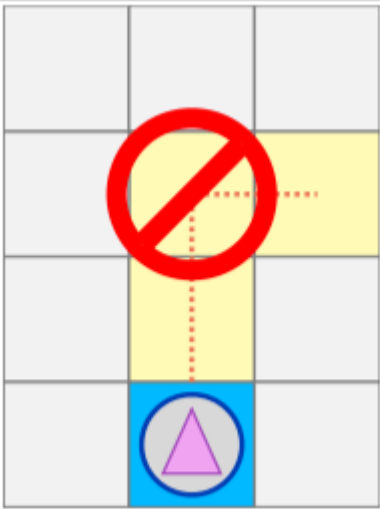
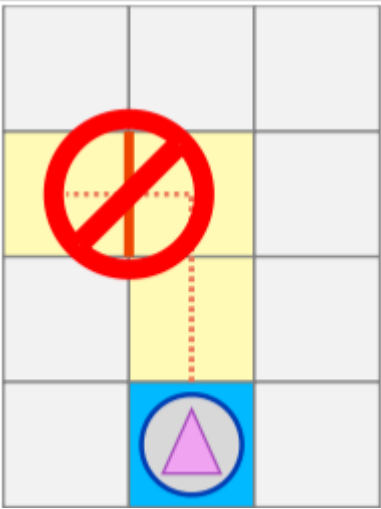
Kreisais

Vācija

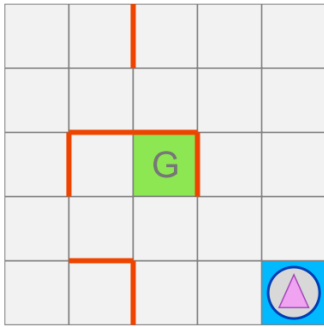
Robots Kreisais pārvietojas rūtiņu laukumā pa iepriekšatzīmētām rūtiņām, kuru starpā nav šķēršļu. Tas var veikt tikai šādus gājienus:

Paiet uz priekšu	Pagriezties pa kreisi un paiet uz priekšu
	

Daži piemēri, kā Kreisais iet nevar:

Robots nevar pagriezties pa labi	Robots nevar iet pāri šķērslim (sienai starp rūtiņām)
	

Ir dots rūtiņu laukums ar dažām sienām.

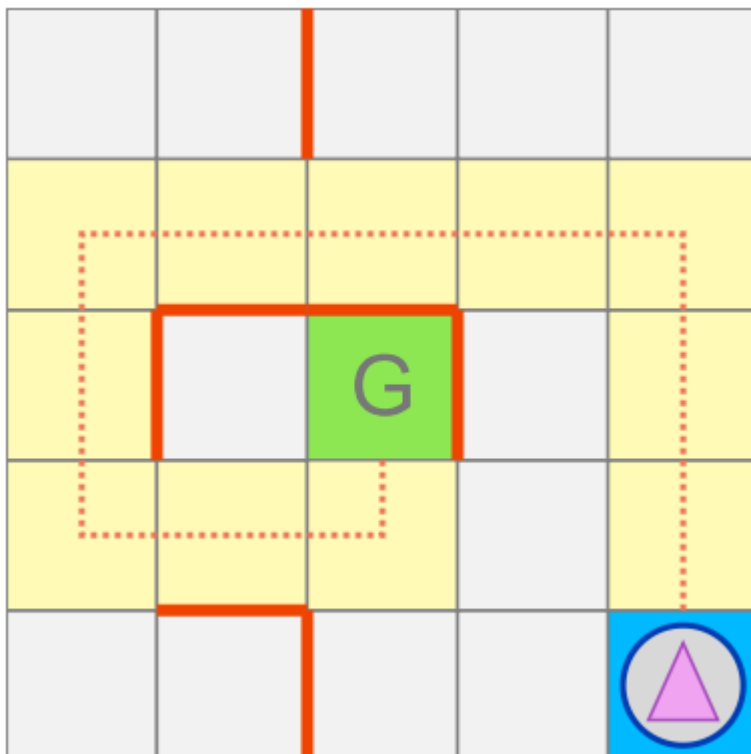


Atzīmējiet laukuma rūtiņas pa kurām Kreisais var sasniegt mērķi - zaļo rūtiņu "G". Ceļā uz mērķi Kreisajam ir jāapmeklē pēc iespējas mazāk laukuma rūtiņu. Atlasiet visas laukuma rūtiņas, kuras Kreisais apmeklēs ceļā uz mērķi.

Interaktivitātes apraksts

Uz katras laukuma rūtiņas (izņemot sākuma un mērķa rūtiņas) var noklikšķināt, lai pārslēgtu to stāvokli starp atlasītu un neatlasītu. Atlasītā rūtiņa tiek iezīmēta ar citu krāsu. Sākotnēji visas noklikšķināmās rūtiņas nav atlasītas.

Atbildes skaidrojums



Zīmējumā parādīts (a) kā Kreisais sasniedz mērķi, virzoties tikai pa kreisi, un (b) rūtiņas, kas jāatlasa pareizajā atbildē. Sienu novietojums neļauj citādi sasniegt mērķi ar mazāku vai vienādu apmeklēto laukuma rūtiņu skaitu.

Tā ir informātika

Nabaga Kreisais! Šķiet, ka tā darbība ir stipri ierobežota. Ja vien viņš varētu veikt arī citus gājienus! Ja viņš varētu pagriezties pa labi un varbūt pat kāpt pāri sienām, viņa dzīve laukumā būtu daudz vieglāka. Kreisais justos daudz pārliecinātāks, ja tam būtu sarežģītāks instrukciju kopums. (Robotam tā pamatkustības kontrolē atbilstošas instrukcijas robota programmatūrā.) Bet vai tas tiešām ir nepieciešams? Piemēram, Kreisais varētu pagriezties pa labi, veicot trīs pagriezienus pa kreisi pēc kārtas. Mums tikai jāatbrīvojas no noteikuma, ka Kreisajam pēc viena pagrieziņa ir jāpārvietojas uz priekšu. Tad tas varētu pārvietoties visos virzienos. Un tā vietā, lai kāptu pāri sienai, viņš varētu vienkārši apiet to (ja iespējams). Kreisais varētu dzīvot laimīgi līdz mūža galam ar savu samazināto instrukciju kopu – ja programmētāji izmantotu to gudri. Informātikā tieši šie divi procesoru instrukciju kopu projektēšanas veidi ir vispopulārākie: Daži procesori ir CISC (kompleksās instrukciju kopas datori), citi ir RISC (samazinātas instrukciju kopas datori) – tāpat kā Kreisais šajā uzdevumā. CISC parasti ir daudz dažādu instrukciju, kas var būt ļoti spēcīgas (piemēram, kāpšana pāri sienai), savukārt RISC ir tikai tās instrukcijas, kas tiek patiešām un bieži izmantotas. Abiem arhitektūras veidiem ir savas priekšrocības un trūkumi. Pazīstamu zīmolu procesori ir vai nu CISC, vai RISC, un RISC procesori pēdējā laikā ir nedaudz populārāki.

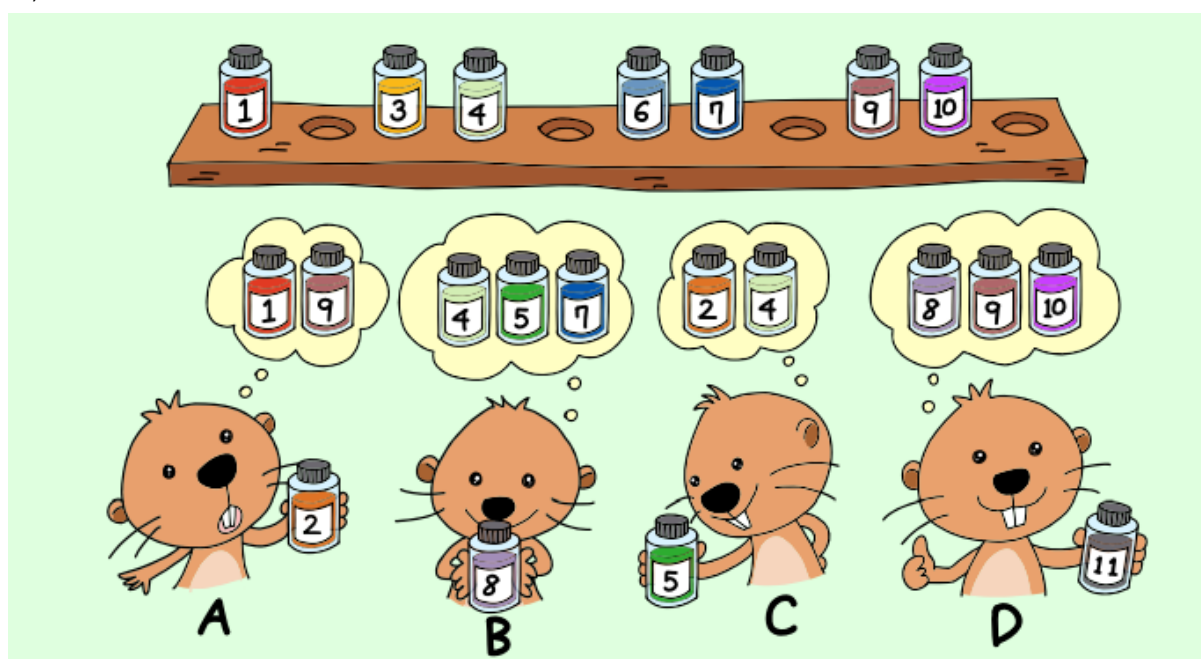
Šī ir skaitļošanas domāšana

Uzdevums ietver algoritmisko domāšanu, jo, pamatojoties uz divām pamata komandām, ir jāizveido īsākais maršruts.

Smilšu zīmējumi

Dienvidkoreja

Gleznošana, izmantojot smiltis, ir aktivitāte, kurā uz papīra ber krāsainas smiltis un pievieno līmi, lai tās nostiprinātu. Tā kā līme ātri žūst, skolēniem pirms darba uzsākšanas jāizvēlas visas krāsas. Katrā burciņā, kas numurēta no 1 līdz 11, ir dažādas krāsas smiltis. Skolotājs katram skolēnam iedod burciņu. Attēlā ir parādītas numurētās burciņas, kuras pašlaik tur Andis, Bora, Centis un Dona, un tās burciņas, kuras katrs no viņiem vēlas izmantot.



Katram skolēnam, pirms gleznošanas uzsākšanas, jāiegūst visas vēlamās krāsas. Pēc gleznas pabeigšanas visiem jāatdod visas savas izmantotās burciņas. Ja skolēns vēlas izmantot burciņu, kas ir citam skolēnam, viņam jāgaida, līdz burciņa tiek atdota un kļūst pieejama.

Jautājums / Izaicinājums

Kurš ir pēc kārtas pēdējais skolēns, kurš sāks gleznot?

Atbilžu varianti

- A) Andis
- B) Bora
- C) Centis
- D) Dona

Atbildes skaidrojums

Pareizā atbilde ir D) Dona.

Andis vēlas 1. un 9. burciņu, bet nevienam citam nav 1. un 9. burciņa, tāpēc viņš var nekavējoties sākt gleznot. Centis vēlas 2. burciņu, kas jau ir Andim, tāpēc Centim jāgaida, kamēr Andis pabeigs gleznot. Bora vēlas 5. burciņu, bet tā ir Centim, tāpēc Borai jāgaida, kamēr Centis pabeigs gleznot. Dona vēlas 8. burciņu, kas ir Borai, tāpēc Donai jāgaida, kamēr Bora pabeigs gleznot. Andis nekavējoties sāk gleznot. Kad viņš ir pabeidzis, gleznot sāk Centis. Kad Centis ir pabeidzis, gleznot var sākt Bora. Kad Bora ir pabeigusi, gleznot var sākt Dona. Tātad secība ir Andis, Centis, Bora un Dona. Tāpēc pēc kārtas pēdējais skolēns, kurš sāks gleznot, būs Dona.

Šī ir informātika

Šis uzdevums ir strupceļa situācijas piemērs. Strupceļš rodas, ja vairāki procesi konkurē par vieniem un tiem pašiem ierobežotajiem resursiem, piemēram, failiem, datubāzēm vai datora atmiņu. Strupceļā iesaistītie procesi nevar virzīties uz priekšu, ja tie ir bloķēti, gaidot nepieejamo resursu atbrīvošanu. Strupceļi var izraisīt sistēmas nestabilitāti, samazinātu veiktspēju un pat pilnīgu sistēmas kļūmi. Šajā uzdevumā strupceļš rodas, ja vienam vai vairākiem skolēniem ir nepieciešama burka, ko tur cits, izraisot gaidīšanas stāvokli.

Šī ir skaitļošanas domāšana

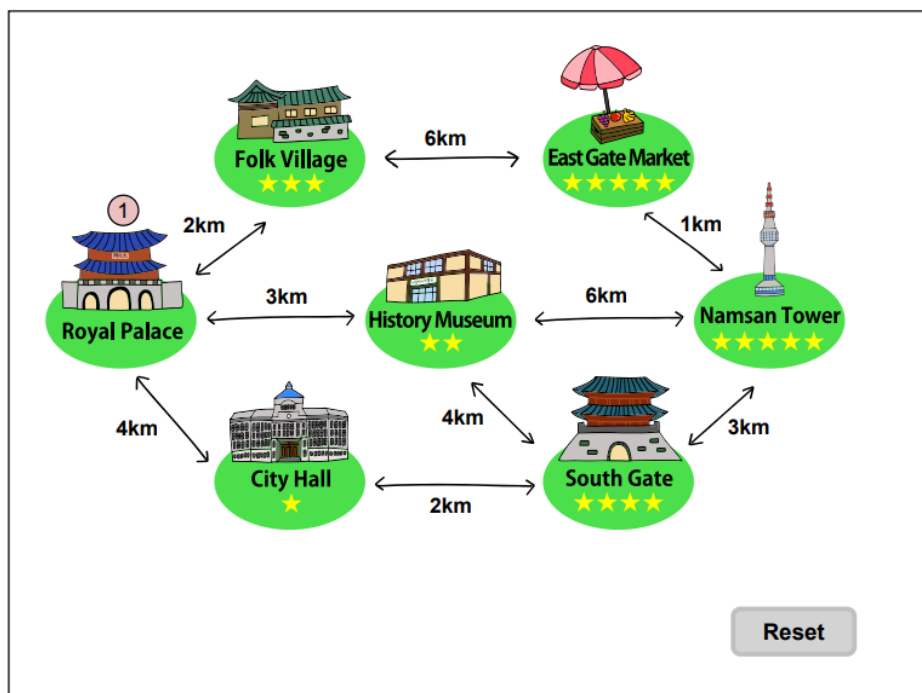
Algoritmi: Šajā uzdevumā skolēni izmanto algoritmisko domāšanu, lai noteiktu secību, kādā skolēni var sākt gleznot.

Sadalījums: Šī problēma ir pārāk sarežģīta, lai to atrisinātu kopumā. Tā vietā tā ir jāsadala mazākās daļās, kuras var atrisināt pa vienai. Piemēram, skolēni var pēc kārtas analizēt katram bebram nepieciešamās krāsas: kuras krāsas bebram ir vajadzīgas un kuram katra krāsa ir tagad.

Seulas apskate

Dienvidkoreja

Dienvidkorejas galvaspilsēta Seula nodrošina ekskursiju autobusus, kas ļauj tūristiem apmeklēt Seulas galvenos tūrisma objektus. Zemāk redzamajā attēlā ir parādītas galvenās apskates vietas, attālums starp tām un zvaigznes norāda to popularitāti tūristu vidū.



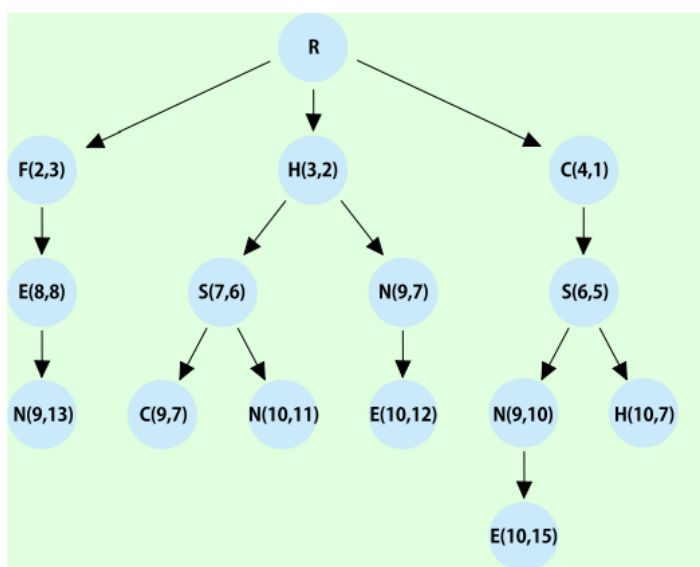
Jautājums / Izaicinājums

Pēc Karaliskās pils (Royal Palace) apmeklējuma Sjūzija vēlas braukt ar ekskursiju autobusu, taču viņai ir transporta biļete, kas ļauj nobraukt ne vairāk kā 10 kilometrus. Izveidojiet tādu apskates objektu apmeklēšanas maršrutu, lai viņa varētu nopelnīt pēc iespējas vairāk zvaigžņu. Katru apskates objektu drīkst apmeklēt tikai vienreiz.

Atbildes skaidrojums

Atbilde ir Karaliskā pils → Rātsnams → Dienvidu vārti → Namsanas tornis → Austrumu vārtu tirgus (Royal Palace → City Hall → South Gate → Namsan Tower → East Gate Market). Viens veids, kā atrisināt problēmu, ir izmantot diagrammu, kas sastāv no mezgliem, kur katrs apzīmē kādu tūrisma objektu. Diagrammas augšpusē ir sākuma mezgls, kas apzīmēts ar burtu: R (apzīmē Karalisko pili). Sākot no R, var uzzīmēt katru iespējamo ceļu, kas kopā nepārsniedz 10 km. Katrs mezgls diagrammā attēlo apskates objektu un ir apzīmēts ar apskates objekta sākotnējo burtu angļu valodā, kopējo

attālumu no sākuma, kopējo līdz šim savākto zvaigžņu skaitu. Piemēram, viens no iespējamajiem ceļiem ir apmeklēt Tautas ciematu, kas atrodas 2 km attālumā no sākuma punkta un kam ir 3 zvaigžņu vērtējums. Šis mezgls ir apzīmēts ar F (2, 3). No turienes, ja dodaties uz Austrumu vārtu tirgu (E), tas palielina kopējo attālumu līdz 8 km un kopējo zvaigžņu vērtējumu līdz 8, mezgls tiek apzīmēts ar E (8, 8). Nākamā izvēle ir Namsanas tornis (N), kas palielina kopējo attālumu līdz 9 km un kopējo zvaigžņu vērtējumu līdz 13, tāpēc tas ir apzīmēts ar N (9, 13).



Lai atrastu labāko ceļu, pārbaudiet mezglus visu derīgo ceļu galos un izvēlieties to, kuram ir visaugstākā zvaigžņu vērtība, nepārsniedzot 10 km robežu. Diagrammā mezgls ar visaugstāko zvaigžņu vērtību ir E(10, 15). Tas atbilst ceļam:

Karaliskā pils → Rātsnams → Dienvidu vārti → Namsanas tornis → Austrumu vārtu tirgus. Vai arī izmantojot iniciāļus: R → C (4, 1) → S (6, 5) → N (9, 10) → E (10, 15).

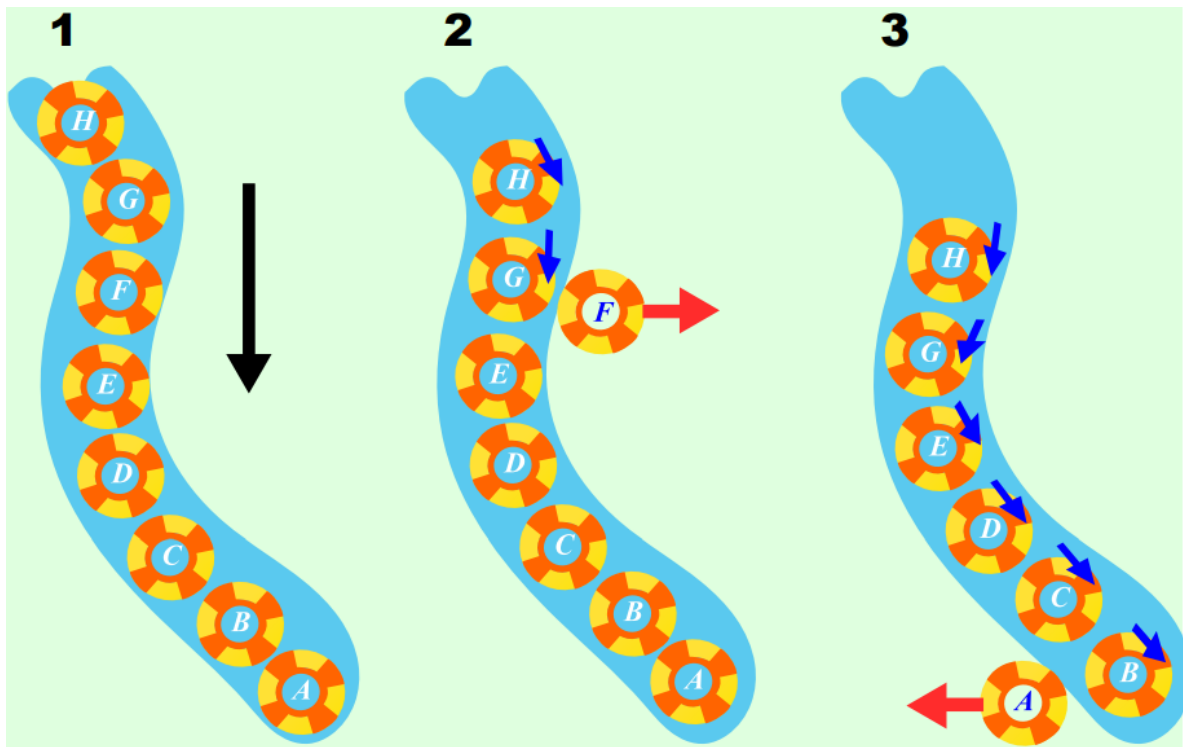
Šī ir informātika.

Optimizācijas problēmas ietver lēmumu pieņemšanu, lai maksimāli palielinātu efektivitāti (piemēram, rezultātu vai peļņu) ierobežotu resursu (piemēram, laika vai naudas) ietvaros. Grafu modelēšana ļauj veikt tādu elementu kā apskates objektu un ceļu strukturālu analīzi, lai izstrādātu optimālus maršrutus. Šī pieeja tiek izmantota, lai risinātu reālas pasaules problēmas tādās jomās kā tūrisma maršrutu plānošana, loģistika un transporta sistēmas. Uzdevums iepazīstina ar galvenajiem principiem gan resursu pārvaldības, gan maršrutu optimizācijas izpratnē.

Piepūšamie riņķi

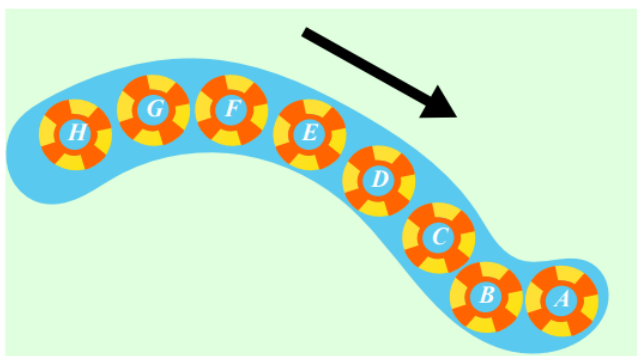
Kanāda

Piepūšamie riņķi peld lejup pa straumi (→) īsā šaurā ūdens kanālā, kura galā tie sablīvējas rindā, kā parādīts zīmējumā:



Kad kāds riņķis tiek izvilīts no ūdens (→), katrs riņķis aiz tā papēd vienu pozīciju uz priekšu, lai aizpildītu izveidojušos spraugu (zilā bultiņa). Mūsu piemērā kopējais pavirzīšanos par vienu pozīciju skaits ir 8 (divas pēc F izvilšanas un sešas pēc A izvilšanas).

Nosakiet, kāds ir kopējais pavirzīšanos par vienu pozīciju kopskaits, ja riņķi ir sablīvējušies šādā secībā:



un tiek izvilkti pa vienam riņķim pēc kārtas šādā secībā: B, G, E, D, H!

Atbildes opcijas / interaktivitātes apraksts

A) 10 b) 11 c) 12 d) 13

Atbildes paskaidrojums

Pareizā atbilde ir b) 11.

Sāukumā, sākot no kanāla gala, riņķu rinda ir A, B, C, D, E, F, G, H.

Izvilktais riņķis	Riņķi, kas turpina peldēt	Jaunā riņķu rinda kanāla galā
B	C, D, E, F, G un H (6 riņķi)	A, C, D, E, F, G, H
G	H (1 riņķis)	A, C, D, E, F, H
E	F un H (2 riņķi)	A, C, D, F, H
D	F un H (2 inner tubes)	A, C, F, H
H	- (neviens nepeld)	A, C, F

Pavisam $6+1+2+2=11$ pavirzīšanās.

Šī ir informātika

Ir tikai dabiski, ja dati vai informācija rodas kā virkne. Ikdienas dzīvē tās varētu būt lidmašīnas, kas gaida pacelšanos uz skrejceļa, dokumentā izdarītie labojumi (to

veikšanas secībā), vai piepūšamie riņķi upē, kā šī uzdevuma stāstā. Kad mēs izmantojam datorprogrammu, lai atrisinātu problēmas, kas saistītas ar šiem scenārijiem, mums parasti ir jāuzglabā šie dati. Viens veids, kā to izdarīt, ir saglabāt datus atmiņā secīgi. Tas ir, dažādus datus (piemēram, lidmašīnas, labojumus vai piepūšamos riņķus) saglabāt secīgās atmiņas vietās. Viena no priekšrocībām, to darot, ir tā, ka parasti ir viegli atrast elementu virknē, ņemot vērā tā atrašanās vietu atmiņā. Viens no trūkumiem ir tāds, ka, izņemot elementu no virknes, mums ir jāpārvieto elementi, lai tie saglabātos secīgās atmiņas vietās. Tas ir līdzīgi riņķiem šajā uzdevumā, kas peld pa straumi, lai aizpildītu spraugu, ko atstājis izvilktais riņķis. Tas prasa laiku, un, ja tas ir pietiekami bieži, var ievērojami palēnināt programmu. Ņemiet vērā, ka pat tad, ja "reālās pasaules" dati nav sakārtoti virknē (piemēram, cilvēku populācija), mēs joprojām varētu izvēlēties tos saglabāt atmiņā. Pieņemot lēmumu kā to izdarīt, mums jāsabalansē priekšrocības un trūkumi. Lai to izdarītu, ir nepieciešama izpratne par to, kā dati tiek glabāti atmiņā un kā datorprogrammas darbojas/apstrādā datus.

Šī ir skaitļošanas domāšana

Šīs problēmas risināšanai ir jāizpilda soļu virkne, kas saistīta ar riņķu izvilkšanu un spraugu starp tiem aizpildīšanu. Šis ir klasisks algoritmiskās domāšanas gadījums, kurā, kamēr tiek iegūts rezultāts, ir jāizpilda iteratīvs process (kur soļu izpildes secība ir svarīga). Vispārinājums ir nepieciešams, lai atrisinātu uzdevumu atšķirīgai riņķu izvilkšanas secībai.

Telpiskā domāšana

Čehija

Bebrs Ksavjērs programmē savu pirmo tiešsaistes spēli un mācās pārveidot attēlu. Ar katru attēlu drīkst veikt tikai divas darbības: M (atspoguļošana) vai R (rotācija), kā parādīts zīmējumā:

Ievaddati	Darbība	Izvaddati
	M (atspoguļot pret spoguļi, kas atrodas uz vertikālas ass)	
	R (pagriezt par 90 grādiem pulksteņrādītāja virzienā)	

Sākotnējais attēls ir šāds:



Pēc kārtas attēlam no kreisās uz labo pusi tiek pielietota šāda darbību virkne:

R R R R M R M

Jautājums / Izaicinājums

Kā izskatīsies attēls pēc šīs darbību virknes veikšanas?

Atbilžu varianti / Interaktivitātes apraksts

A)	B)	C)	D)
			

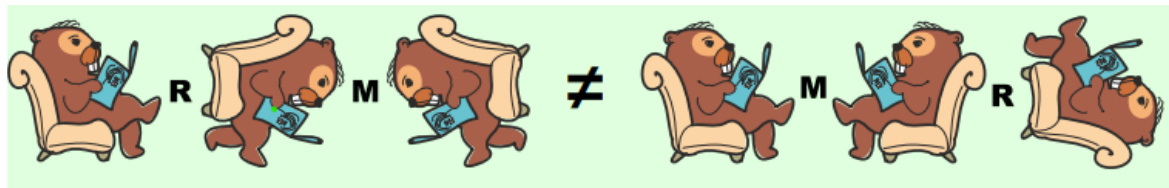
Atbildes skaidrojums

A) ir pareizā atbilde.

Šādi izskatās attēls pēc katras veiktās darbības:



Var pamanīt, ka četras rotācijas novieto attēlu tādā pašā orientācijā kā sākumā, un to pašu dara divas atspoguļošanas. No pirmā acu uzmetiena varētu pieņemt, ka piecas rotācijas un divas atspoguļošanas varētu būt tas pats, kas viena rotācija, un atbilde būtu B). Kāpēc tā nav taisnība? Jo ne tikai rotāciju un atspoguļošanas reižu skaits ietekmē rezultātu, bet arī darbību secību. Piemēram, R M rezultāts atšķiras no M R rezultāta:



Šī ir informātika

Viena no pirmajām informātikas kompetencēm ir spēja simulēt īsu programmu, kas sastāv no labi saprotamu pamatdarbību secības, izpildi. Arī spēja vizualizēt iegūto rezultātu pēc darbību secības ir svarīgs uzdevums jebkurā nozarē. Telpiskā domāšana ir īpaši svarīga, ja algoritmi ir saistīti ar robotu programmēšanu.

Šī ir skaitļošanas domāšana

Uzdevumu var atrisināt, veicot vienkāršu algoritmu. Tomēr to var atrisināt daudz ātrāk, ja izdodas atklāt, ka četrām secīgām rotācijām un divām secīgām atspoguļošanām nav seku (tās atgriež attēlu sākotnējā stāvoklī), kas ir noderīga prasme daudzās automatizācijas situācijās, kā arī simulācijās. Attēla rotēšana un atspoguļošana ir sarežģītas darbības, bet mēs tās pierakstām ar burtiem (simboliem), kas pati par sevi ir abstrakcija.

Emmas dārzs

Kipra

Emma vēlas izveidot maģisku dārzu un šodien ir stādīšanas diena. Viņas dārzā ir vieta četriem maģisku ziedu veidiem. Viņai ir deviņi dažādu veidu ziedi, no kuriem izvēlēties. Katram ziedu veidam nepieciešams noteikts dienu skaits no iestādīšanas brīža, pirms tas sāk ziedēt, un tad tas zied noteiktu dienu skaitu. Emma vēlas izvēlēties ziedus tā, lai būtu vismaz 15 dienu periods, kurā ziedētu vismaz viena veida ziedi. Viņai rūpīgi jāpārdomā, cik ilgs laiks nepieciešams katram ziedu veidam no iestādīšanas līdz uzziedēšanai un cik ilgi tas zied. Šie ir ziedu, no kuriem Emmai jāizvēlas, veidi:

Ziedu veida numurs	Nosaukums	Dienų skaits no iestādīšanas līdz uzziedēšanai	Ziedēšanas dienu skaits
1	Margrietiņas	8	4
2	Lavanda	3	4
3	Lilijas	10	2
4	Kliņģerītes	4	3
5	Orhidejas	13	3
6	Peonijas	13	4
7	Rozes	3	6
8	Saulespuķes	12	5
9	Tulpes	2	2

Jautājums / Izaicinājums

Šodien ir stādīšanas diena. Kurus četrus ziedu veidus viņai vajadzētu iestādīt, lai ziedi ziedētu katru dienu vismaz 15 dienas?

Atbilžu iespējas / Interaktivitātes apraksts

Četri naturāli skaitļi - ziedu veidu numuri.

Atbildes skaidrojums

Pareizā atbilde: 1 — margrietiņa, 7 — roze, 8 — saulespuķe, 9 — tulpe

Šo uzdevumu var atrisināt, izmantojot algoritmisku pieeju vai datu vizualizāciju.

Algoritmiskā pieeja:

Ziedi zied ļoti īsu laiku, tāpēc mēs sākam, izvēloties ziedu, kas uzzied pēc iespējas ātrāk:

Tulpe sāk ziedēt 2. dienā, ziedot divas dienas → nākamajam ziedam jāsāk ziedēt ne vēlāk kā 4. dienā.

Nākamais zieds varētu būt lavanda, kliņģerītes vai roze. No šīm rozei zied visilgāk: līdz 8. dienai. Tāpēc mēs izvēlamies rozi. → nākamajam ziedam jāsāk ziedēt ne vēlāk kā 9. dienā.

Vienīgā iespēja ir margrietiņa, kas sāk ziedēt 8. dienā, ziedot četras dienas līdz 11. dienai. → nākamajam ziedam jāsāk ziedēt ne vēlāk kā 12. dienā.

Vienīgā iespēja ir saulespuķe, kas zied piecas dienas līdz 16. dienai.

Izvēloties šo četru ziedu, ziedi zied katru dienu no 2. līdz 16. dienai, kas ir 15 dienas.

Datu vizualizācija:

Otrs veids, kā atrisināt šo uzdevumu, ir izveidot tabulu, kurā parādīts, kad katrs zieds zied (tabulā atzīmēts ar b).

	Dienu skaits kopš iestādīšanas																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1.									b	b	b	b					
2.				b	b	b	b										
3.											b	b					
4.					b	b	b										
5.														b	b	b	
6.														b	b	b	b
7.				b	b	b	b	b	b								
8.													b	b	b	b	b
9.			b	b													

Mēs redzam, ka 2. dienā zied tikai tulpe, tāpēc šis zieds ir jāizmanto. Tas pats attiecas uz 7. dienu (roze), 9. dienu (margrietiņa) un 12. dienu (saulespuķe). Iestādot šos četrus ziedu veidus, mēs varam redzēt, ka no 2. līdz 16. dienai zied vismaz viens zieds, tātad kopā 15 dienas katru dienu zied vismaz viens ziedu veids.

Šī ir informātika.

Šāda veida problēmas varētu atrisināt ar datoru, izmantojot intervālu pārklāšanas algoritmu, kas ir rījīgā algoritma veids. Rījīgie algoritmi iteratīvi atrod risinājumus, katrā solī izdarot labāko izvēli, neņemot vērā kopējo ietekmi uz galīgo risinājumu.

Šī ir skaitļošanas domāšana.

Datu attēlošana: datu attēlošana divdimensiju tabulā piedāvā efektīvu veidu, kā atrast pareizo atrisinājumu. Datu vizualizācija ir informācijas un datu grafiska attēlošana. Atbildes skaidrojumā ir parādīts datu vizualizācijas piemērs, kurā rindas apzīmē dažādus ziedu veidus, kolonnas — dažādas dienas un burti "b" apzīmē dienas, kurās zieds zied.

Emīlijas puķes

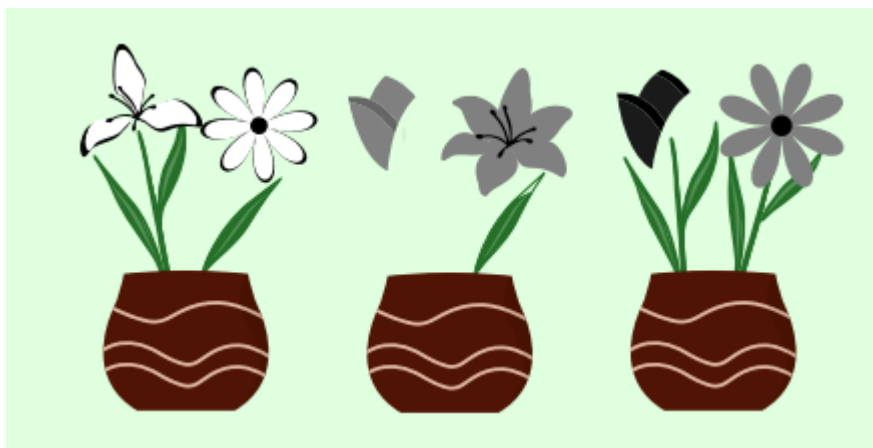
Brazīlija

Emīlija sakārto puķes trīs vāzēs. Viņai ir trīs veidu puķes: tulpes, lilijas un margrietiņas, kā parādīts attēlā:

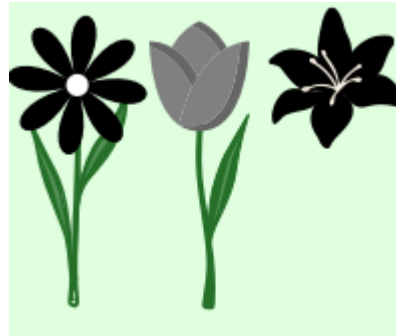


Katras puķes zieds (zīmējumā, ne dabā!) ir melns, balts vai pelēks, un puķei uz kāta ir 0, 1 vai 2 lapas.

Emīlija vienmēr savās vāzēs tur vienādu puķu skaitu. Viņa arī nodrošina, ka katrā vāzē puķes ir sakārtotas pēc viena no trim noteikumiem: vienā vāzē ir vai nu viena veida vai tās pašas krāsas puķes, vai arī puķes ar vienādu lapu skaitu. Visas vāzes ir jāsakārto, izmantojot vienu un to pašu no aprakstītajiem trim noteikumiem. Bet tagad viņai ir trīs vāzes ar sešām puķēm, kur katrai vāzei izpildītais noteikums atšķiras (krāsa pirmajā un otrajā vāzē, bet lapu skaits trešajā vāzē), kā parādīts attēlā zemāk:



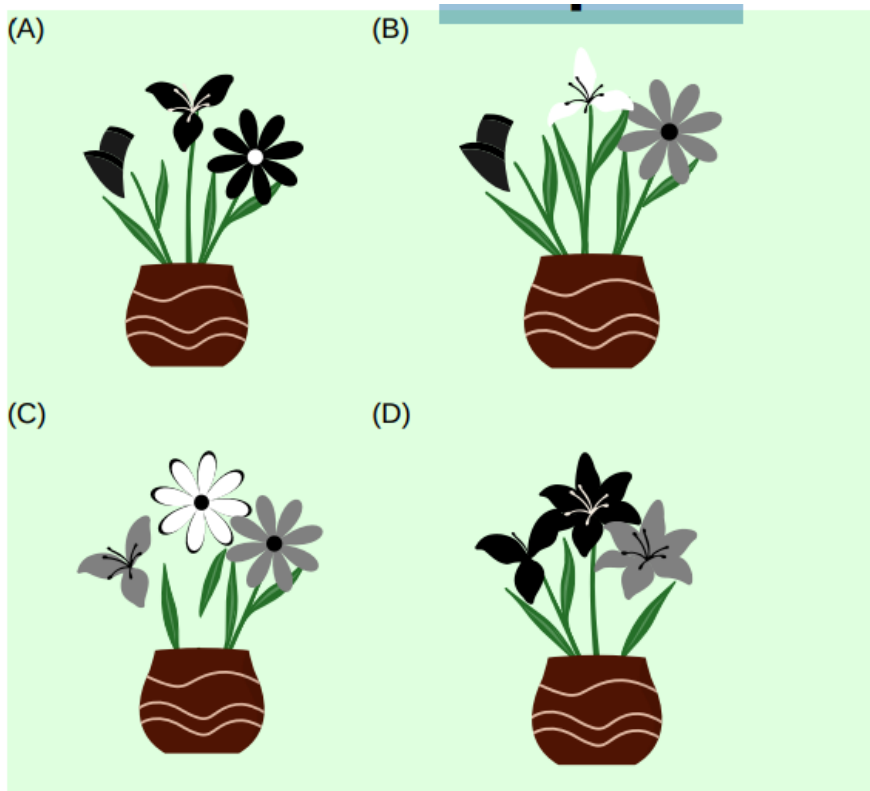
Viņa ņems vēl trīs puķes (parādītas labajā pusē) un reorganizēs visas deviņas puķes trīs vāzēs saskaņā ar vienu no iepriekšējiem noteikumiem.



Jautājums / Izaicinājums

Kura iespēja pareizi attēlo vienu no vāzēm beigās?

Atbilžu varianti / Interaktivitātes apraksts



Atbildes skaidrojums

Pareizā atbilde ir (D). Emīlijai ir šādas deviņas puķes:



Tātad katrā vāzē jābūt trim puķēm.

Ir divas baltas puķes, trīs melnas puķes un četras pelēkas puķes. Tā kā katras krāsas puķu skaits ir atšķirīgs, tās nevar sakārtot pēc krāsas.

Ir divas puķes ar 0 lapām, trīs puķes ar 1 lapu un četras puķes ar 4 lapām. Tātad puķes nevar sakārtot arī pēc lapu skaita.

Tādējādi, tā kā ir trīs tulpes, trīs lilijas un trīs margrietiņas, puķes jāsakārto pēc to veida. Vāžu galīgais izkārtojums izskatīsies šādi:



(A) variants nav pareizs, jo šajā vāzē puķes ir sakārtotas pēc krāsas.

(B) variants nav pareizs, jo šajā vāzē puķes ir sakārtotas pēc lapu skaita.

(C) variants nav pareizs, jo vāzē nav ievērots neviens no trim aprakstītajiem noteikumiem: tajā ir tikai divas viena veida puķes trīs vietā, tikai divas vienas krāsas puķes un tikai divas puķes ar vienādu lapu skaitu.

Šī ir informātika.

Šis uzdevums apvieno ierobežojumu ievērošanu un modeļu atpazīšanu. No vienas puses, modeļu atpazīšana ietver objektu klasificēšanu grupās, pamatojoties uz to kopīgajām īpašībām. No otras puses, noteikumi ir ierobežojumi, kas jāievēro. Šajā uzdevumā puķes ir objekti, un to veids, krāsa un lapu skaits ir to īpašības. Datoriem ir ļoti svarīgi iemācīties organizēt un saprast informāciju, tā tie iemācās atšķirt lietas vai atrast to, kas tām ir kopīgs. Datori aplūko īpašības, apkopo šo informāciju un izlemj, kā

grupēt objektus (tāpat kā mēs darām ar puķēm, kas jāsakārto katrā vāzē). Šis process ir pamatā mākslīgā intelekta metodei, ko sauc par uzraudzītu mašīnmācīšanos, kur dators iemācās kārtot lietas, izmantojot dotos piemērus.

Šī ir skaitļošanas domāšana

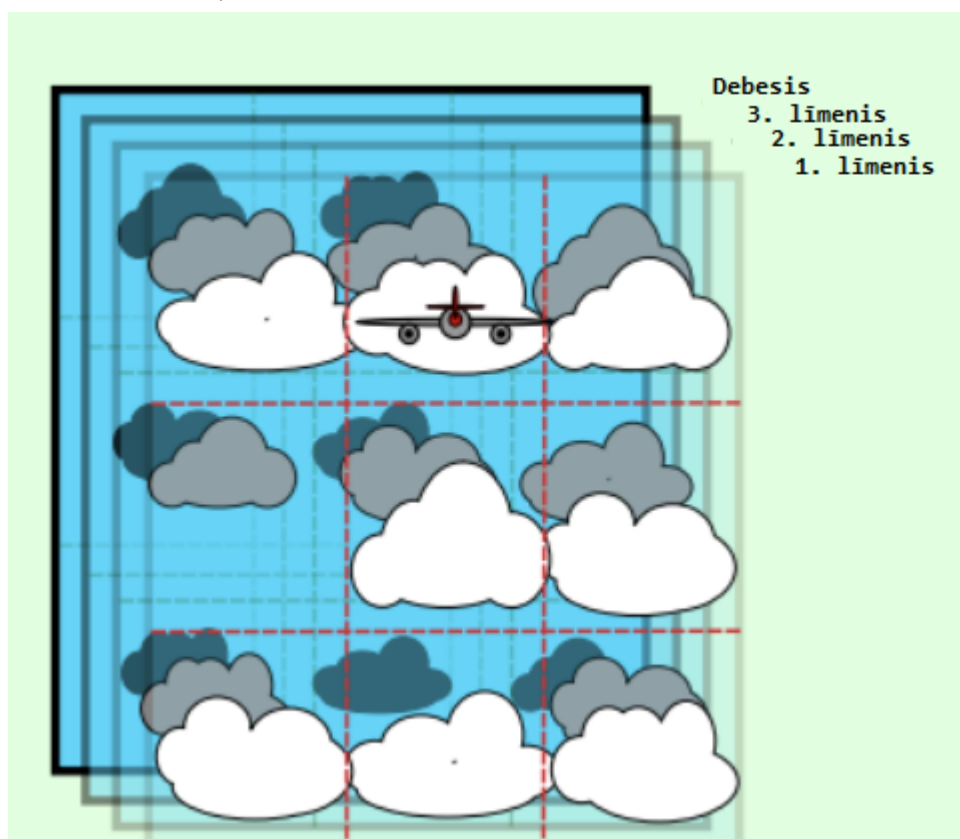
Šajā uzdevumā tiek attīstīta spēja izveidot un salīdzināt objektu (puķu) modeļus (attēlus), identificējot būtiskās īpašības un atribūtus (ko attēlo puķu veidi, krāsas un lapu skaits, kas vairākām puķēm ir kopīgs).

Modeļiem ir izšķiroša loma objektu klasificēšanā un īpašību izvēlē, lai definētu dažādas objektu grupas. Tāpēc modeļa izveides gaitā ir jāanalizē objektu īpašības. Šajā kontekstā uzdevums ir par modeļu atpazīšanu. Turklāt uzdevums ietver dekompozīcijas un novērtēšanas prasmes, jo ir nepieciešams izgūt un izpētīt objektu īpašības, kā arī atrisināt to pa soļiem, aplūkojot puķes pēc krāsas, puķu veida un lapu skaita, lai pārbaudītu, vai katram no šiem kritērijiem ir iespējams katrā vāzē ievietot vienādu puķu skaitu.

Izvairoties no mākoņiem

Īrija

Pilots, lidojot cauri iedomātam 3x3x3 režģim, gatavojas sastapties ar mākoņiem. Viņš vēlas no šiem mākoņiem izvairīties, tā vietā lidojot cauri skaidrām debesīm. Katru reizi, kad pilots pagriežas, lidmašīna, virzoties uz priekšu par vienu soli, lidos augšup, lejup, pa kreisi, pa labi vai pa diagonāli. Baltie mākoņi ir vistuvāk, gaiši pelēkie mākoņi ir tālāk, un tumšākie mākoņi ir vistālāk.



Tieši šobrīd lidmašīna gatavojas ielidot 3x3x3 režģī. Ja tā nepagriezīsies, tā ietrieksies baltajā mākonī augšējās rindas vidū.

Jautājums / Izaicinājums

Kuri norādījumi ļaus izlidot cauri 3x3x3 režģim, neielidojot mākoņos?

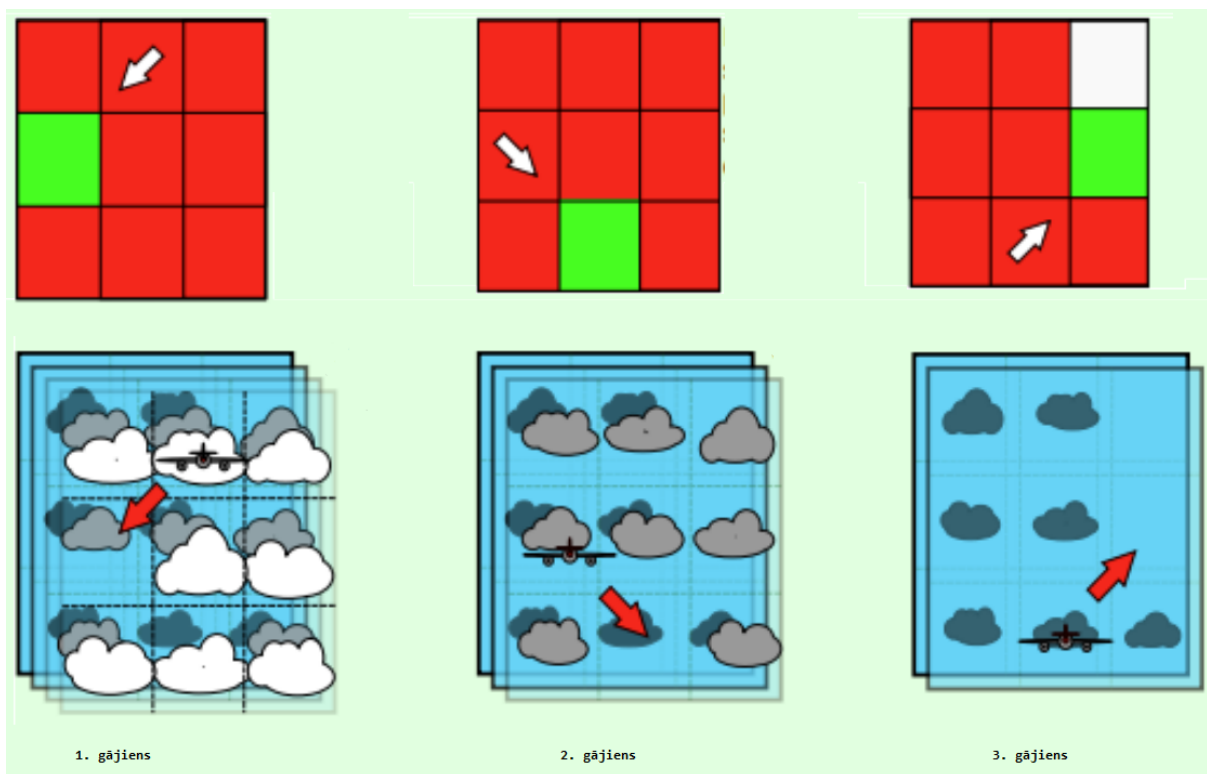
Atbilžu varianti / Interaktivitātes apraksts

A.	B.	C.	D.
			

Atbildes skaidrojums

Pareizā atbilde ir C.

Šis uzdevums ietver mākoņu izvietojuma atpazīšanu. Tie visi ir izvietoti trīs slāņos un attēlo zonas, kurās lidmašīna nedrīkstētu ielidot. Aplūkojot trīs slāņus, kas attēloti kā tabulas, ir vieglāk saprast, kurā virzienā jalido.



Visu lidojumu var sadalīt trīs “gājienos”, kas attiecas pēc kārtas uz 1., 2. un 3. slāni:

1. slānī pilotam ir tikai viena rūtiņa kuru var izmantot (iekrāsota zaļa),
2. slānī pilotam ir tikai viena rūtiņa kuru var izmantot,
3. slānī pilotam ir divas rūtiņas bez mākoņiem, bet tikai vienu rūtiņu var izmantot.

Šī ir informātika

Šis uzdevums prasa izprast divdimensiju attēlu kā trīsdimensiju telpu un pēc tam tajā orientēties. Navigācija trīsdimensiju telpā parādās daudzās dažādās ar informātiku saistītās jomās, sākot no datorspēlēm līdz telpisko datu attēlošanai. Diskrēta telpa, piemēram, šajā uzdevumā ilustrētais lodziņu režģis, ir grafa piemērs. Grafu navigācijai ir daudz algoritmu. Lai gan šajā uzdevumā ir tikai viens iespējamais ceļš, daudzas reālās pasaules datorzinātņu problēmas ietver izvēli starp dažādiem ceļiem grafā, piemēram, īsākā ceļa atrašanu. Algoritmi arī orientējas telpās ar vairāk nekā trim dimensijām, risinot sarežģītas problēmas, optimizējot procesus un pierādot programmatūras pareizību.

Šī ir skaitļošanas domāšana

Novērtēšana. Studentam jānovērtē dažādu algoritmu, kas izmanto ierobežotu programmēšanas valodu, pareizība.

Pārvadāšanas stratēģija

Azerbaidžāna

Bebram jāpaņem pieci apslimuši eksotisku sugu kaķi, lai tos nogādātu pie veterinārārsta. Viņam ir vairāki dzīvnieku pārvadāšanas konteineri. Katram konteineram ir pieļautā kopējā svara ierobežojums kilogramos.

Konteiners	Pieļaujamā svara ierobežojums kilogramos
A	10
B	15
C	20
D	5

Bebra mērķis ir pārvest visus kaķus, izmantot pēc iespējas mazāku konteineru skaitu, nevienam nepārsniedzot atļautā svara ierobežojumu. Katram kaķim ir zināms tā svars:

Kaķa vārds	Svars kilogramos
<i>Tiny</i>	3
<i>Blacky</i>	7
<i>Fatboy</i>	10
<i>Colorina</i>	5
<i>Lazyboy</i>	6

Jautājums / Izaicinājums

Kura no tālāk minētajām iespējām ļaus pārvadāt visus kaķus, izmantojot pēc iespējas mazāku konteineru skaitu?

Atbilžu varianti / Interaktivitātes apraksts

a) *Tiny* un *Blacky* konteinerā B; *Fatboy*, *Colorina* un *Lazyboy* konteinerā C.

b) *Tiny* un *Blacky* konteinerā A, *Fatboy* konteinerā B, *Colorina* konteinerā D, *Lazyboy* konteinerā C.

c) *Tiny* un *Colorina* konteinerā A, *Blacky* un *Lazyboy* konteinerā B, *Fatboy* konteinerā C.

d) *Tiny*, *Blacky* un *Fatboy* konteinerā C, *Colorina* konteinerā D, *Lazyboy* konteinerā A.

e) *Fatboy* un *Colorina* konteinerā B, *Tiny*, *Blacky* un *Lazyboy* konteinerā C.

Atbildes skaidrojums

Pareizā atbilde ir e variants: *Fatboy* (10 kg) + *Colorina* (5 kg) = 15 kg ietilpst konteinerā B (15 kg) *Tiny* (3 kg) + *Blacky* (7 kg) + *Lazyboy* (6 kg) = 16 kg ietilpst konteinerā C (20 kg). Lai ātri atrastu pareizo atbildi, varat saskaitīt katrā variantā izmantoto konteineru skaitu. a un e variantā tiek izmantoti divi konteineri. c un d variantā tiek izmantoti trīs konteineri. b variantā tiek izmantoti četri konteineri. Tātad, jūs sākat, pārbaudot a un e variantus.

Izrādās, ka a variants nav iespējams, jo trīs kaķi kopā svērtu 21 kg. Tas pārsniedz konteineru C ietilpību (20 kg). Tātad paliek tikai e atbilde, un tā atbilst svara ierobežojumiem.

Šī ir informātika

Šis uzdevums modelē resursu sadali ar ierobežojumiem, kas ir informātikas pamatkonceptija. Tas atgādina mugursomas problēmas variāciju, kombinatorisku optimizāciju, kur ir dots priekšmetu skaits un to svars, kā arī noliktava ar svara ierobežojumu; risinājums maksimāli palielinās to priekšmetu skaitu, kurus var uzglabāt, nepārsniedzot svara ierobežojumu.

Šajā gadījumā kaķi ir priekšmeti, un krātuvi pārstāv konteineri:

katram konteineram ir svara ierobežojums. Mērķis ir samazināt izmantoto konteineru skaitu.

Studentiem jānovērtē derīgas kombinācijas, lai optimizētu izmantotos resursus, kas ir galvenā ideja operētājsistēmās, atmiņas pārvaldībā un failu sistēmās.

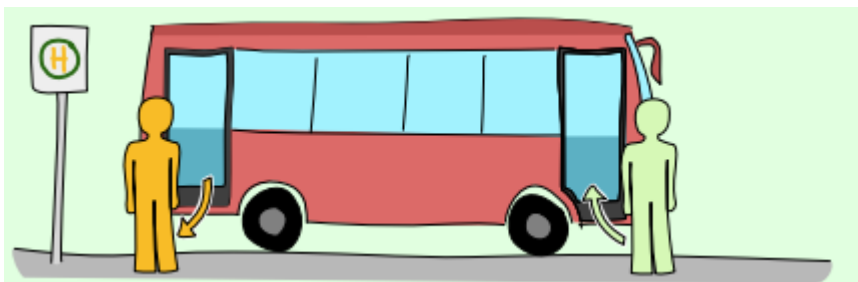
Šī ir skaitļošanas domāšana.

Uzdevums ietver dekompozīciju, abstrakciju, kombināciju sistemātisku testēšanu un risinājumu salīdzināšana optimalitātes noteikšanai (izmantots vismazāk konteineru).

Autobuss

Beļģija

Starppilsētu autobuss bez vadītāja kursē pa noteiktu maršrutu. Parasti autobuss apstājas pieturā, kad cilvēki gaida iekāpšanu vai kad pasažieri vēlas izkāpt no autobusa. Taču šobrīd autobusa dators nedarbojas pareizi, un autobuss apstājas pieturā tikai tad, kad ir pasažieri, kas gaida iekāpšanu, un vienlaikus ir pasažieri, kas vēlas izkāpt no autobusa. Autobuss vienmēr apstājas galapunktos (pirmajā un pēdējā pieturā), neskatoties uz aprakstīto datora kļūmi.



Autobusu pieturas ir numurētas no 1 līdz 5. Tajās gaida šādi pasažieri: -

- 1. pieturā Marija vēlas iekāpt un viņa vēlas izkāpt
- 2. pieturā -
- 1. pieturā Vasils vēlas iekāpt un viņš vēlas izkāpt
- 4. pieturā -
- 2. pieturā Simons vēlas iekāpt un viņš vēlas izkāpt
- 5. pieturā -
- 3. pieturā Jeļena vēlas iekāpt un viņa vēlas izkāpt
- 4. pieturā -
- 3. pieturā Kalina vēlas iekāpt un viņa vēlas izkāpt
- 5. pieturā.

Jautājums / Izaicinājums

Kāds ir visu to pasažieru saraksts, kuri nevarēs nokļūt savā vēlamajā pieturā?

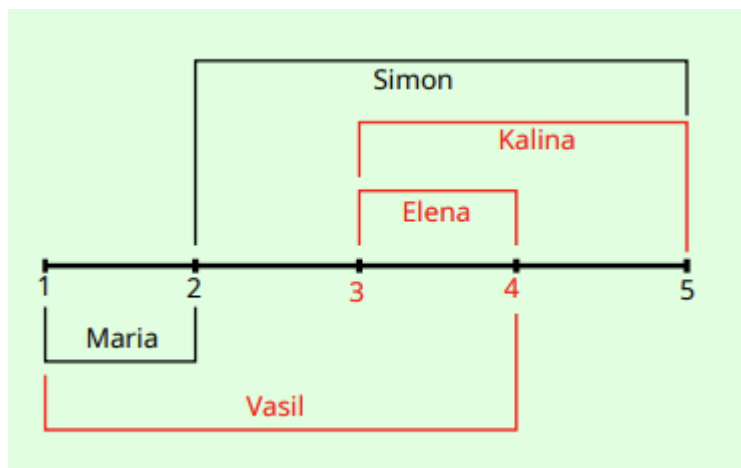
Atbilžu varianti / Interaktivitātes apraksts

- A. Jeļena
- B. Vasils un Kalina
- C. Vasils, Jeļena un Kalina
- D. Vasils

Atbildes skaidrojums

Pareizā atbilde: C

Autobuss neapstāsies 3. un 4. pieturā, jo, vai nu tur negaida cilvēki, vai arī nav pasažieru, kuri tur vēlas izkāpt. Tomēr autobuss pieturēs 1., 2. un 5. pieturā. Tādējādi divas personas, kas gaida 3. pieturā, Jeļena un Kalina, nevarēs iekāpt un tāpēc nerasniegs savas vēlamās pieturas. Vasils iekāpj 1. pieturā un vēlas izkāpt 4. pieturā, bet autobuss tur neapstājas. Tādējādi Vasils, Jeļena un Kalina nerasniegs savas vēlamās pieturas.



Interaktīvs jautājums/izaicinājums

Atzīmējiet visus pasažierus, kuri var nokļūt savā vēlamajā pieturā?

Atbilde:

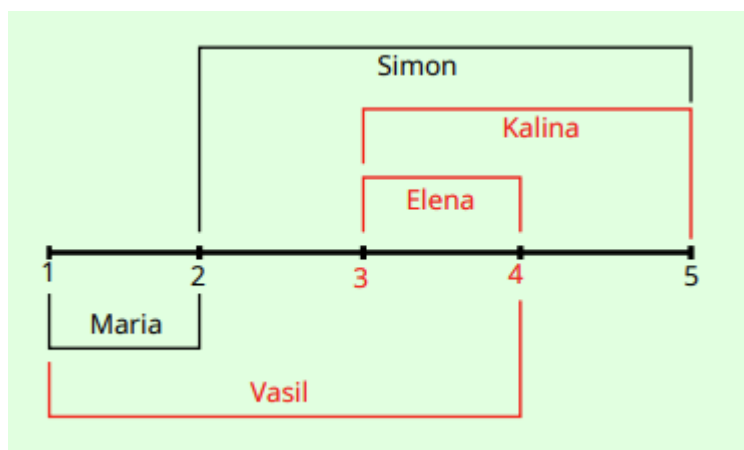
Marija un Saimons

Atbildes skaidrojums

Tikai Marija un Saimons nokļūs savās vēlamajās pieturās.

Autobuss neapstāsies 3. un 4. pieturā, jo vai nu tur nav cilvēku, kas gaida, vai arī nav pasažieru, kuri tajās vēlētos izkāpt. Tomēr autobuss pieturēs 1., 2. un 5. pieturā.

Tādējādi divas personas, kas gaida 3. pieturā, Jeļena un Kalina, nevarēs iekāpt un tāpēc nerasniegs savas vēlamās pieturas. Vasils iekāpj 1. pieturā un vēlas izkāpt 4. pieturā, bet autobuss tur neapstājas. Tādējādi Vasils, Jeļena un Kalina nerasniegs savas vēlamās pieturas. To ilustratīvi parāda arī sarkanās līnijas nākamajā diagrammā. Izslēgšanas rezultātā atlikušie pasažieri Marija un Simons nokļūs savās vēlamajās pieturās. To ilustratīvi parāda arī melnās līnijas nākamajā attēlā.



Alternatīvs skaidrojums: No uzdevuma apraksta autobuss apstāsies 1., 2. un 5. pieturā, jo autobuss vienmēr pieturēs 1. un 5. pieturā, un 2. pietura ir vienīgā, kurā ir pasažieris, kas vēlas iekāpt un izkāpt. Tādējādi mums jāatzīmē tikai pasažieri, kas vēlas iekāpt un izkāpt 1., 2. un 5. pieturā, un nav jāņem vērā pasažieri, kas vēlas iekāpt vai izkāpt 3. un 4. pieturā. Tam atbilst tikai Marija (1. un 2. pietura) un Simons (2. un 5. pietura). To ilustratīvi parāda arī melnās līnijas iepriekš redzamajā diagrammā.

Šī ir informātika

Uzdevums prasa atpazīt un ņemt vērā izmaiņas apstāšanās shēmā. Parastais modelis ir "apstāties, ja kāds iekāpj VAI izkāpj". Izmainītais (bojātais) modelis ir "apstāties, ja kāds iekāpj UN izkāpj". Pirmā un pēdējā pietura attēlo fiksētu, beznosacījumu modeli. Problēmas izpratne ietver noteikumu vai nosacījumu kopumu, ko ievēro autobusa dators. Bojājums ievieš konkrētu, mainītu noteikumu. Šis uzdevums ir vienkārša reālās pasaules transporta sistēmas modelēšana ar konkrētiem noteikumiem, kas regulē tās darbību. Lēmums apstāties ir atkarīgs no datiem (vai cilvēki gaida un vai pasažieri vēlas izkāpt) un loģiskajiem nosacījumiem, kas tiek piemēroti šiem datiem. Datu apstrāde un loģiskās operācijas ir informātikas centrā.

Šī ir skaitļošanas domāšana

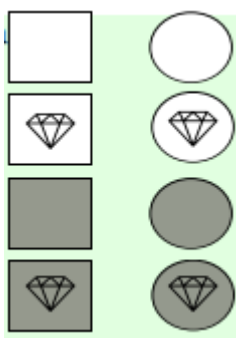
Visām izmaiņām, kas notiek laika gaitā, ir jāseko līdzi. Datordomāšanā tā ir tipiska notikumu izsekošana ciklā un izmaiņu veikšana sarakstos atkarībā no loģisko nosacījumu izpildes. Identificējamais modelis ir tas, vai pieturā ir gan pasažieri, kas vēlas iekāpt, GAN izkāpt, kā arī triviāli sākuma un beigu pieturu robežgadījumi.

Trešais

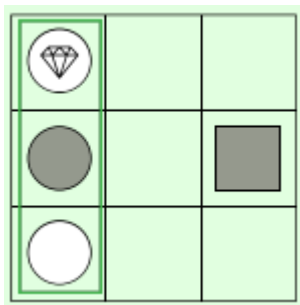
Brazīlija

Bebri Anna un Bento spēlē spēli ar nosaukumu "Trešais". Viņi izmanto 3x3 rūtiņas lielu spēles laukumu un astoņas unikālas figūriņas ar trim raksturīgām īpašībām:

- Krāsa: pelēka vai balta.
- Forma: kvadrāts vai aplis.
- Marķējums: ar vai bez romba.

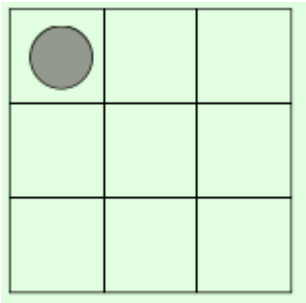


Lai uzvarētu, spēlētājam jāaizpilda laukuma rinda, kolonna vai diagonāle, kur trim figūriņām ir vismaz viena kopīga iepriekš aprakstītā īpašība. Piemēram:

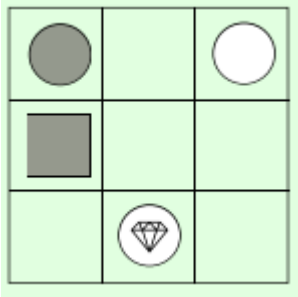


Iepriekš redzamajā spēles laukumā viens spēlētājs ir uzvarējis, jo novietojis trešo figūriņu pirmajā kolonnā, iegūstot trīs figūriņas ar vienādu raksturlielumu vienā līnijā: apla formas figūriņas vienā kolonnā.

Spēle sākas ar tukšu spēles laukumu. Spēlētāji gājienus izdara pēc kārtas, bet pretinieks izvēlas figūriņu, kas jānovieto. Pēc figūriņas novietošanas spēlētājs izvēlas nākamo figūriņu, kas laukumā jānovieto pretiniekam utt. Anna sāka spēli, izvēloties pelēko apli bez romba (), lai Bento spēlētu, un pēc tam Bento to novieto uz spēles laukuma.



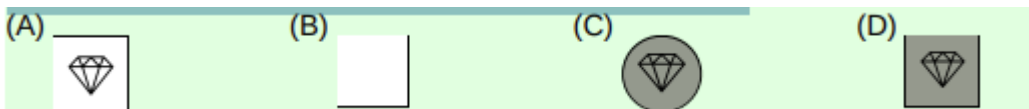
Viņi turpināja pārmaiņus, līdz spēles laukumā iestājās šāda situācija:



Jautājums / Izaicinājums

Kuru no tālāk norādītajām figūrām Anna var dot Bento novietot uz galda, lai nodrošinātu, ka viņš NEuzvar savā gājienā?

Atbilžu varianti / Interaktivitātes apraksts

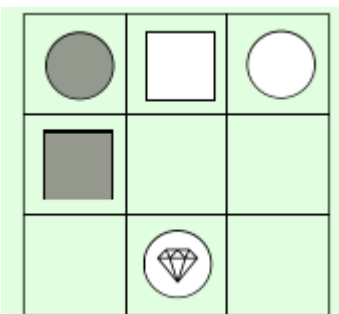


Atbildes skaidrojums

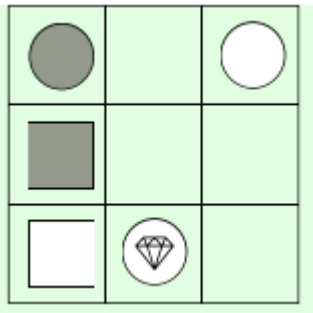
Pareizais variants ir (A).

Novietojot figūru no (A) iespējas, tiek nodrošināts, ka Bento nevarēs novietot trīs figūras ar kopīgu īpašību, tādējādi neļaujot viņam uzvarēt. (B) iespēja ir nepareiza. Ir divas iespējamās vietas, kur Bento varētu novietot šo figūru un uzvarēt:

- Viņš var novietot trīs figūras bez romba pirmajā rindā.

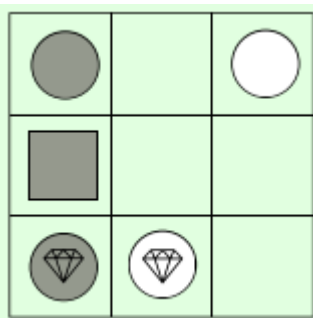


- Viņš var novietot trīs figūras bez romba pirmajā kolonnā.

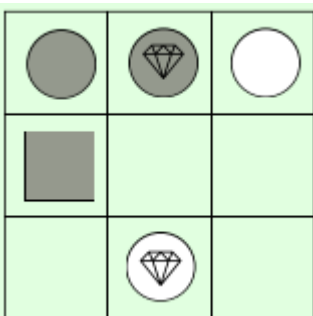


(C) iespēja ir nepareiza. Ir divas iespējamās vietas, kur Bento varētu novietot šo figūru un uzvarēt:

- Viņš var novietot trīs pelēkas figūras pirmajā kolonnā.

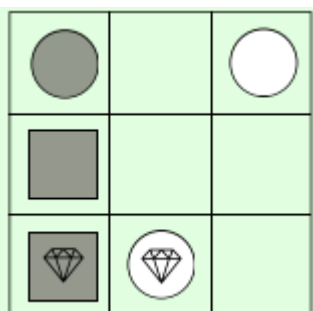


- Viņš var novietot trīs figūras ar apļa formu pirmajā rindā.



(D) iespēja ir nepareiza. Ir viena iespējama vieta, kur Bento varētu novietot šo figūriņu un uzvarēt:

- Viņš var novietot trīs pelēkas figūriņas pirmajā kolonnā.



Šī ir informātika

Atribūtu jēdziens datorzinātnēs ir saistīts ar objekta raksturlielumu vai īpašību. Piemēram, objektorientētās programmēšanas valodās atribūti ir dati, kas apraksta objekta stāvokli (piemēram, krāsa, forma, izmērs utt.). Spēlē šī ideja izpaužas ļoti skaidri: katrai figūriņai spēlē ir trīs binārie atribūti, ir $2^3=8$ unikālas figūriņas, katrai ar atšķirīgu šo atribūtu kombināciju. Spēles mērķis ir novietot trīs figūriņas, kurām ir viens un tas pats atribūts, taisnā līnijā neatkarīgi no tā, kurš tās novietoja. Un, lai uzvarētu šajā spēlē, jums ir jāanalizē katrs gājieni un jāparedz pretinieka iespējamie atbildes gājieni. Tas nozīmē, ka jums ir jāizvērtē potenciālie gājieni un to sekas, lai izdarītu labāko izvēli. Tas padara spēli labi piemērotu programmēšanai, izmantojot lēmumu kokus, piemēram, "ja-citādi" paziņojumus. Turklāt spēlētājiem ir jāievēro spēles noteikumi. Noteikumi ir ļoti svarīgi ne tikai šajā spēlē, bet arī datu apstrādē datorsistēmās. Tie palīdz definēt attēlu failu, kredītkaršu numuru un daudzu citu strukturētu datu tipu formātus.

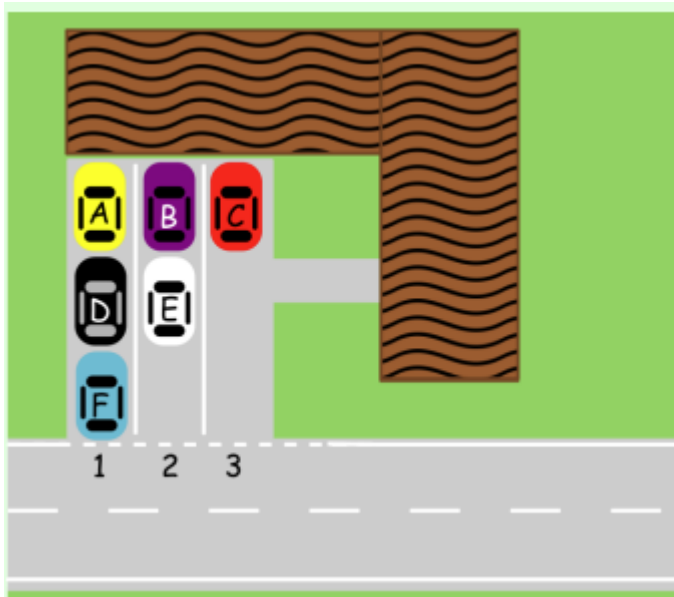
Šī ir skaitļošanas domāšana.

Lai atrisinātu šo uzdevumu, skolēniem ir jāizvērtē dažādi gājieni un jāparedz pretinieka iespējamās atbildes, lai katrā kārtā pieņemtu labāko lēmumu (Novērtēšana). Turklāt figūru pozīciju modeļu atpazīšana palielina viņu izredzes uzvarēt un palīdz novērst negaidītus pārsteigumus no pretinieka puses (Modeļu atpazīšana).

Stāvvietā

Slovēnija

Aija organizē dzimšanas dienas ballīti uz kuru uzaicinātie draugi ierodas ar automašīnām. Aijas piebraucamajā ceļā ir vieta deviņām automašīnām, kas jānovieto trīs kolonnās pa trim automašīnām viena pēc otras. Katram viesim tiek lūgts novietot automašīnu pirmajā brīvajā vietā jebkurā kolonnā.



Šajā piemērā automašīna A ieradās pirmā, automašīna D - otrā, automašīna C - trešā, un automašīnai F jāaizbrauc, pirms to var izdarīt automašīna D.

Viesi ierodas šādā secībā: Anna, Bobs, Cilda, Deivids, Elena, Frenks, Georgs, Heilija, Ivans.

Viņi pametīs ballīti šādā secībā: Georgs, Deivids, Frenks, Ivans, Heilija, Cilda, Bobs, Anna, Elena.

Jautājums / Izaicinājums

Sakārtojiet automašīnas kolonnās viesu ierašanās secībā tā, lai neviena automašīna neaizšķērsotu automašīnu, kurai jāaizbrauc agrāk. Izvēlieties pareizo variantu.

Atbilžu varianti / Interaktivitātes apraksts



Atbildes skaidrojums

Pareizā atbilde ir A.

Mēs varam analizēt situāciju katram viesu pārim:

- Anna ierodas pirms Cildas, un Cilda dodas prom pirms Annas. Tātad viņas var novietot automašīnas vienā kolonnā. Tas pats attiecas uz Annu un Bobu, kā arī uz Bobu un Cildu.

- Anna un Elena ierodas tādā pašā secībā, kādā viņi aizbrauc, tāpēc viņi nevar novietot automašīnas vienā kolonnā. Tas pats attiecas uz Bobu un Elenu; Cildu un Elenu; Deividu pret Elenu, Frenku, Heiliju un Ivanu; Frenku attiecībā pret Heiliju un Ivanu; Georgu pret Heiliju un Ivanu.

Tātad Anna, Bobs un Cilda visi var novietot automašīnu 1. kolonnā savā ierašanās secībā. Kad ierodas Deivids, lai viņš novieto automašīnu 2. kolonnā, jo 1. kolonna ir pilna. Kad ierodas Elena, viņai jānovieto automašīna citā kolonnā nekā Deivida automašīna, tāpēc viņai jānovieto automašīna 3. kolonnā. Tad ierodas Frenks, arī viņam jānovieto automašīna citā kolonnā nekā Deivida automašīna, tāpēc lai viņš novieto automašīnu 3. kolonnā, kas ir labi, jo viņš aizbrauks pirms Elenas, kura tur ir novietojusi automašīnu. Kad Georgs ierodas, tā kā viņš ir pirmais, kas dosies prom, viņam jānovieto kā pēdējam kolonnā, tāpēc ļaujiet viņam novietot automašīnu kā trešajai automašīnai 3. kolonnā. Šajā brīdī mums ir divas vietas 2. kolonnā. Ierodas Heilija, kam seko Ivans, un Ivans aizbrauks pirms Heilijas, tāpēc viņi var novietot automašīnu 2. kolonnā ierašanās secībā. Pēc ballītes draugi aizbrauc šādā secībā: Georgs, Deivids, Frenks, Ivans, Heilija, Cilda, Bobs, Anna, Elēna. Nevienu neaizšķērso kāda cita automašīna, jo:

1. Vispirms aizbrauc Deivids un Georgs.
2. Kad Georgs aizbrauc, Frenka automašīna tiek atbloķēta.
3. Deivids atbloķē Boba automašīnu.
4. Frenks atbloķē Cildas automašīnu.
5. Ivans atbloķē Heilijas automašīnu.
6. Heilija atbloķē Elēnas automašīnu.

7. Cilda iztukšo 2. kolonnu.
8. Bobs atbloķē Annas automašīnu.
9. Anna un Elēna var aizbraukt, jo viņu automašīnas ir vienīgās viņu kolonnās.

Atbilde B ir nepareiza, jo Deivids ierodas pirms Frenka, tāpēc viņš nevar novietot automašīnu aiz viņa.

Atbilde C ir nepareiza, jo Deividam jāaizbrauc pirms Heilijas, bet Heilija ir novietojusi automašīnu aiz viņa.

Atbilde D ir nepareiza, jo, lai gan viesu automašīnas ir novietotas to ierašanās secībā, Georga automašīna ir bloķēta, un viņam jāaizbrauc pirmajam.

Šī ir informātika

Automašīnu novietošanas kolonnās Aijas piebraucamajā ceļā atbilst tam, ko informātikā sauc par steku. Steks ir datu struktūra, ko izmanto, lai glabātu secīgas datu vai objektu kolekcijas. Datu struktūras ir svarīgas informātikā, jo tās ļauj mums organizēt datus tā, lai tos varētu efektīvi apstrādāt, uzglabāt un no tām izgūt datus. Steks darbojas kā trauku kaudze: kad ir jāuzglabā jauns objekts, tas tiek novietots steka augšpusē. Vienmēr var tikai paskatīties (ielūkoties) vai noņemt (izņemt) objektu no steka augšas. Datu struktūru ar šāda veida uzvedību sauc arī par "pēdējais iekšā, pirmais ārā". Tātad, izmantojot steku, var tieši piekļūt un noņemt tikai pēdējo stekam pievienoto elementu, tāpat kā automašīnas šajā uzdevumā. Problēma, kas saistīta ar daudzu automašīnu izvietošanu vairākos stekos, lai tās nebloķētu viena otru, kad tām ir jāaizbrauc, un līdzīgas problēmas, tiek sauktas par vairāku steku problēmām, un algoritmi to risināšanai tiek izmantoti resursu piešķiršanā uzdevumiem, piemēram, daudzkodolu procesoru ierīcēs, kur var notikt paralēla apstrāde.

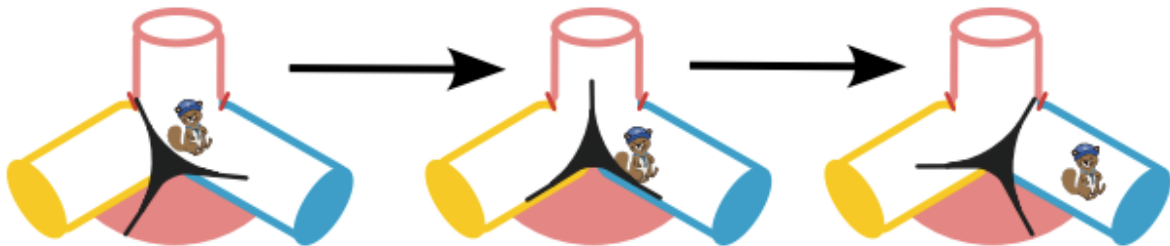
Šī ir skaitļošanas domāšana

Skolēniem ir jāseko automašīnu ierašanās secībai un jāpārbauda, kur tās var novietot, un pēc tam jāpārbauda arī, kad katra automašīna varēs atstāt savu stāvvietu. Ar šī uzdevuma palīdzību viņi attīsta algoritmisko domāšanu un loģisko spriešanu.

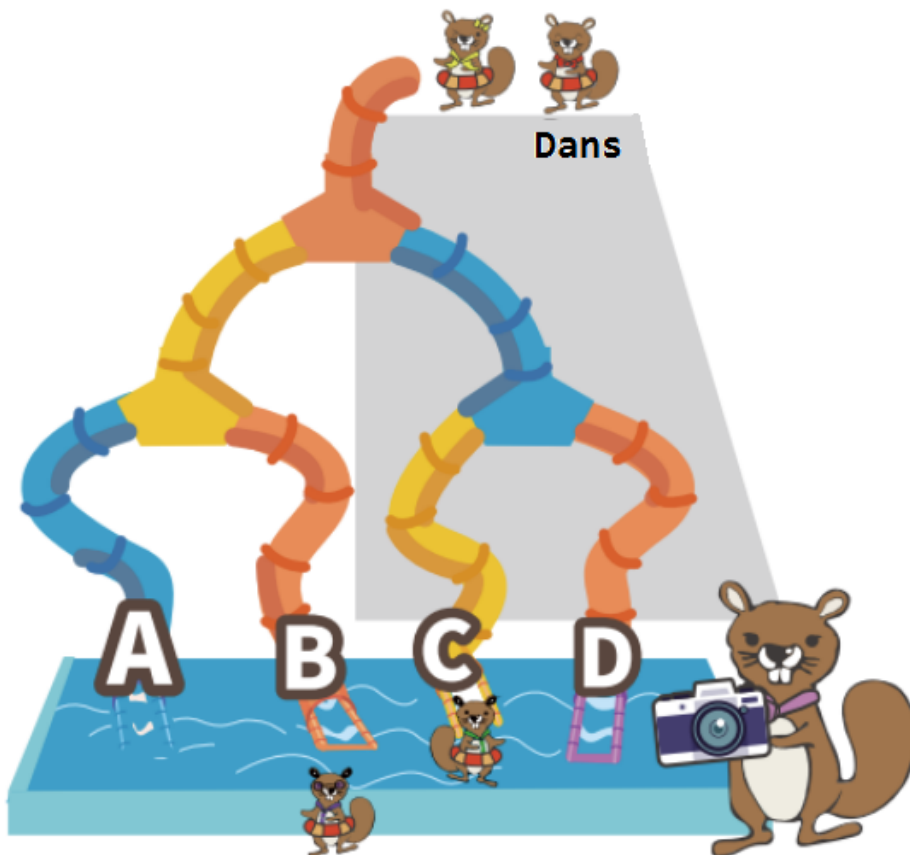
Aizraujošais nobrauciens

Taivāna

Bebru ūdens atrakciju parkā ir aizraujoša dinamisku ūdens slidkalniņu kaskāde, kas beidzas ar baseinu. Šīs diagrammas parāda, kā tas darbojas: mehānisms kontrolē katra slidkalniņa sazarojuma izejas virzienu; ikreiz, kad bebris izslīd cauri atzarojumam, izejas virziens mainās.



Kā redzams nākamajā attēlā, mazais bebris Dans vēlas izmēģināt šos ūdens slidkalniņus. Bebru mamma vēlas zināt, pa kuru slidkalniņu Dans iekritīs baseinā, lai varētu uzņemt labas bildes. Pirmais bebru iekrīt baseinā no B slidkalniņa, un pēc tam nākamais - no C slidkalniņa. Rindā uz nobraucieni gaida vēl viens bebris un Dans būs nākamais pēc tā.



Jautājums / Izaicinājums

No kura slidkalniņa mazais bebrs Dans iekritīs baseinā?

Atbilžu varianti / Interaktivitātes apraksts

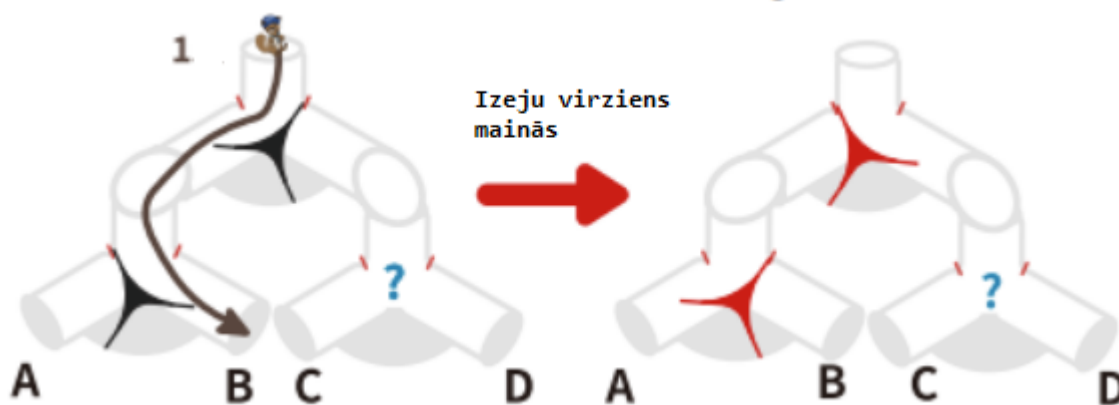
- A. No A slidkalniņa
- B. No B slidkalniņa
- C. No C slidkalniņa
- D. No D slidkalniņa

Atbildes skaidrojums

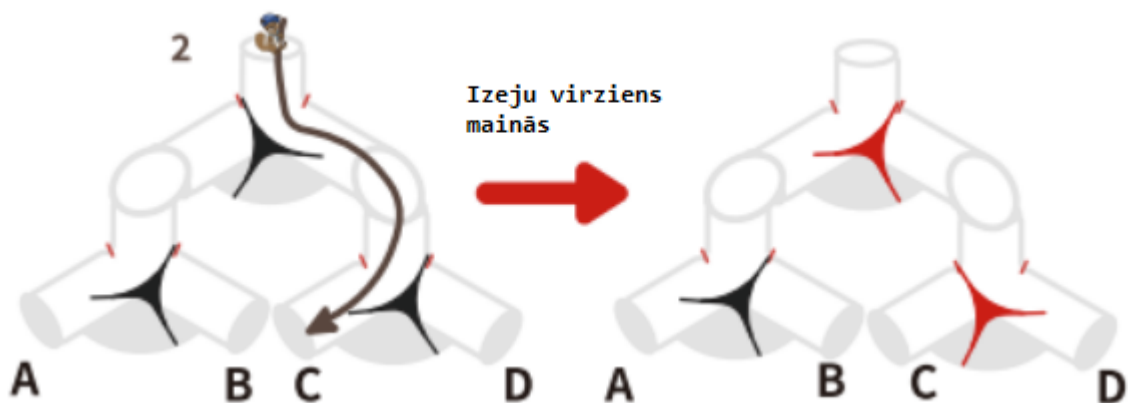
Atbilde ir D.

Tā kā izejas virziens mainās katru reizi, kad bebrs izslīd cauri atzarojumam, mēs varam noteikt atzarojuma izejas virzienu pēc tam, kad pirmie divi beбри ir izslīdējuši cauri.

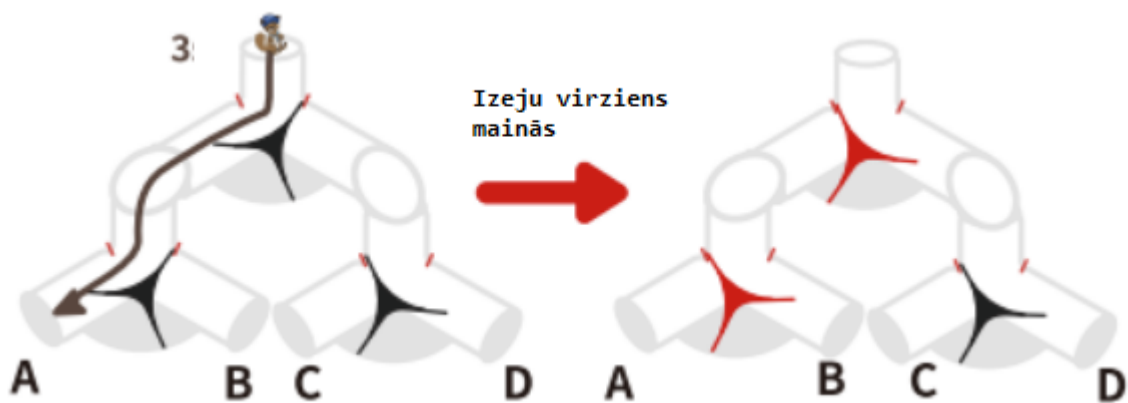
Kā redzams zemāk esošajā attēlā, pirmais bebrs iekrīt baseinā no B slidkalniņa; Mēs varam secināt, ka bebrs pirmajā atzarojumā iet pa kreisi un otrajā pa labi. Pēc tam šo divu atzarojumu izejas virziens mainās: pirmais atzarojums tiek atvērts pa labi, bet otrais (apakšējais kreisais) atzarojums - pa kreisi. Šajā brīdī mēs joprojām neko nezinām par apakšējā labā atzarojuma izejas virzienu.



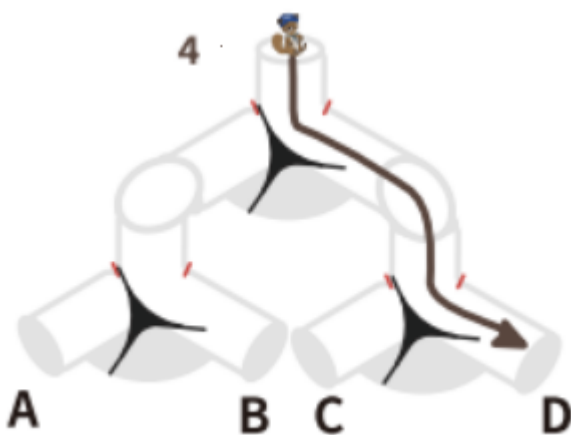
Līdzīgi, kā parādīts attēlā zemāk, nākamais bebrs iekrīt baseinā no C slidkalniņa; mēs varam secināt, ka apakšējā labā atzarojuma izejas virziens bija pa kreisi pirms bebra izslīdēšanas un pēc tam pagriežas pa labi.



Šajā brīdī mēs esam noskaidrojuši visu atzarojumu izejas virzienus, tāpēc varam izsekot maršrutu bebram pirms Dana, un atzarojumu izejas virzieniem pēc tam, kā parādīts attēlā zemāk.



Visbeidzot, mēs zinām, ka Dans slīdēs pa maršrutu, kas parādīts attēlā zemāk, un iekritīs baseinā pa D slidkalniņu.



Šī ir informātika

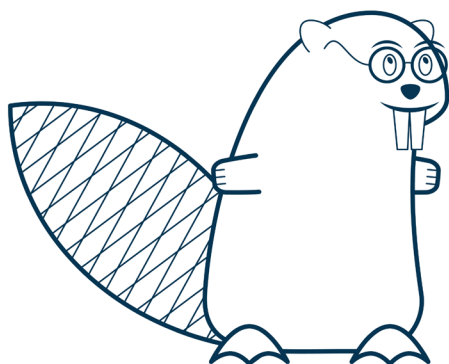
Šajā uzdevumā katra atzarojuma izejas virziens nosaka, pa kuru slidkalniņu bebrs slīdēs tālāk. Tas ir līdzīgi tam, kā programmēšanā darbojas nosacījuma instrukcijas. Tipisks

nosacījums atbilst šādam modelim: "Ja nosacījums ir patiess, dariet to; pretējā gadījumā dariet ko citu." Šajā uzdevumā mēs varam aprakstīt atzarojuma mehānisma darbību šādi: "Ja tas ir vērsts pa kreisi, bebrš tam slīdēs cauri pa kreiso slidkalniņu un mehānisms pārslēgsies uz labo pusi; pretējā gadījumā bebrš slīdēs cauri pa labi un mehānisms pārslēgsies uz kreiso pusi." Kad bebrš izslīd cauri katram atzarojuma mehānismam, tas pats process atkārtojas. Bebrš seko mehānisma pašreizējam virzienam, un mehānisms pārslēdzas pretējā virzienā. Tāpēc katram mehānismam ir jāatceras patreizējais virziens, kas ir informācijas daļa, ko programmēšanā var attēlot kā mainīgo. Piemēram, mēs varam izmantot 0, lai attēlotu "pa kreisi", un 1, lai attēlotu "pa labi". Kad bebrš pārvietojas pa struktūru, programma varētu pārbaudīt katra mainīgā vērtību (mehānisma virzienu), izlemt, kuru ceļu izvēlēties, un atjaunināt mainīgo, lai atspoguļotu jauno stāvokli. Šis mehānisms ir līdzīgs arī triggeru digitālajās shēmās — pamatvienībai, kas glabā vienu datu bitu un pārslēdzas starp diviem stāvokļiem. Katrs atzarojums uzvedas kā triggeris, mainot savu stāvokli katru reizi, kad tas tiek izmantots.

Lai noskaidrotu, kur bebrš nonāks, mēs soli pa solim simulējam katru darbību. Tas ir līdzīgi koda izsekošanai programmēšanā, kur programmas loģika tiek sekota rindiņai pa rindiņai, lai redzētu, kādu rezultātu tā rada.

Šī ir skaitļošanas domāšana

Studentiem ir jāpielieto loģiskā spriešana, lai izsekotu izmaiņām atzarojuma virzienos un precīzi prognozētu rezultātus. To darot, viņiem jāignorē stāsta elementi, piemēram, kas ir bebrš vai kā izskatās augļu dārzs, un tā vietā jākoncentrējas uz uzdevuma būtisko struktūru: katra atzarojuma secību un stāvokli. Šis vienkāršošanas process, koncentrējoties tikai uz būtiskajiem aspektiem, ir pazīstams kā abstrakcija.



copyright Bebras