VISMA

Bebr[a]s '25

2025. gada 1. kārtas uzdevumi ar atbildēm

7.-8. klase

INFORMATĪVIE ATBALSTĪTĀJI

Izglītības un zinātnes ministrija

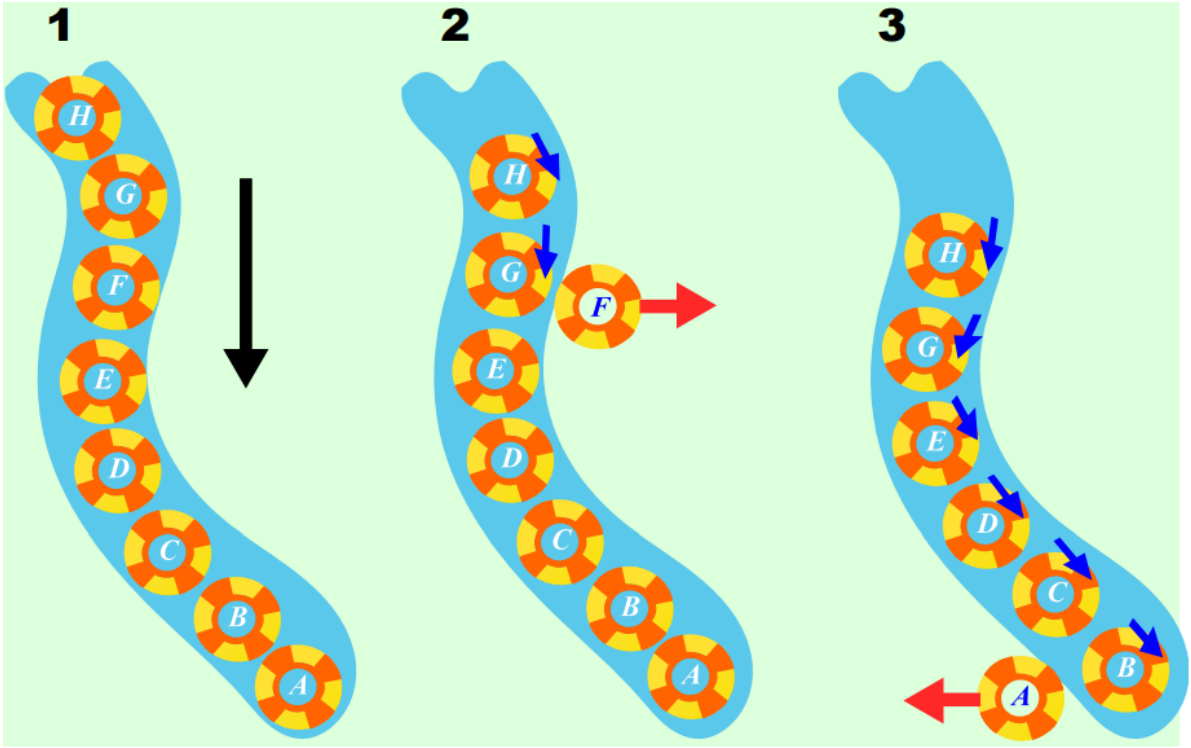
start(it)
www.startit.lv

Satura rādītājs

Piepūšamie riņķi

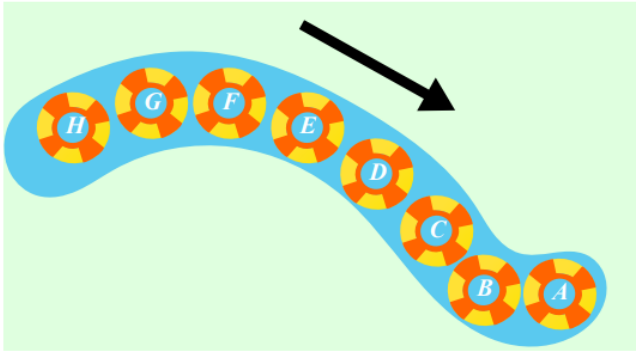
Kanāda

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



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

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



,

un tiek izvilkts 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.

Sākumā, 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 spoguli, kas atrodas uz vertikālas ass)	
	R (pagriezt par 90 grādiem pulksteņrādītāja virzienā)	

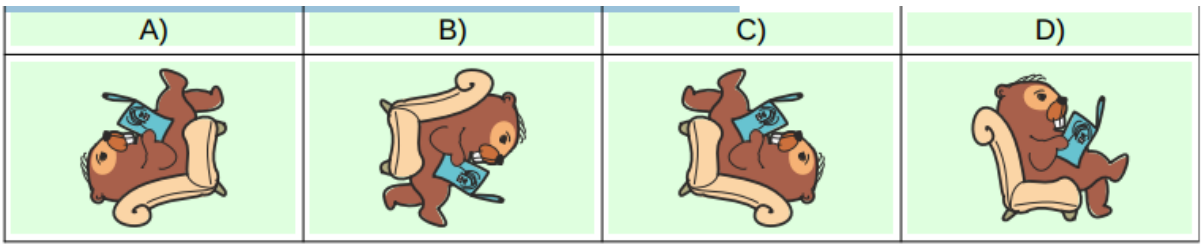
Sākotnējais attēls ir šāds:

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

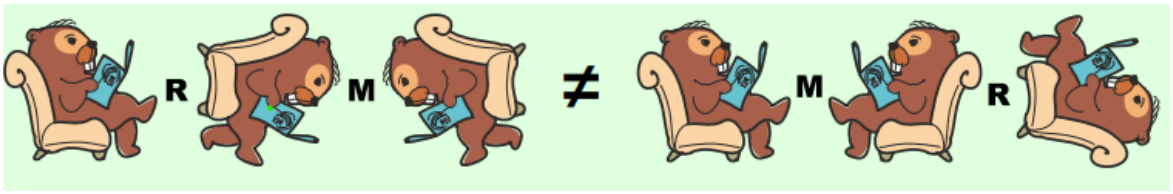


Atbildes skaidrojums

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 secība. 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	Dienu 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 rijīgā algoritma veids. Rijīgie 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.

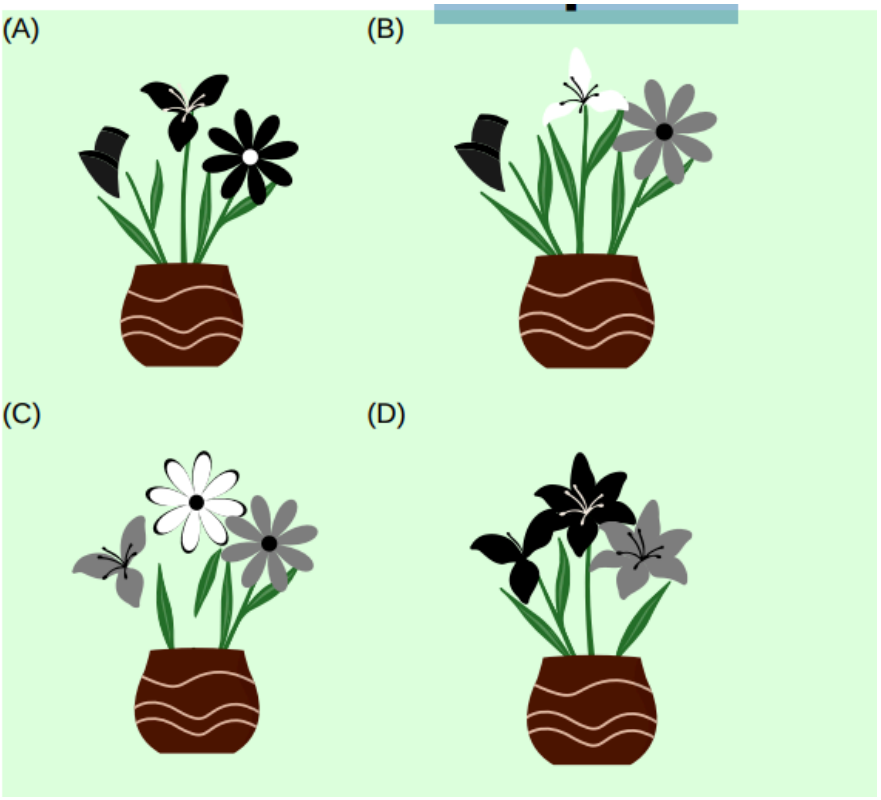
Emīlija vienmēr savās vāzēs tur vienādu puķu skaitu. Viņa arī nodrošina, ka katrā vāzē puķes ir sakārtotas pēc viena no trim noteikumiem: vienā vāzē ir vai nu viena veida vai tās pašas krāsas puķes, vai arī puķes ar vienādu lapu skaitu. Visas vāzes ir jāsakārto, izmantojot vienu un to pašu no aprakstītajiem 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

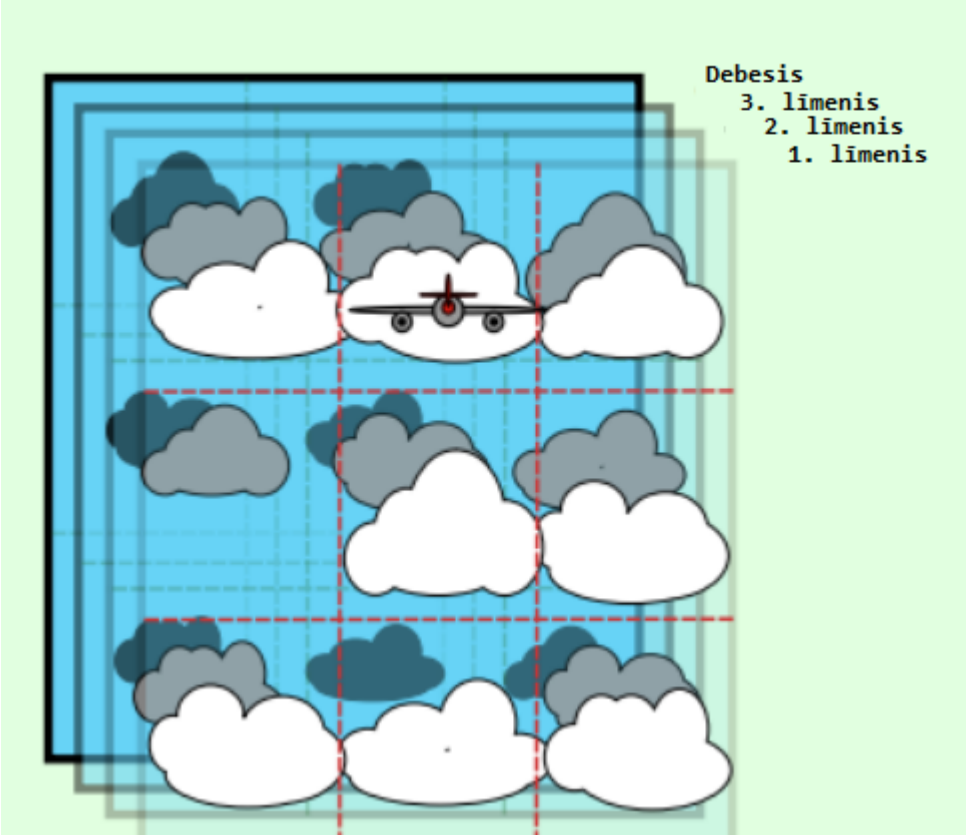
Šajā uzdevumā tiek attīstīta spēja izveidot un salīdzināt objektu (puķu) modeļus (attēlus), identificējot būtiskās īpašības un atribūtus (ko attēlo puķu veidi, krāsas un lapu skaits, kas vairākām puķēm ir kopīgs).

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.

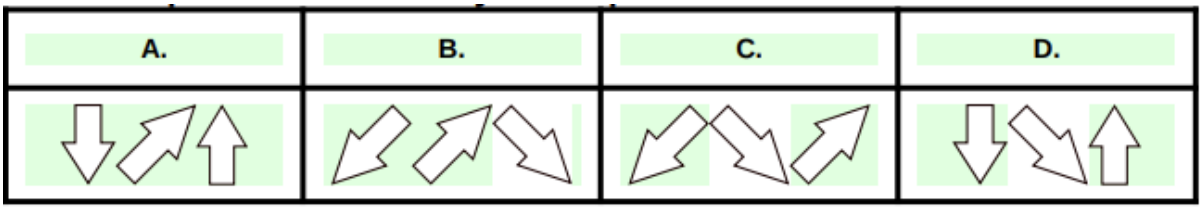


Tieši šobrīd lidmašīna gatavojas ielidot 3x3x3 režģī. Ja tā nepagriezīsies, tā ietrieksies baltajā mākonī augšējās rindas vidū.

Jautājums / Izaicinājums

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

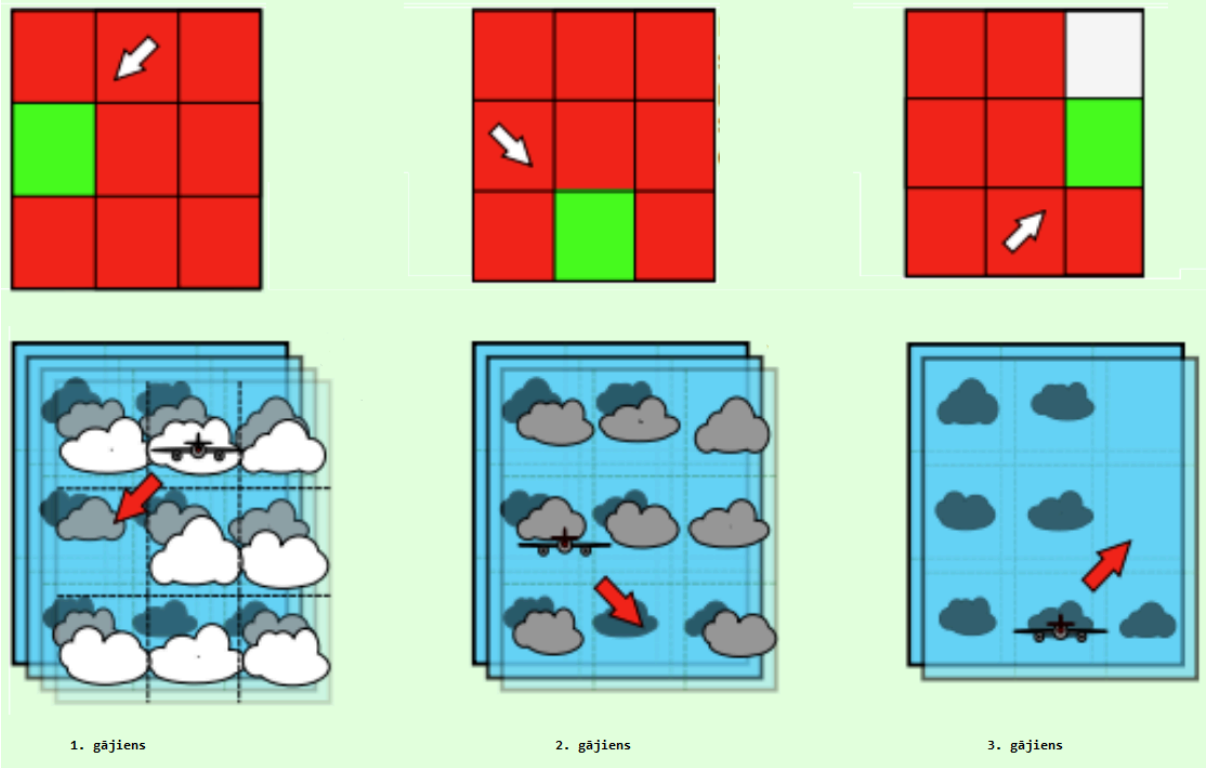
Atbilžu varianti / Interaktivitātes apraksts



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
Tiny	3
Blacky	7
Fatboy	10
Colorina	5
Lazyboy	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 konteinera 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 pamatkoncepcija. 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.

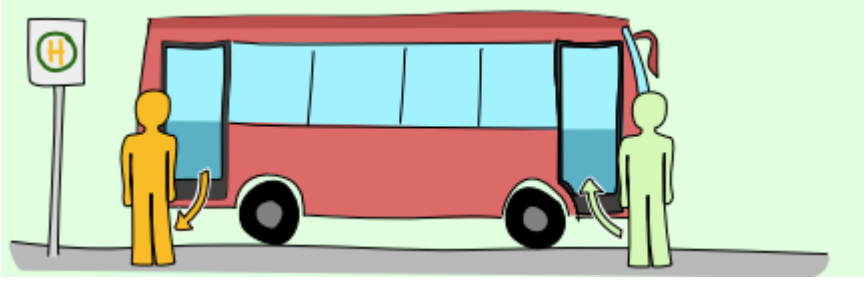
Šī ir skaitļošanas domāšana.

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?

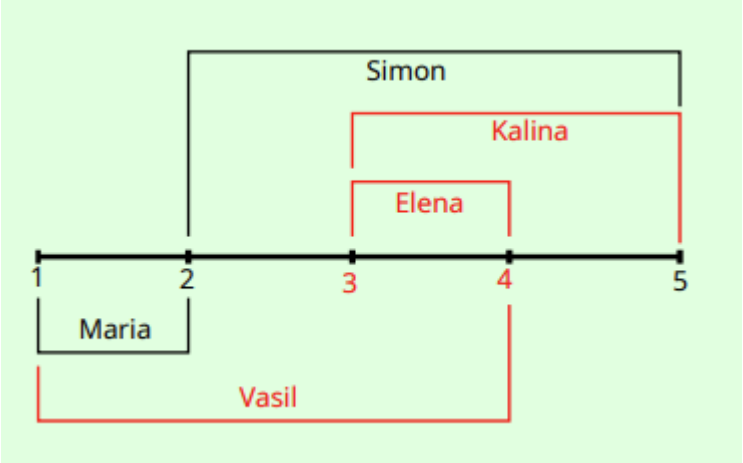
Atbilžu varianti / Interaktivitātes apraksts

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 nesasniegs savas vēlamās pieturas. Vasils iekāpj 1. pieturā un vēlas izkāpt 4. pieturā, bet autobuss tur neapstājas. Tādējādi Vasils, Jeļena un Kalina nesasniegs savas vēlamās pieturas.



Interaktīvs jautājums/izaicinājums

Atzīmējiet visus pasažierus, kuri var nokļūt savā vēlamajā pieturā?

Atbilde:

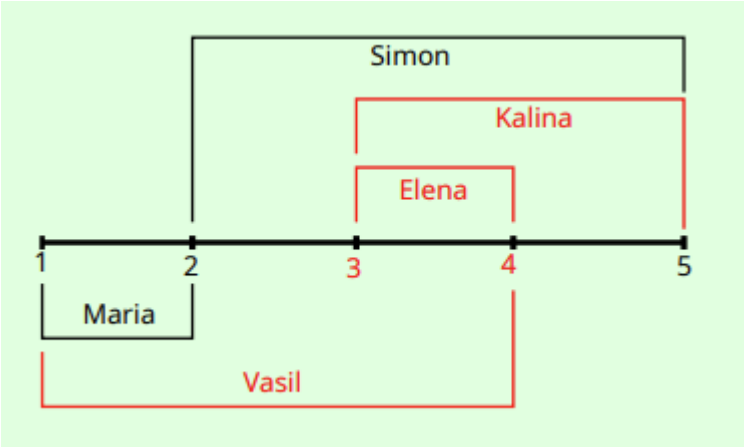
Marija un Saimons

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 nesasniegs savas vēlamās pieturas. Vasils iekāpj 1. pieturā un vēlas izkāpt 4. pieturā, bet autobuss tur neapstājas. Tādējādi Vasils, Jeļena un Kalina nesasniegs 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. Šīs 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

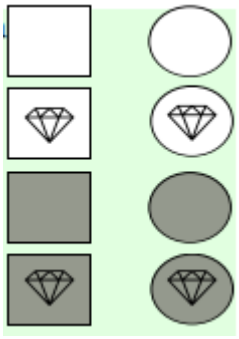
Visām izmaiņām, kas notiek laika gaitā, ir jāseko līdzi. Datordomāšanā tā ir tipiska notikumu izsekošana ciklā un izmaiņu veikšana sarakstos atkarībā no loģisko nosacījumu izpildes. Identificējamais modelis ir tas, vai pieturā ir gan pasažieri, kas vēlas iekāpt, GAN izkāpt, kā arī triviāli sākuma un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

Trešais

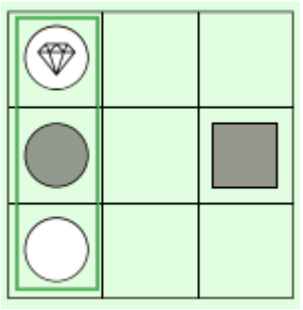
Brazīlija

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.

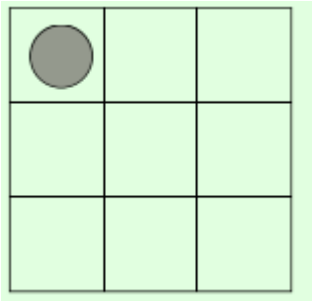


Lai uzvarētu, spēlētājam jāaizpilda laukuma rinda, kolonna vai diagonāle, kur trim figūriņām ir vismaz viena kopīga iepriekš aprakstītā īpašība. Piemēram:

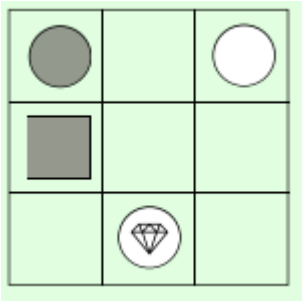


Iepriekš redzamajā spēles laukumā viens spēlētājs ir uzvarējis, jo novietojis trešo figūriņu pirmajā kolonnā, iegūstot trīs figūriņas ar vienādu raksturlielumu vienā līnijā: apļa formas figūriņas vienā kolonnā.

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.



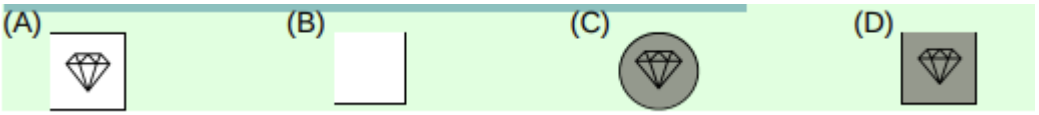
Viņi turpināja pārmaiņus, līdz spēles laukumā iestājās šāda situācijai:



Jautājums / Izaicinājums

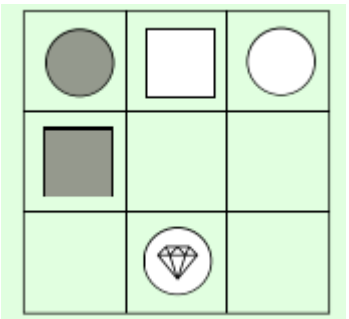
Kuru no tālāk norādītajām figūrām Anna var dot Bento novietot uz galda, lai nodrošinātu, ka viņš NEuzvar savā gājienā?

Atbilžu varianti / Interaktivitātes apraksts

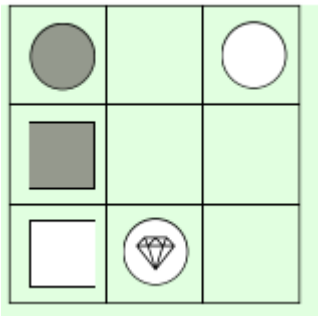


Atbildes skaidrojums

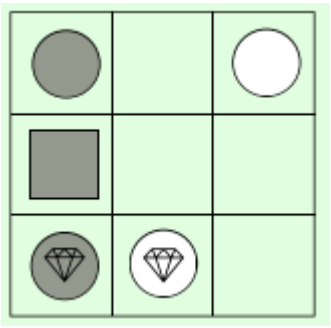
Pareizais variants ir (A).
Novietojot figūru no (A) iespējas, tiek nodrošināts, ka Bento nevarēs novietot trīs figūras ar kopīgu īpašību, tādējādi neļaujot viņam uzvarēt. (B) iespēja ir nepareiza. Ir divas iespējamās vietas, kur Bento varētu novietot šo figūru un uzvarēt:
- Viņš var novietot trīs figūras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



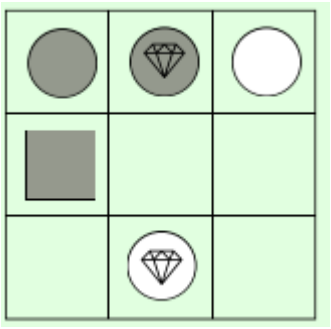
- Viņš var novietot trīs figūras bez romba pirmajā kolonnā.



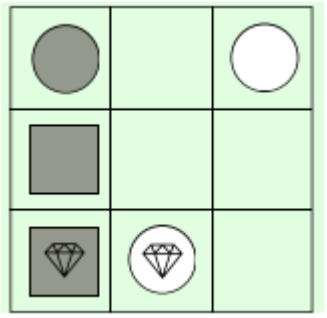
(C) iespēja ir nepareiza. Ir divas iespējamās vietas, kur Bento varētu novietot šo figūru un uzvarēt:
- Viņš var novietot trīs pelēkas figūras pirmajā kolonnā.



- 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 gājiens 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.

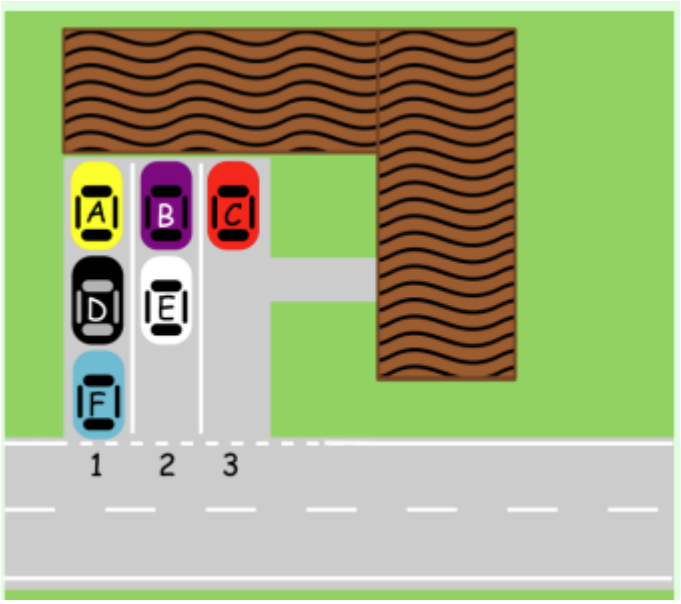
Šī ir skaitļošanas domāšana.

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).

Stāvvieta

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.

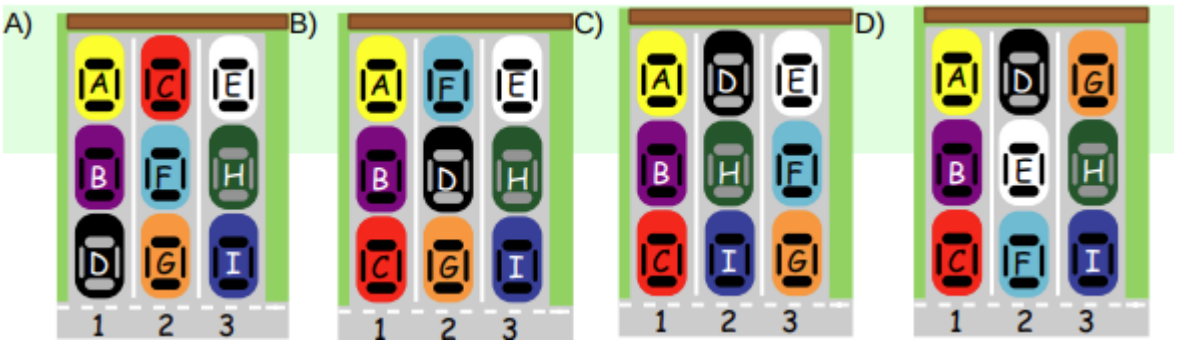
Viesi ierodas šādā secībā: Anna, Bobs, Cilda, Deivids, Elena, Frenks, Georgs, Heilija, Ivans.

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.

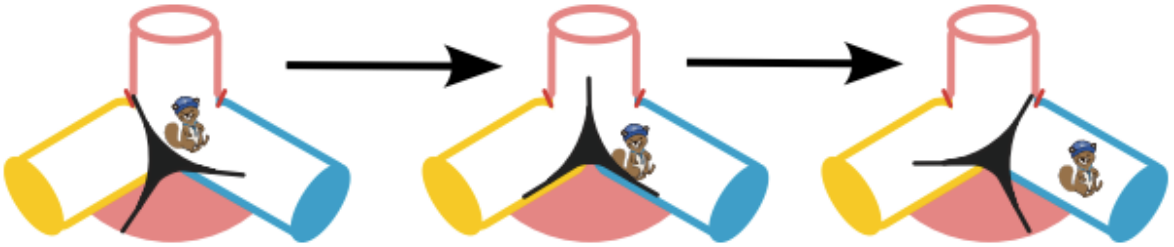
Šī ir skaitļošanas domāšana

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.

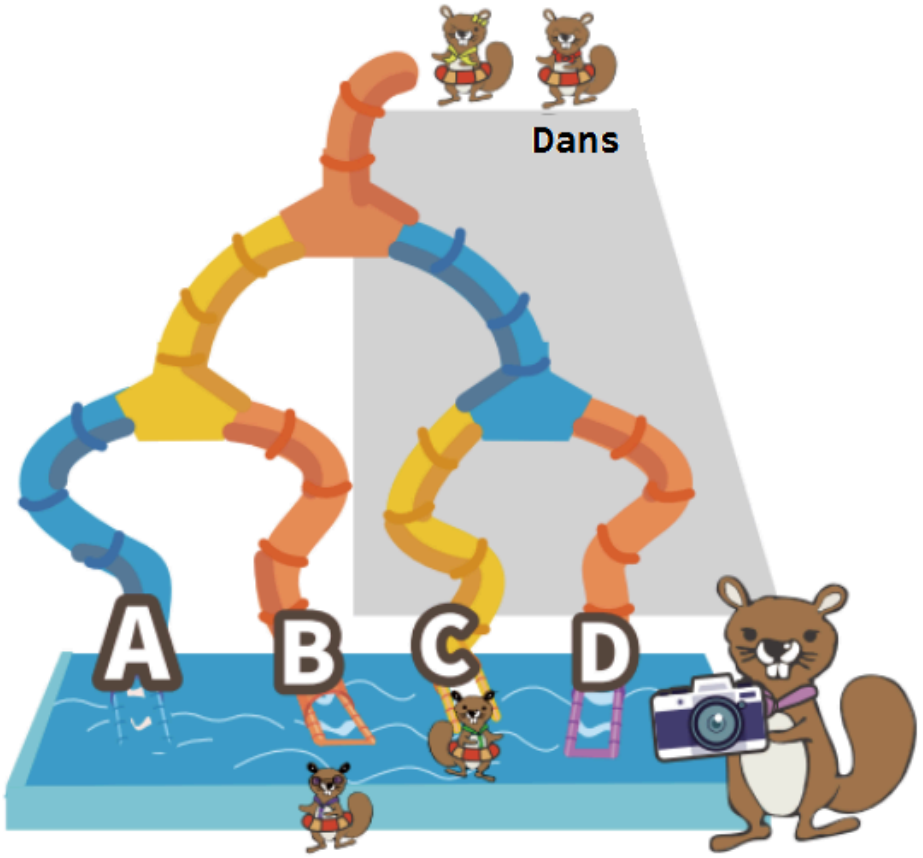
Aizraujošais nobrauciens

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 bebrs izslīd cauri atzarojumam, izejas virziens mainās.



Kā redzams nākamajā attēlā, mazais bebrs 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 nobraucienu gaida vēl viens bebrs un Dans būs nākamais pēc tā.



Jautājums / Izaicinājums

No kura slidkalniņa mazais bebrs Dans iekritīs baseinā?

Atbilžu varianti / Interaktivitātes apraksts

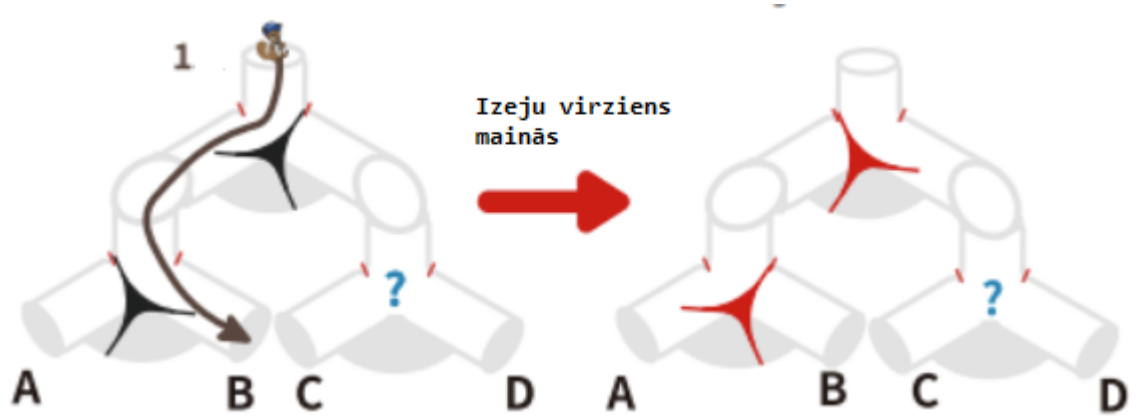
A. No A slidkalniņa
B. No B slidkalniņa
C. No C slidkalniņa
D. No D slidkalniņa

Atbildes skaidrojums

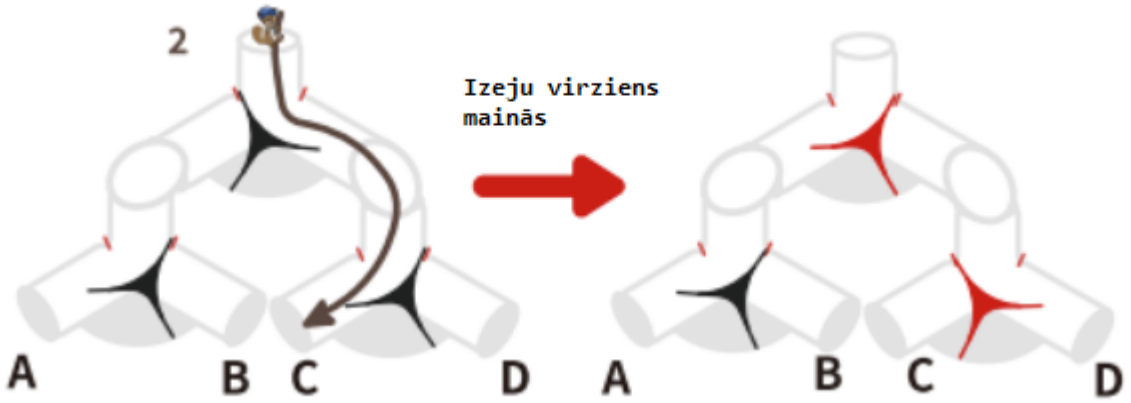
Atbilde ir D.

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

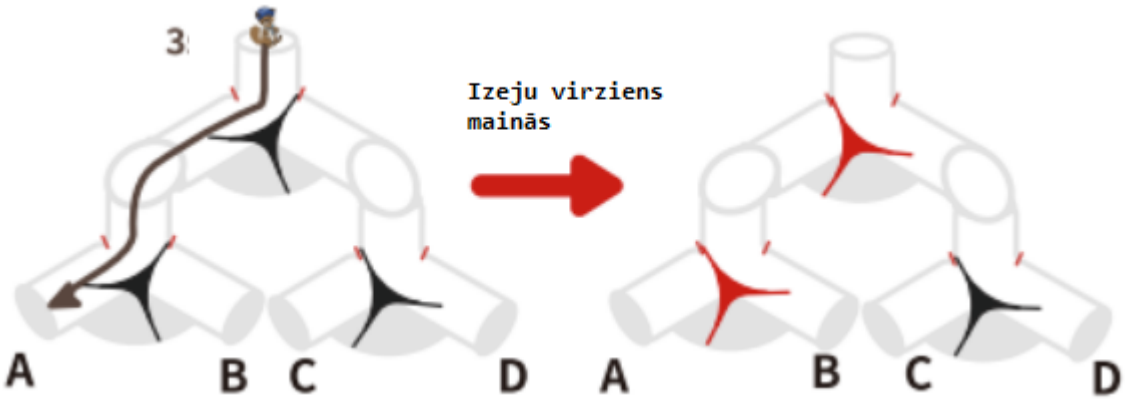
Kā redzams zemāk esošajā attēlā, pirmais bebrs iekrīt baseinā no B slidkalniņa; Mēs varam secināt, ka bebrs pirmajā atzarojumā iet pa kreisi un otrajā pa labi. Pēc tam šo divu atzarojumu izejas virziens mainās: pirmais atzarojums tiek atvērts pa labi, bet otrais (apakšējais kreisais) atzarojums - pa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



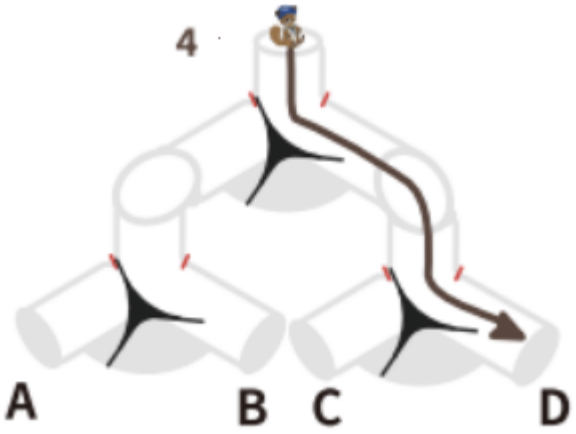
Līdzīgi, kā parādīts attēlā zemāk, nākamais bebrs iekrīt baseinā noi C slidkalniņa; mēs varam secināt, ka apakšējā labā atzarojuma izejas virziens bija pa kreisi pirms bebra izslīdēšanas un pēc tam pagriežas pa labi.



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, bebrs tam slīdēs cauri pa kreiso slidkalniņu un mehānisms pārslēgsies uz labo pusi; pretējā gadījumā bebrs slīdēs cauri pa labi un mehānisms pārslēgsies uz kreiso pusi.”
Kad bebrs izslīd cauri katram atzarojuma mehānismam, tas pats process atkārtojas. Bebrs 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 bebrs 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 trigeru digitālajās shēmās — pamatvienībai, kas glabā vienu datu bitu un pārslēdzas starp diviem stāvokļiem. Katrs atzarojums uzvedas kā trigeris, mainot savu stāvokli katru reizi, kad tas tiek izmantots.

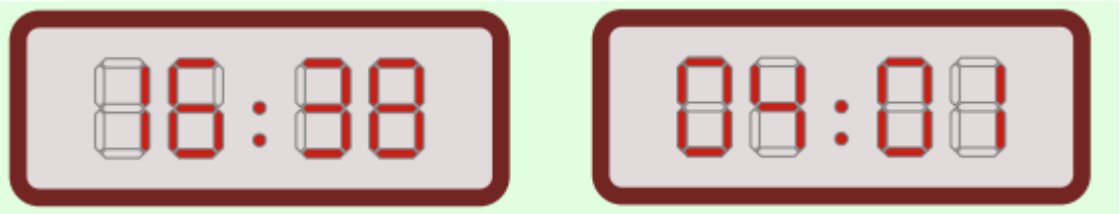
Lai noskaidrotu, kur bebrs nonāks, mēs soli pa solim simulējam katru darbību. Tas ir līdzīgi koda izsekošanai programmēšanā, kur programmas loģika tiek sekota rindiņai pa rindiņai, lai redzētu, kādu rezultātu tā rada.

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 bebrs 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.

Pulksteņa segments

Čehija

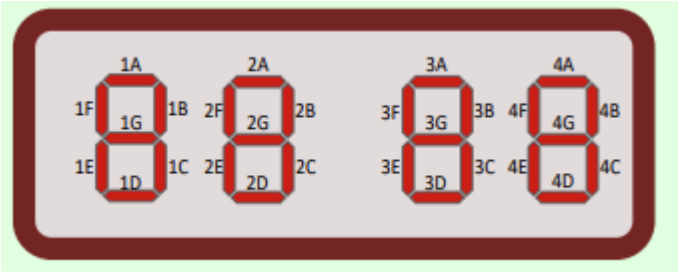


Standarta digitālais pulkstenis, kura rādījums mainās ik minūti, rāda laiku, izmantojot četrus ciparus. Katru ciparu attēlo septiņi gaismas segmenti. Lūk, kā, izmantojot segmentus, var attēlot skaitļus no 0 līdz 9:

Segmenti nolietojas katru reizi, kad tie tiek aktivizēti (pārslēgti no izslēgta stāvokļa uz ieslēgtu). Pēc kāda laika kā pirmais būs jānomaina segments, kas tiek aktivizēts visvairāk reižu.

Jautājums / Izaicinājums

Kurš no 28 pulksteņa segmentiem ir jānomaina vispirms? Norādiet segmenta kodu (cipars, kam seko burts, skatīt attēlu).



Interaktivitātes apraksts:

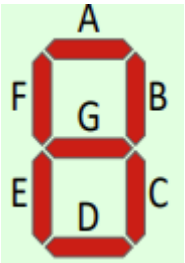
Kurš no 28 segmentiem ir jānomaina vispirms? Atzīmējiet to, noklikšķinot uz tā.

Katrs no 28 interaktīvajiem segmentiem tiek iezīmēts pēc noklikšķināšanas uz tā. Vienlaikus var atzīmēt tikai vienu segmentu.

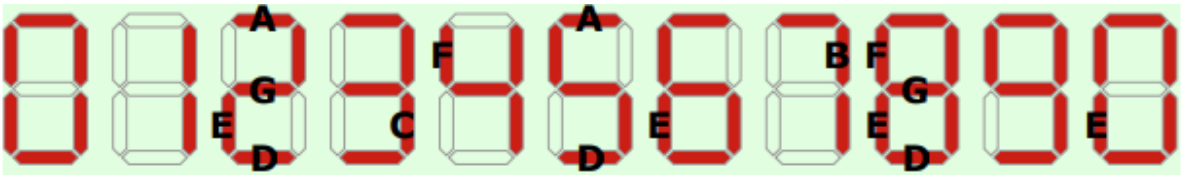
Atbildes skaidrojums

Pareizā atbilde ir: 4E (interaktīvā versija): 

Katrs cipars displejā apzīmē atšķirīgu laika periodu: desmitus stundu, stundas, desmitus minūšu un minūtes. Cipars visvairāk pa labi, t. i., ceturtais cipars jeb cipars ar kodu 4, rāda minūšu izmaiņas. Pirms jebkura cita cipara izmaiņām mainīsies visi ceturtā cipara segmenti. Visvairāk nolietotais segments būs ceturtajā ciparā, un tā segmenta kods ir E.



Mēs apskatīsim visus septiņus segmentus ceturtajā ciparā un noskaidrosim, cik reižu tie tiks aktivizēti (ieslēgti), pirms tajā tiks nomainīti visi cipari no 0 līdz 0.



Attēlā redzama šāda pārbaude. Aktivizētais (tikko ieslēgtais) segments ir segments, kas iedegas uz pašreizējā cipara, bet nedega uz tuvākā cipara pa kreisi. Katra segmenta aktivizācija ir atzīmēta ar šī segmenta burtu. Aktivizāciju skaits ir burtu skaits visā attēlā. Tabulā ir apkopota šī pārbaude. Centrālajā kolonnā ir parādīti cipari, uz kuriem segments tika aktivizēts. Labajā kolonnā ir uzskaitīti šie cipari.

Segments	Cipars, kas izraisa segmenta aktivizēšanu	Aktivizēšanas reižu skaits
A	2, 5	2
B	7	1
C	3	1
D	2, 5, 8	3

E	2, 6, 8, 0	4
F	4, 8	2
G	2, 8	2

Visvairāk mainītais segments ir E uz ceturtā cipara, un tā kods ir 4E.

Šī ir informātika

Šajā uzdevumā laika attēlošanai tiek izmantots septiņu segmentu displejs. Septiņu segmentu displejs ir izplatīts tehnoloģiskajos produktos. Tas attēlo skaitļus no 0 līdz 9, izmantojot septiņas gaismas diodes. Turklāt šis uzdevums ietver skaitīšanas sistēmas koncepciju. Skaitīšanas sistēmām ir liela nozīme datorzinātnēs. Visbiežāk izmantotā skaitīšanas sistēma ir decimālsistēma, savukārt datori datu attēlošanai izmanto bināro sistēmu. Tomēr mūsu ikdienas dzīvē pastāv arī citas skaitīšanas sistēmas. Piemēram, standarta pulkstenī minūtes tiek attēlotas sešdesmitnieku (ar bāzi 60) sistēmā, bet stundas - divdesmitčetrinieku (ar bāzi 24) sistēmā.

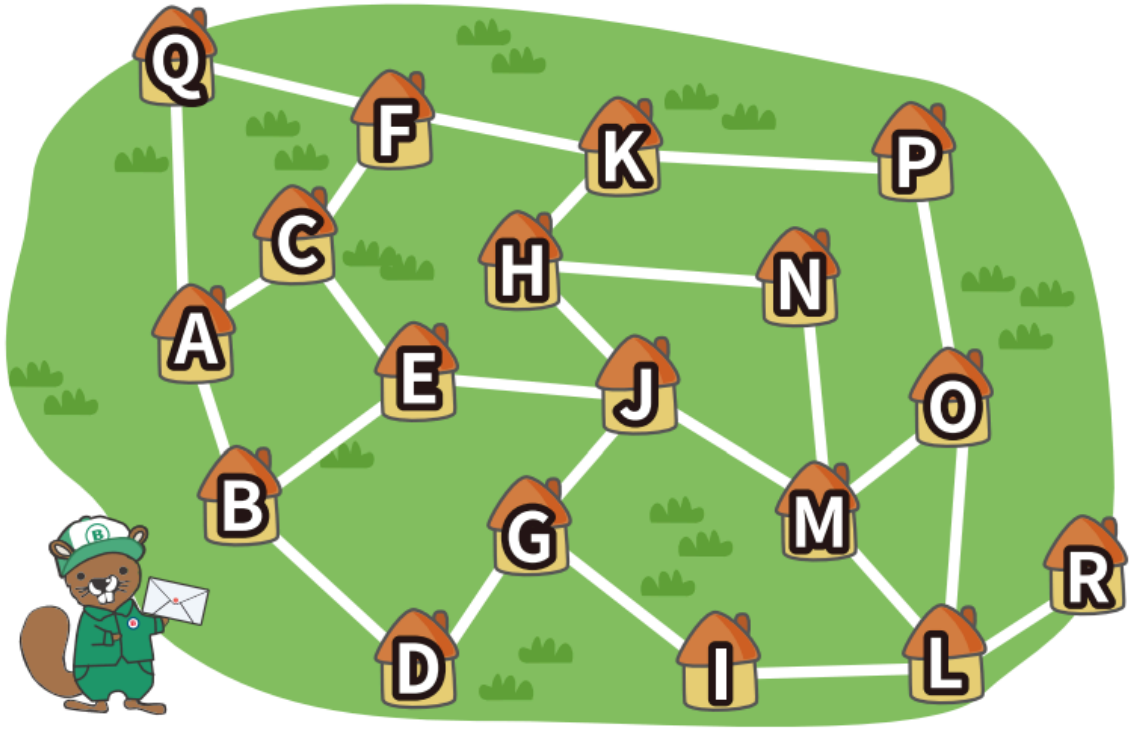
Šī ir skaitļošanas domāšana

Šis uzdevums paģēr, lai skolēni izmantotu loģisko spriešanu un veselo saprātu, lai ieraudzītu, ka standarta digitālā pulksteņa pēdējais cipars mainās visbiežāk. Tāpēc uzdevumu var sašaurināt līdz segmenta izmaiņu analīzei labajā malā esošajā ciparā. Pēc tam skolēniem jāanalizē izmaiņu skaits katrā septiņu segmentu displeja segmentā pilna izmaiņu cikla laikā. Šādas izmaiņas tiek attēlotas kā displeja segmentu pāreja no izslēgta stāvokļa uz ieslēgtu stāvokli. Šim procesam ir nepieciešama spēja atpazīt modeļus.

Bebrusalas biļetens

Taiwāna

Bebrusalā ir 18 ciemati, kā parādīts attēlā zemāk. Katrā ciematā ir daudz pastnieku - ikreiz, kad no ciemata ir jānosūta ziņojums vai tajā tiek saņemts jauns ziņojums, ciemata pastnieki nākamajā dienā nogādās ziņojumu visiem saistītajiem kaimiņu ciematiem.



Piemēram, ja no ciemata A nosūta ziņojumu, ziņojuma nonākšanai ciematos B, C un Q nepieciešama 1 diena. Lai ziņojums sasniegtu ciematus D, E un F, nepieciešamas 2 dienas, un tā tālāk, līdz visi ciemati ir saņēmuši ziņojumu.

Jautājums / Izaicinājums

Ja jauns ziņojums tiek sūtīts no ciemata J, cik dienas nepieciešamas, lai tas sasniegtu visus ciematus?

Atbilžu varianti / Interaktivitātes apraksts

Vesels skaitlis no [1, 18]

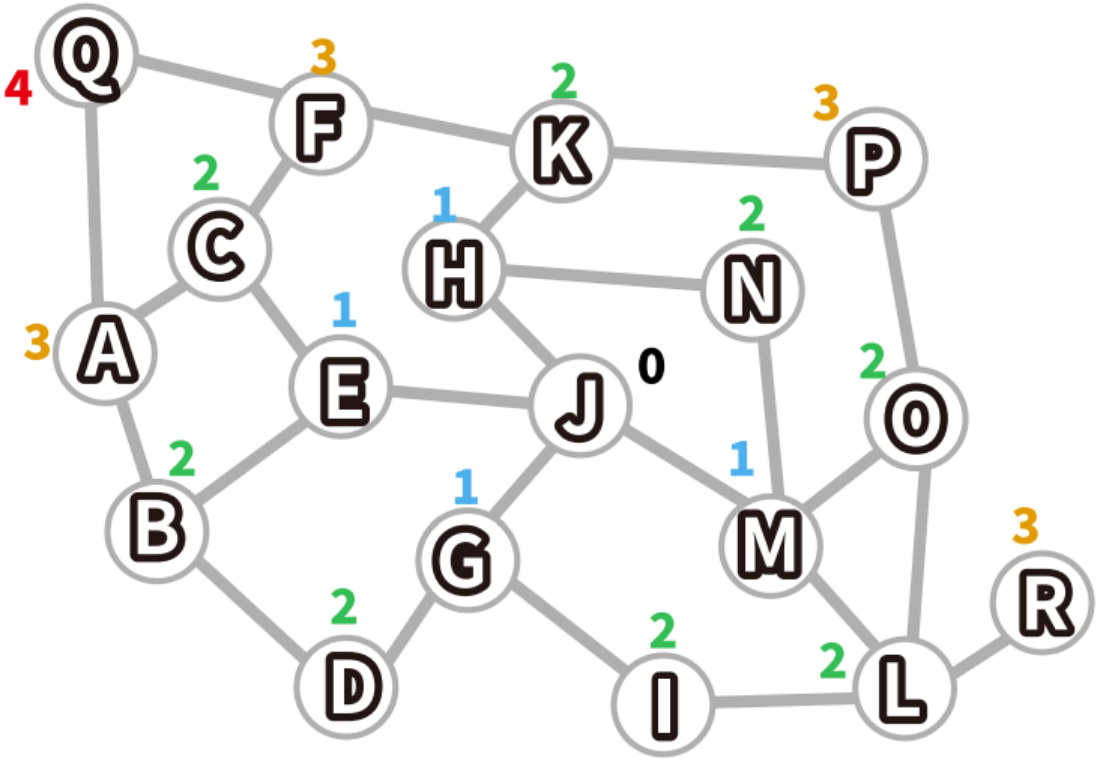
Atbildes skaidrojums

Sākot no ciemata J, mēs varam izsekot, cik dienu nepieciešams, lai ziņojums sasniegtu katru ciematu, un uzzināt atbildi:

1. diena: Ciemati E, G, H un M saņem ziņojumu.
2. diena: Ciemati B, C, D, I, K, N, O un L saņem ziņojumu, proti: ciemats B un C to saņem no E. Ciemati D un I to saņem no G. Ciemati K un N to saņem no H. Ciemati N, O un L to saņem no M.

3. diena: Līdzīgi ziņojumu saņem ciemati A, F, P un R. Šajā brīdī ziņojumu vēl nav saņēmis tikai ciemats Q.

4. diena: Ciemats Q saņem ziņojumu no F un A.



Tāpēc kopumā būs nepieciešamas četras dienas, lai nodrošinātu, ka visi ciemati ir saņēmuši ziņojumu no ciemata J.

Tā ir informātika

Šajā uzdevumā ziņojumu izplatīšanas metode ir līdzīga meklēšanas platumā (platkursija, BFS) algoritma veikšanai grafā, sistemātiski izpētot vai meklējot mezglus. Platkursijas algoritms sāk no konkrēta mezgla, vispirms apmeklējot tā blakus esošos mezglus, un pēc tam turpina apmeklēt vēl neapmeklētos blakus esošos mezglus tādā secībā, kādā tie tiek atrasti. Platkursijas pamatkoncepcija ir "paplašināšana slāni pa slānim", nodrošinot,

ka vispirms tiek izpētīti mezgli, kas atrodas tuvāk sākumam, pirms paplašināšanas uz attālākiem mezgliem.

Platkursijas praktiskos pielietojumos ietilpst meklētājprogrammas, kas pārmeklē tīmekļa lapas slāni pa slānim, izmantojot saites, sociālo mediju platformas, kas analizē, kā ziņojumi izplatās no viena lietotāja citiem, un sabiedrības veselības iestādes, kas izseko kontaktus un simulē infekcijas slimību izplatību. Saistīts jēdziens ir pludināšanas-aizpildīšanas algoritms, ko parasti izmanto, lai risinātu problēmas, kas saistītas ar savienotu apgabalu paplašināšanu vai izpēti. Piemēram, pieskaroties apgabalam krāsošanas lietotnē, algoritms aizpilda visu sadaļu ar krāsu. Līdzīgi tas tiek pielietots, kad tīrīšanas robots kartē un tīra visas savienotās telpas, sākot no sākotnējās atrašanās vietas. Iekšēji plūdu aizpildīšana parasti balstās uz meklēšanu platumā vai dziļumā (dziļkursija, DFS), lai sistemātiski un rūpīgi izpētītu visus blakus esošos savienotos mezglus.

Šī ir skaitļošanas domāšana

Lai atrisinātu šo uzdevumu, studentiem vispirms ir jāsaprot un jāpielieto noteikums par to, kā ziņojumi tiek nodoti no ciemata uz ciematu. Viņiem sistemātiski jāseko ziņojuma izplatībai dienu no dienas, kas ietver algoritmisku domāšanu. Šī procesa laikā studenti iesaistās arī simulācijā, jo viņi garīgi vai vizuāli modelē, kā ziņojums laika gaitā “plūst” tīklā. Viņiem ir jāatbrīvojas no nevajadzīgām detaļām un jākoncentrējas uz galvenajiem elementiem: savienojumiem starp ciematiem un ziņojuma izplatīšanas laiku.

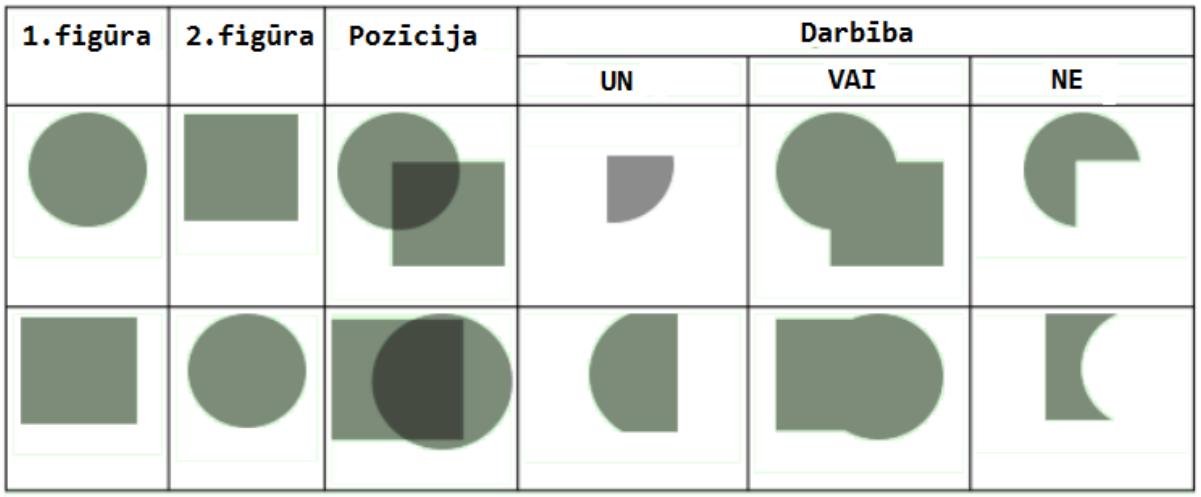
Informātikas atslēgvārdi un tīmekļa vietnes

Meklēšana plašumā, pludināšanas-aizpildīšanas algoritms

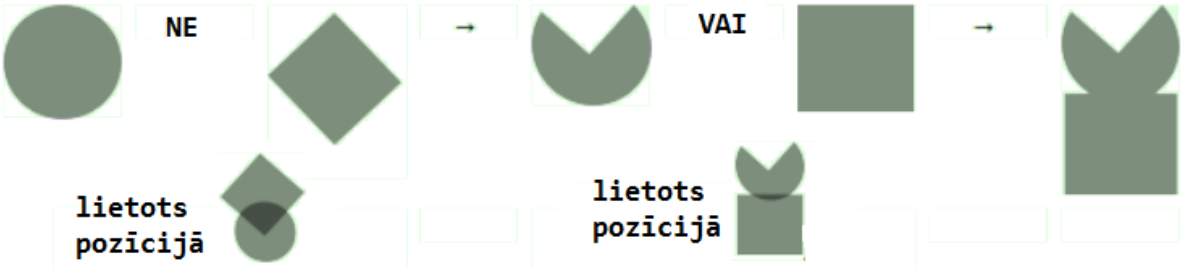
Būla figūras

Īrija

Būla darbības tiek izmantotas datorgrafikā. Šīs darbības ļauj no vienkāršākām figūrām veidot sarežģītākas. Zemāk ir sniegts trīs Būla pamatdarbību piemērs: UN (angliski AND), VAI (OR) un NE (NOT).

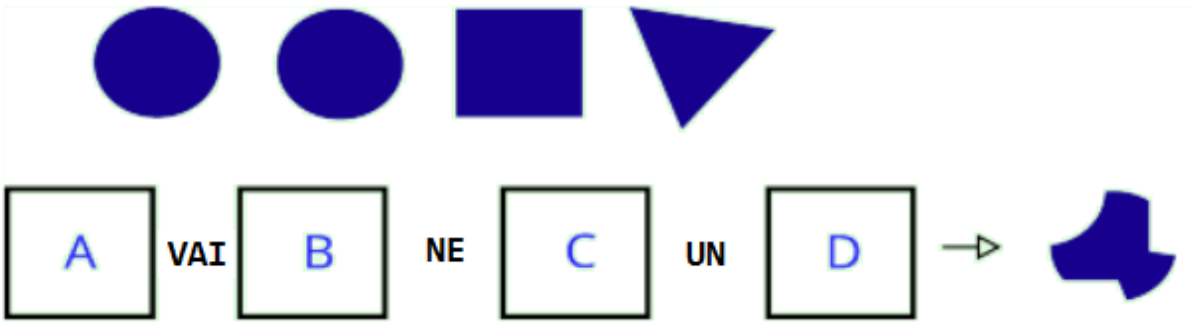


Mēs varam izmantot šīs darbības secīgi, lai uzzīmētu/izveidotu sarežģītākas figūras.

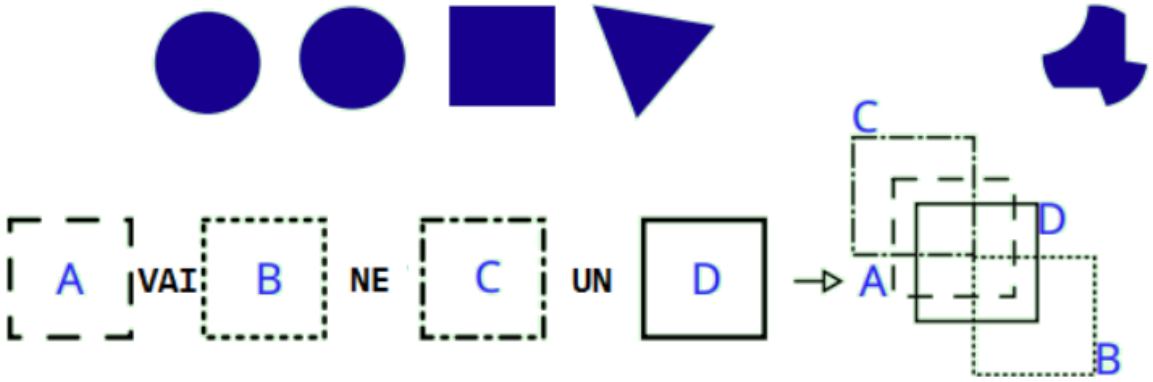


Jautājums / Izaicinājums

Jums ir dotas četras figūras un trīs Būla operācijas. Velciet un nometiet figūras pareizajās vietās tā, lai rezultāts pēc darbību izpildes būtu tāds, kāds parādīts aiz bultiņas.



Versija: Ja vēlaties sniegt skolēniem lielāku atbalstu pozīcijas noteikšanā, varat izmantot mērķa kvadrātus “pozīcija”:

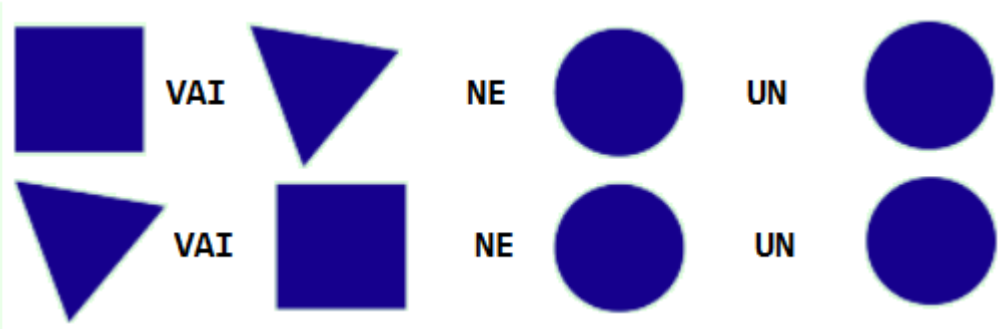


Atbilžu varianti / Interaktivitātes apraksts

Interaktivitātes apraksts: četras figūras jāvelk uz kvadrātiem, kas apzīmēti ar A, B, C, D. Ja tās atrodas tuvu atzīmētajiem kvadrātiem, tām vajadzētu “ielēkt” iekšā. Figūras var noņemt, atgriežot sākotnējā vietā (ja tās tiek pārvietotas ārpus atzīmētajiem kvadrātiem, tās atgriežas sākotnējā vietā).

Atbildes skaidrojums

Pareizā atbilde ir viena no šīm:



Lai atrisinātu šo uzdevumu, mums ir jāidentificē iegūtā attēla galvenās iezīmes.

Pirmā dotā darbība ir VAI, kas izveidos iegūtā attēla “ķermeni”, apvienojot divas figūras. Šeit jums jāizmanto kvadrāts un trīsstūris. Tā kā mēs izmantojam VAI, figūru secībai nav nozīmes. VAI rezultāts ir vienāds abos gadījumos.

Nākamā dotā darbība ir NE. Tas nozīmē, ka mēs noņemsim daļu no attēla. Mēs redzam, ka mums būs jāatņem kaut kas no mūsu figūras augšējā kreisā stūra. Tā kā tas ir apļveida, mums jāizmanto aplis, lai noņemtu daļu ar NE.

Pēdējā dotā operācija ir UN, kas saglabās visu, kas ir kopīgs abās figūrās. Rezultāta iegūšanai varam izmantot pēdējo apli.

Šī ir informātika

Būla darbības jau sen tiek izmantotas datorgrafikā. Tādas darbības kā VAI (divu figūru apvienojums), UN (divu figūru šķēlums) un NE (vienas figūras “atņemšana” no citas) ļauj no vienkāršākām figūrām izveidot sarežģītākas figūras. Šīs darbības parasti ir pieejamas gan bitkartes, gan vektoru zīmēšanas programmās. Šādas darbības, veiktspējas apsvērumu dēļ, labāk veikt ar vektoru attēlojumiem nekā ar bitkartēm. Tas, savukārt, ir novedis pie veselas algoritmu klases, slaucīšanas līnijas algoritmu, izmantošanas, lai efektīvi veiktu šīs darbības ar vektoru attēlojumiem. (IESPĒJA: Ja vēlaties sniegt skolēniem/skolotājiem vairāk infigūrācijas par terminu Būla loģika, tad tas attiecas uz ko tādu, kam var būt tikai viena no divām vērtībām: patiesa vai nepatiesa. Būla loģikai ir svarīga loma programmēšanā. Programmēšanā Būla apgalvojumi, piemēram, tiek izmantoti, lai izveidotu nosacījumus cikliem un zarošanās operatoriem: if punkti > 100: print(“Apsveicu”) Būla apgalvojums vienmēr ir patiess vai nepatiess: piemēram, “punkti > 100”, “Bobs ir vecāks par Alisi”, “Līst lietus”. Būla operatorus, piemēram, UN, VAI, NE, var izmantot, lai izveidotu sarežģītākus apgalvojumus vai izteiksmes: piemēram, “Līst lietus” UN “Māja ir dzeltena”. Šis apgalvojums ir patiess tikai tad, ja gan līst, gan māja ir dzeltena.

“Līst lietus” VAI “Māja ir dzeltena”. Šis apgalvojums ir patiess, ja līst lietus vai/un māja ir dzeltena.

NE “Māja ir dzeltena”. Šis apgalvojums ir patiess, ja māja NAV dzeltena.)

Šī ir skaitļošanas domāšana

Abstrakcija: Lai atrisinātu šo uzdevumu, skolēnam ir jāaplūko rezultātā iegūstamā figūra un jāatrod līdzības un atšķirības, lai izlemtu, kuras figūras izmantot un kādā secībā.
Sadalīšana: Mums ir jāsadala uzdevums mazākos, nosakot figūras, kas jāizmanto katrā atsevišķā darbībā.

Puķu stādīšana

Slovākija

Robots puķu dobē rindā stāda pilnībā izaugušas puķes, pamatojoties uz norādēm. Tukšā vietā nav ne norādes, ne puķes. Robots puķu stādīšanai izmanto šādas instrukcijas:

0: Doties uz vietu, kas atzīmēta ar X.

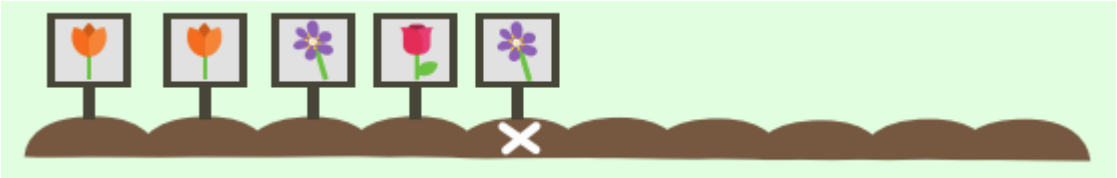
Atkārtot 1.–5. darbību, līdz vairs nav nevienas vietas ar norādi.

1: Ja šajā vietā ir norāde, iestādīt šajā vietā uz norādes attēloto puķi.

2: Iegaumēt tikko iestādīto puķi.

3: Noņemt norādi.

4: Dodieties pa labi, līdz sasniegta tukša vieta, pēc tam iestādīt tur puķi, ko iegaumējāt kā pēdējo.
5: Dodieties pa kreisi, līdz sasniegta vieta ar norādi vai līdz izkāpts ārpus puķu dobes.



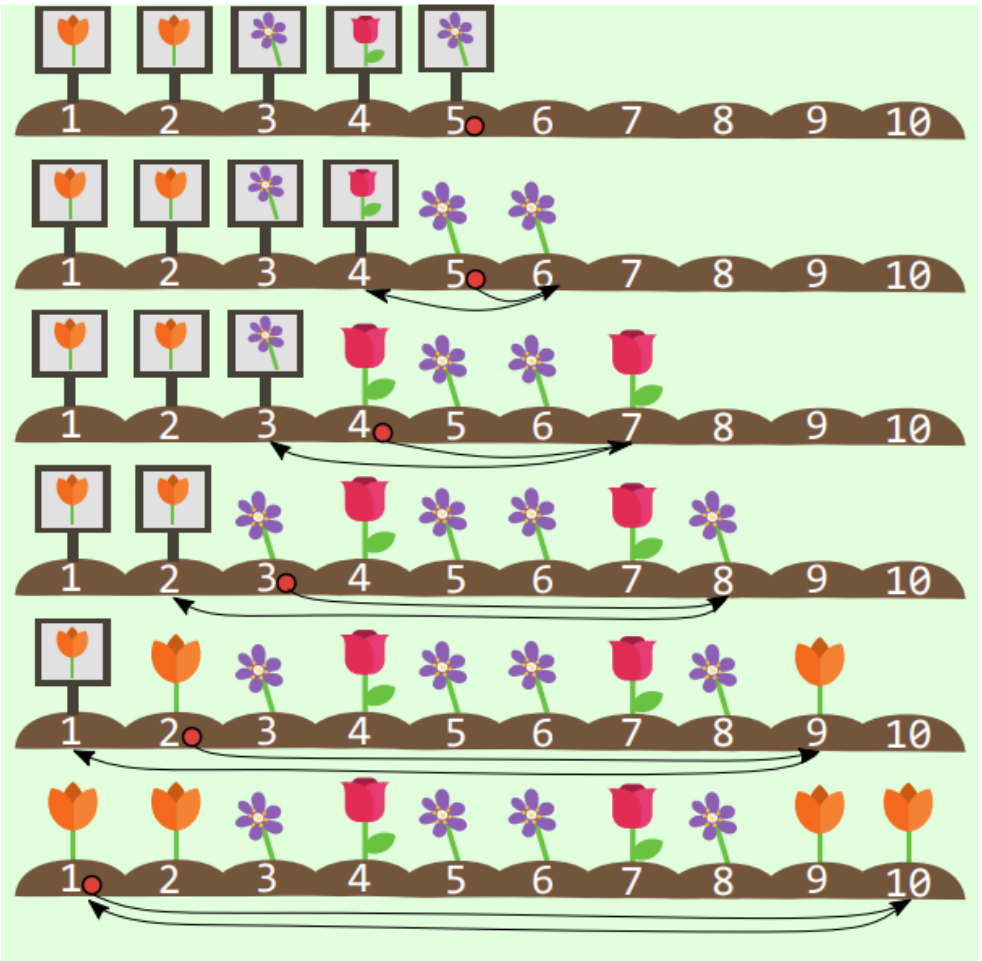
Jautājums / Izaicinājums

Kā puķu dobe izskatīsies pēc stādīšanas?

Atbilžu varianti / Interaktivitātes apraksts

Atbildes skaidrojums

Pareizā atbilde ir A.
Sanumurēsim puķu vietas. Robots sāk no pēdējās apzīmētās vietas, kas ir 5. vieta. Tur tas iestāda vijolīti, iegaumē to un iet pa labi. Tieši blakus ir tukša vieta (6. vieta), tāpēc robots iestāda pēdējo iegaumēto puķi (vijolīti) un tad iet pa kreisi, līdz atrod pirmo vietu ar norādi. Tā ir 4. vieta. Tur tas iestāda rozi un iegaumē to. Tad robots dodas pa labi uz pirmo brīvo vietu, kas ir 7. vieta, un iestāda rozi. Tālāk tas dodas pa kreisi, līdz atrod vietu ar norādi – tā ir 3. vieta. Robots tur iestāda vijolīti, iegaumē to un iet pa labi uz pirmo brīvo vietu (8. vieta), kur iestāda vēl vienu vijolīti. Gan 2. un 9. vietā, gan 1. un 10. vietā tas iestādīs tulpes. Var redzēt, ka robots tukšajās vietās stāda puķes pretējā secībā. Attēlā sarkanais punkts apzīmē robota pozīciju, un bultiņas parāda, kā tas pārvietojās.



Šī ir informātika

Šajā uzdevumā Tjūringa mašīnas koncepcija ir paslēpta aiz robota instrukcijām. Tjūringa mašīna ir mašīnas jēdziens, kas izpilda algoritmu un šajā uzdevumā tai ir viena galviņa (robots), kas pārvietojas pa lenti (puķu dobi) ar šūnām (punktiem) un var lasīt (norādes) un rakstīt (noņemt norādi un iestādīt puķes) šajos punktos. Tjūringa mašīnas atmiņa ir lente, un mašīnai var būt vairāk nekā viena lente. Šajā uzdevumā robots iegaumē, kura puķe tika iestādīta kā pēdējā, lai šo informāciju varētu ierakstīt citā lentē.

Šī ir skaitļošanas domāšana

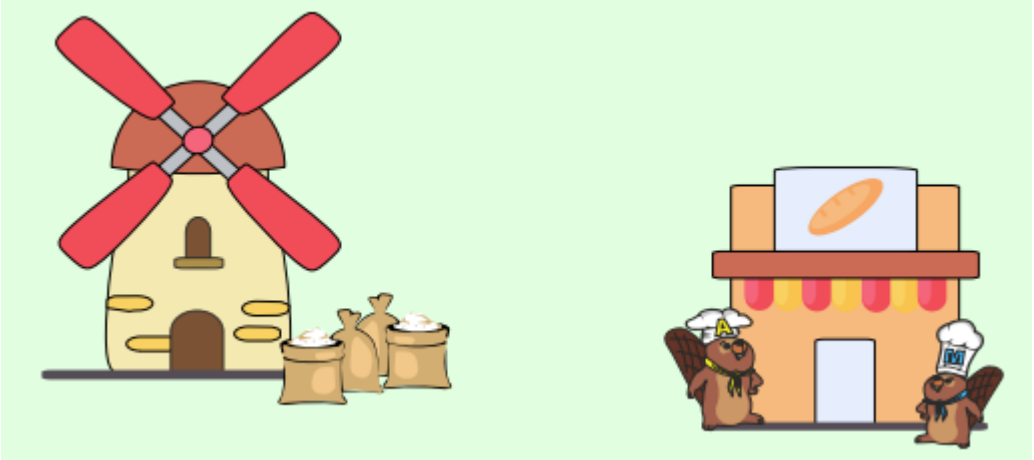
Šajā uzdevumā ir nepieciešama algoritmiskā domāšana. Jums ir jāsaprot un jāseko piedāvātajam algoritmam, kas ir uzrakstīts kā instrukciju kopums, un jums ir jāizpilda šīs instrukcijas noteiktiem ievaddatiem. Šajā uzdevumā ievaddati ir norāžu virkne, un rezultāts (izvaddati) ir iestādīto puķu virkne.

Milti

Taivāna

Divi bebri - Alberts un Mario veido leģendāru miltu pārvadāšanas komandu.
Alberts katrā braucienā pārvadā 13 kg miltu, un ceļš no maiznīcas līdz dzirnavām un atpakaļ aizņem vienu stundu. Mario pārvadā tikai 5 kg, bet katru turp un atpakaļ braucienu veic 30 minūtēs. Viņi abi nevar doties braucienā vienlaikus, jo vismaz vienam jāpaliek maiznīcā apkalpot klientus.

Tāpat kā visiem čakliem strādniekiem, viņiem kaut kad ir arī jāatpūšas. Pēc trim secīgiem braucieniem katram bebram jāpavada vismaz 30 minūtes, pirms viņš dodas nākamajā braucienā. Atpūtas laikā viņi var apkalpot klientus.



Jautājums / Izaicinājums

Alberts un Mario vēlas pārvest pēc iespējas vairāk miltu astoņu stundu laikā.
Kurš no dotajiem apgalvojumiem ir pareizs?

Atbilžu varianti / Interaktivitātes apraksts

A. Albertam jādodas braucienā pirmajam.

B. Mario jādodas braucienā pirmajam.

C. Mario jādodas pēdējā no braucieniem.

D. Alberts nedrīkst būt pēdējā no braucieniem.
E. Mario jādodas tikai vienā braucienā.

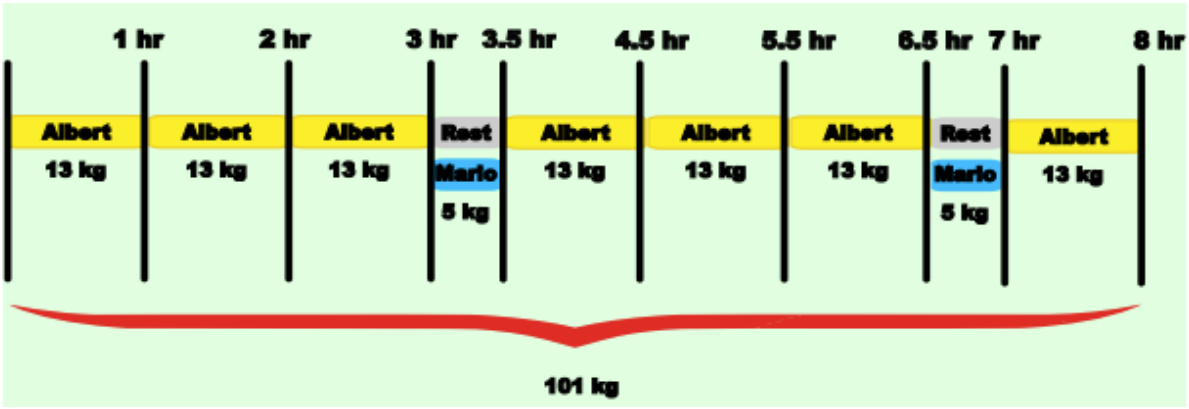
Atbildes skaidrojums

Pareizā atbilde ir A.
Galvenie fakti:
• Alberts: 13 kg vienā reizē, 1 stunda vienā reizē (13 kg/stundā)
• Mario: 5 kg vienā reizē, 30 minūtes vienā reizē (10 kg/stundā)

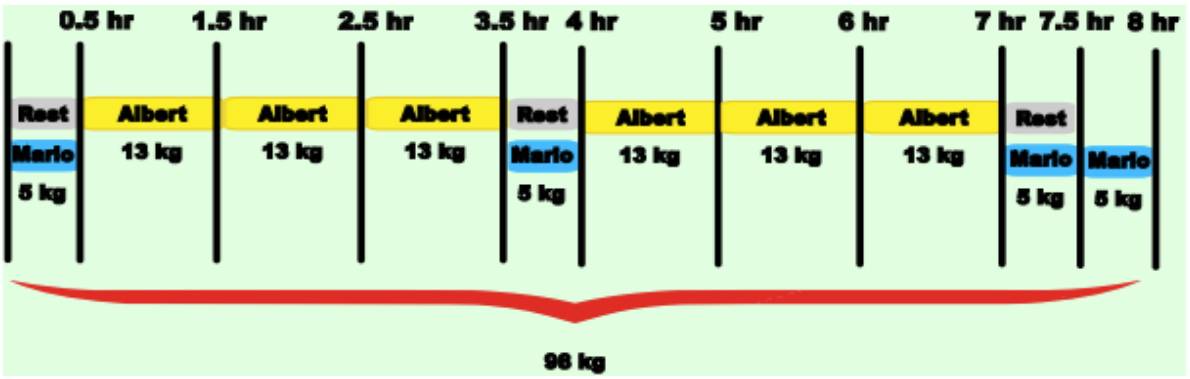
• Vienam bebram jābūt maiznīcā
• Katrs bebrs var izpildīt līdz trim braucieniem pēc kārtas
• Kamēr viens bebrs atpūšas maiznīcā, otrs turpina doties braucienos

Tā kā Alberts var piegādāt 13 kg/stundā, salīdzinot ar Mario 10 kg/stundā, Alberts ir efektīvāks. Tomēr mums ir jāizmanto Mario, kad Albertam ir jāatpūšas. Ja bebrs Alberts (A) pārvadā 3 reizes un Mario (M) vienu reizi, mēs iegūstam maksimālo miltu daudzumu 3,5 stundu laikā:

Tagad Alberts ir atpūties, tāpēc mēs varam atkārtot šo ciklu, līdz laiks ir beidzies:



Ja Mario sāk pirmais, tad pēc diviem (M, A, A) cikliem paliek tikai viena stunda. Tā kā Albertam tajā laikā ir jāatpūšas, Mario ir jāveic pēdējās divas kārtas. Tas noved pie mazāk efektīva grafika salīdzinājumā ar iepriekšējo.



Alberts nevar braukt 8 reizes, jo tas prasa 8 stundas braucienos + 2 * 30 minūšu atpūtu = 9 stundas, kas ir vairāk nekā atļauts uzdevumā. Tātad risinājums, kurā Alberts brauc 7 reizes + Mario 2 reizes, ir optimāls.

Šī ir informātika

Bebru miltu transportēšanas problēma ir plānošanas un optimizācijas problēma, jo:

• Mums ir jāplāno braucieni, ievērojot atpūtas un aizņemtības ierobežojumus.
• Mēs vēlamies maksimāli palielināt piegādāto miltu kopējo daudzumu (optimizācijas mērķis).
• Mums ir efektīvi jāsadala laiks starp Albertu un Mario.

Plānošanas un optimizācijas problēmas rodas daudzos informātikas uzdevumos, tostarp instrukciju izpildē datoru procesoros, kur aritmētiskās loģikas vienības var izpildīt tikai vienu instrukciju vienlaikus (aizņemtības ierobežojumi), instrukcijām, kas lasa datus no atmiņas, ir jāgaida, kamēr dati pienāk (līdzīgi kā atpūtai), un optimāls grafiks var izpildīt programmu daudz ātrāk nekā tad, ja grafiks ir neoptimāls.

Šī ir skaitļošanas domāšana

Šī uzdevuma risināšanā ir ieguvumi no sarežģītas problēmas sadalīšanas apakšproblēmās (dekompozīcija), atziņas, ka apakšproblēmas risinājumus var atkārtot (atkārtošana un modeļu atpazīšana), apakšproblēmas risinājumu sakārtošanas, lai atrastu optimālu risinājumu, un abstrakcijas, kas attēlo optimālo transportējamo kg/stundā. Ir nepieciešama arī loģiska domāšana, lai izslēgtu tos variantus, kas neatbilst tādiem ierobežojumiem kā atpūtas periodi un noslogojums (vismaz viens bebrs maizes ceptuvē).