VISMA

Bebr[a]s '24

2024. gada 1. kārtas uzdevumi ar atbildēm

5.-6. klase

INFORMATĪVIE ATBALSTĪTĀJI

Izglītības un zinātnes ministrija

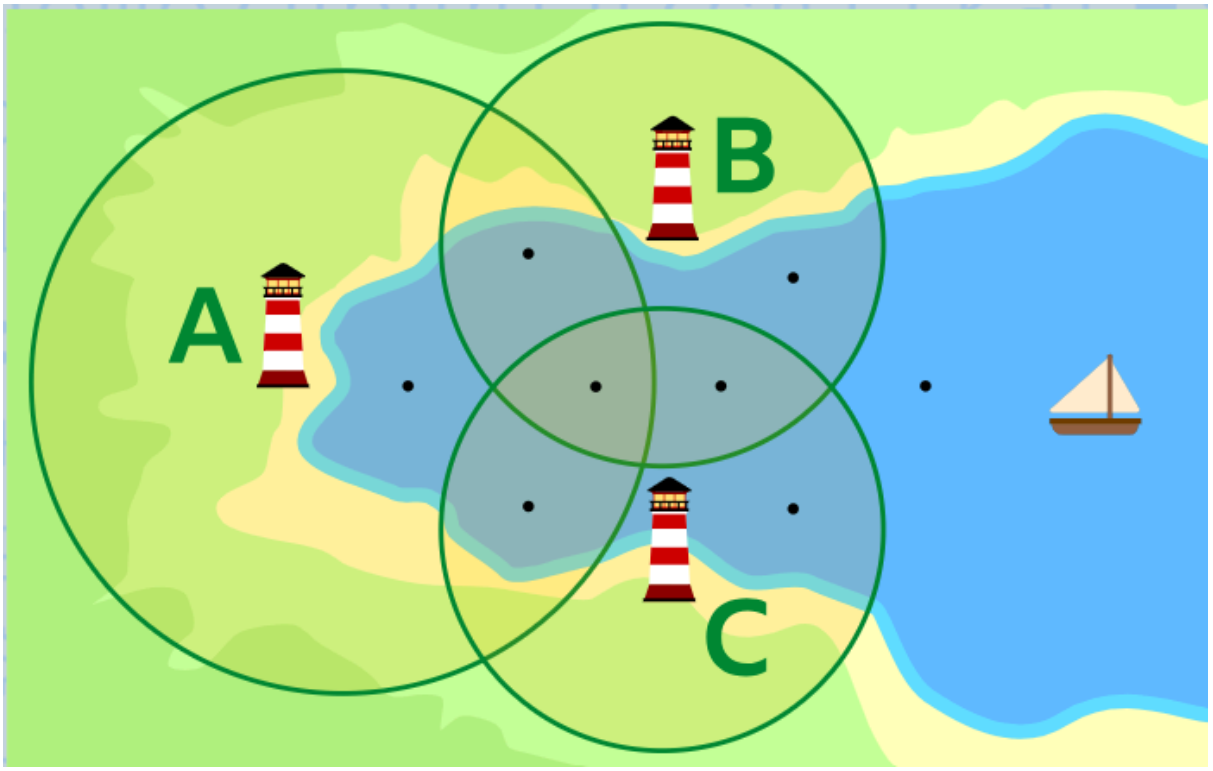
start(it)
www.startit.lv

Saturs

Bākas

Čehija

Kapteinis Bens atrodas uz kuģa. Viņam līča un bāku karte, bet viņš nezin kur viņš ir. Katrai bākai ir aplis apkārt un, kad Bens atrodas iekšā konkrētās bākas aplī, viņš var redzēt to bāku.

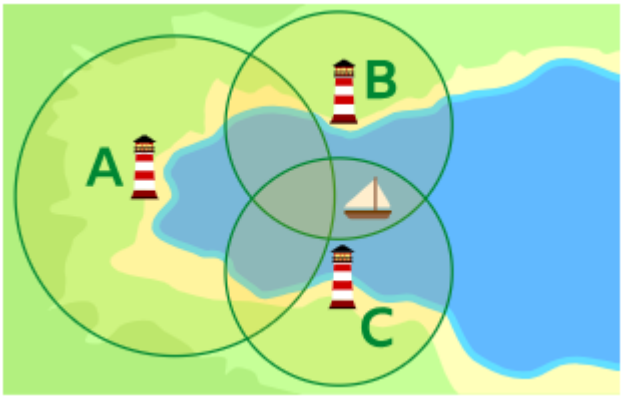


Jautājums

Kapteinis Bens var redzēt B un C bākas. Tomēr viņš nevar redzēt bāku A.
Aizvelciet kuģi ar Benu uz to punktu, kur atrodas Bens!
Nospiediet "saglabāt", kad esat pabeidzis.

Skaidrojums

Bena kuģis ir parādīts šajā bildē:



Bens redz 2 bākas - B un C. Tātad kuģis ir tur kur šie apļi pārklājas, bet nav tajā aplī, kurš ir ap A bāku.

Šī ir informātika

Datorzinātnē mums bieži ir jāapraksta saistība starp objektiem vai objektu kopām. Mēs neizmantojam vārdiskus aprakstus, bet izmantojam simbolu un noteikumu kopumu, ko sauc par Būla algebru. Tad mūsu uzdevumu varētu aprakstīt šādi: B un C, bet nav A.

Šajā uzdevumā izmantoto attēlu sauc par Venna diagrammu. Venna diagrammā izmanto pārklājošos apļus vai citas figūras, tādējādi attēlojot loģiskās attiecības starp divām vai vairākām elementu kopām. Bieži vien tās kalpo, lai grafiski attēlotu lietas, uzsverot, ka izmantotie objekti ir līdzīgi vai atšķirīgi.

Picas ballīte

Vācija

Džons rīko picu ballīti. Viņš zina savu draugu iecienītākās picas piedevas:

Alise			
Bobs			
Kristiāns			
Dana			

Džons vēlas pagatavot vienu picu ar 3 piedevām. Tāpēc viņš vēlas izvēlēties 3 piedevas tā, lai kopumā tiktu apmierināti maksimāli daudz draugu.

Uzdevums

Kādas 3 picas piedevas Džonam ir jāizvēlas, lai iepriecinātu maksimāli daudz viņa draugus?

Skaidrojums

Šī ir pareizā atbilde: sēnes, siers, sīpoli.

Visi vēlas sēnes, trīs cilvēki (gandrīz visi) vēlas sieru un divi cilvēki vēlas sīpolus. Tikai šāda kombinācija var apmierināt visus draugus.

Kā var atrast risinājumu? Var vienkārši pamēģināt visus variantus un noteikt kurus draugus tas apmierina. Tomēr to var arī atrisināt, pieejot sistemātiski.

1. solis: saskaitiet katra piedevas biežumu.

Jāņa draugu iecienītākās piedevas varam attēlot tabulā, kuras augšējā rindā ir piedevas, bet kreisajā kolonā - draugu vārdi. Mēs varam norādīt, kuras piedevas ir iecienītākās, ievietojot skaitli 1 tajā rūtiņā, kur persona un piedeva satiekas. Piemēram, mēs ievietosim skaitli 1 šūnā, kas attiecas uz Alisi un Sarkano piparu.

						
Alise	1	1			1	
Bobs		1	1			1
Kristiāns		1		1	1	
Dana		1			1	1
Cik populāra ir piedeva	1	**4**	1	1	**3**	**2**

2. solis: Nosakiet trīs vispopulārākās piedevas

Pēc tam saskaitām vērtības katrā rindā. Vispopulārākā piedeva ir sēnes, ko minējuši visi 4 draugi, kam seko siers (minēts 3 reizes) un sīpoli (2 reizes pieminētas). Sarkanie pipari, ananāsi un salāti ir vismazāk pieminēti, katrs pieminēts tikai vienu reizi. Tā kā sēnes, siers un sīpoli ir vispopulārākās piedevas, to kombinācija apmierina maksimālo daudz kopējo vēlmju skaitu: 9.

Tā ir informātika:

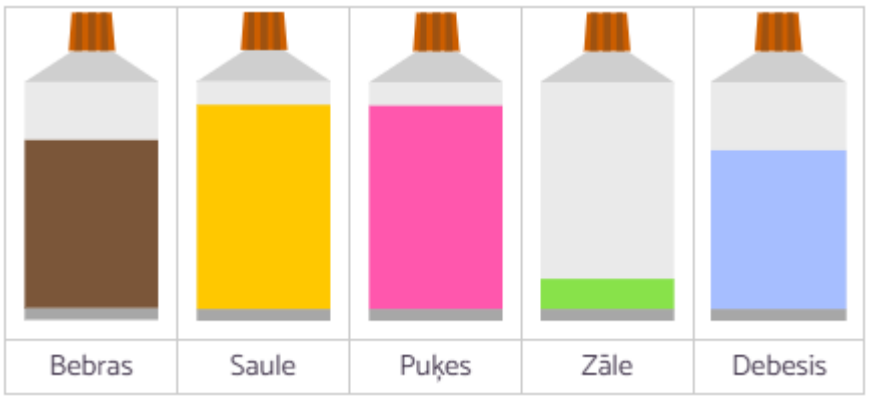
"Picas ballītes uzdevumu" vairāku iemeslu dēļ var uzskatīt par tādu informātikas uzdevumu, kas uzsver koncentrēšanos uz informācijas apstrādi, pārvaldību un analīzi (draugu iecienītākās picas piedevas), lai iegūtu vēlamo informāciju (vēlamākās picas piedevas) un pieņemtu lēmumus, lai atrisinātu uzdevumu, ievērojot konkrētus ierobežojumus (piemēram, ierobežotus resursus).

Šajā uzdevumā budžets mūs ierobežo līdz vienai picai ar trim piedevām, kas atbilst reālās pasaules scenārijiem, kuros risinājumi jāoptimizē, ievērojot noteiktos ierobežojumus. Turklāt, lai atrisinātu uzdevumu, tiek izmantotas algoritmiskas stratēģijas, piemēram, skaitīšanas masīvs vai biežumu masīvs. Katras piedevas biežuma noteikšana un trīs labāko piedevu izvēle, lai maksimizētu apmierinātību, ietver algoritmisko domāšanu, kas informātikā ir viena no pamatprasmēm.

Bildes krāsošana

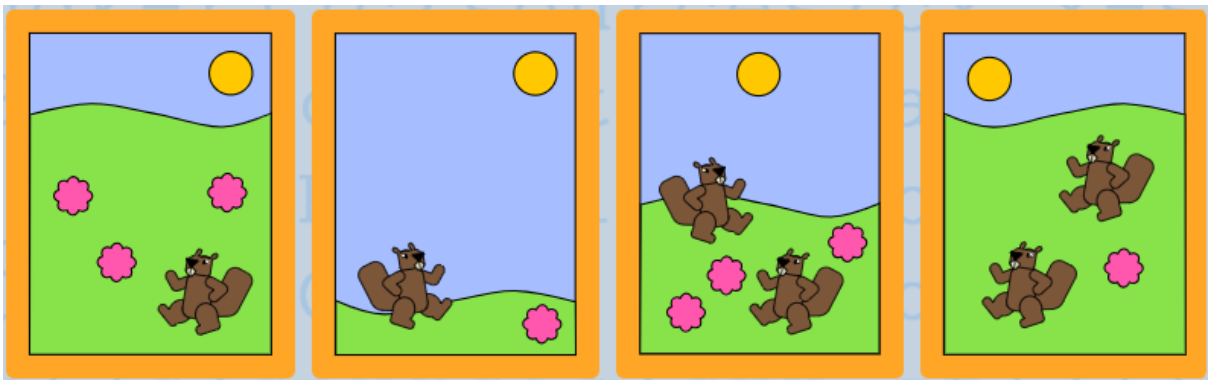
Somija

Laura vēlas izkrāsot dažādas bildes. Viņa sāka ar 5 tūbiņām un ir iztērējusi dažādus apjomus krāsas uz katru bildi. Dažas bildes ir lielākas un, lai tās izkrāsotu, nepieciešams iztērēt vairāk krāsas. Lejā ir redzams cik daudz krāsas ir palikušas tūbiņās.

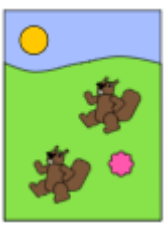


Jautājums

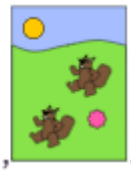
Kuru bildi Laura izkrāsoja?



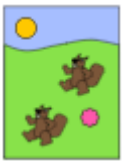
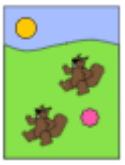
Skaidrojums

Pareizā atbilde ir: 

Lai atrastu pareizo atbildi, jāsalīdzina izkrāsotie elementi ar krāsu tūbiņām. Var redzēt, ka zaļā krāsa ir visvairāk izmantotā un zilā ir otra visvairāk izmantotā. Tas atbilst šīm bildēm

 ,  ,  .

Mēs varam izslēgt šo opciju  jo zilais laukums ir lielāks par zaļo.

Krāsu tūbiņas parāda, ka dzeltanās un rozā tūbiņas ir vienlīdz tukšas. Šajās bildēs, rozā krāsa ir izmantota vairāk nekā dzeltanā:  un  . Tikai šajā bildē abas krāsas ir izmantotas vienādā daudzumā  .Tāpēc atbilde  ir pareiza.

Šī ir informātika

Šajā uzdevumā ir iespējams atpazīt attēlu, vadoties pēc krāsām. Iespējams uzdot sev tādus jautājumus kā: Kuras krāsas attēlā ir izmantotas? Kura krāsa tiek izmantota visbiežāk? Kuras krāsas tiek izmantotas visretāk? Vai kāda krāsa tiek izmantota biežāk nekā cita? Lai atbildētu uz šiem jautājumiem, attēlā jānošķir dažādi laukumi, jānovērtē laukumu lielums, jāsalīdzina laukumu lielumu vērtības un jāpieņem attiecīgi lēmumi. Šādas pārbaudes var veikt arī automātiski, izmantojot datorprogrammas. Datorprogrammas var palīdzēt atrast mobilajā tālrunī saglabātās fotogrāfijas. Izpētot krāsas, datorprogramma var atšķirt ainavas fotogrāfijas no portretiem, jo tajās augšējā daļā ir daudz zilās krāsas, bet apakšā - daudz zaļās. Analizējot satelītattēlu krāsas, datori var noteikt, cik blīvi apbūvēta ir kāda teritorija un vai dabas rezervātā nav notikusi nelikumīga mežu izciršana. Datorzinātņu jomu, kas automātiski analizē attēlus, sauc par datorredzes jomu.

Saulainās dienas

Vācija

Ir saulaina diena

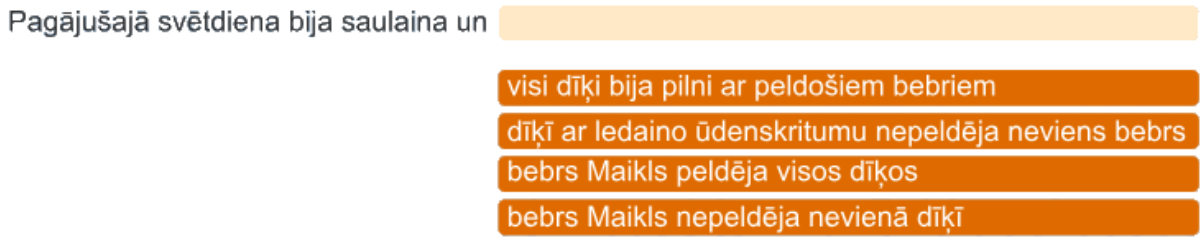
Bebrs Maikls saka: "*Saulainās dienās katrā dīķī peld vismaz viens bebrs.*"

Viņa māsa Kate atbild: "*Tā nav taisnība!*

Pagājušā svētdiena ir piemērs tam, ka tu kļūdies."

Jautājums

Pieņemsim, ka Katei ir taisnība. Pabeidziet šo teikumu:



Skaidrojums

Pareizā atbilde ir:

Pagājušajā svētdiena bija saulaina un dīķī ar ledaino ūdenskritumu nepeldēja neviens bebrs.

Šajā gadījumā ne katrā dīķī peld bebrs, jo dīķī ar ledaino ūdenskritumu tā nav. Turklāt mēs zinām, ka pagājušā svētdiena bija saulaina. Tātad pagājušā svētdiena ir piemērs saulainai dienai, kurā ir vismaz viens dīķis bez peldošiem bebriem. Tādējādi Toms kļūdās, apgalvojot, ka katrā saulainā dienā katrā dīķī peld bebrs.

Pārējie atbilžu varianti nepierāda, ka Toms kļūdās:
A variants saka: "Pagājušā svētdiena bija saulaina, un visos dīķos peldēja bebri." Tas nozīmē, ka katrā dīķī peld bebri, kas atbilst Toma apgalvojumam.

D variants: "Pagājušajā svētdienā bija saulains laiks, un bebrs Maikls nemaz nepeldēja." Tomēr visos dīķos varētu peldēt arī citi bebri, un tādā gadījumā Toma apgalvojums joprojām būtu apmierinošs.

C variants: "Visi dīķi bija pilni ar peldošiem bebriem." Tas nozīmē, ka katrā dīķī ir bebri, kas sakrīt ar Toma teikto.

Tā ir informātika:

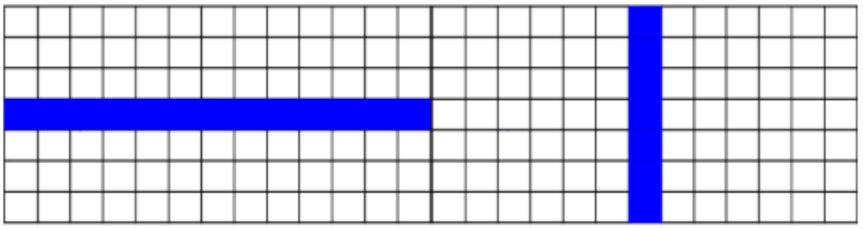
Toms un Kate apspriež Toma apgalvojuma patiesumu, izmantojot Būla loģikas teoriju, kur apgalvojumi tiek divvērtību veidā klasificēti kā patiesi vai nepatiesi. Šī teorija, ko dēvē par "loģiku", ietver sarežģītu apgalvojumu konstruēšanu, izmantojot tādus apzīmējumus kā “un”(AND), “vai”(OR), negācija (NOT), kā arī tādus kvantorus kā “visiem”(EVERY) un “vismaz vienam”(AT LEAST ONE).

Loģiku izmanto diskusijās, palīdz izvairīties no pārpratumiem un kļūdām, piedāvājot strukturētu apgalvojumu pārbaudes metodi. Turklāt šī teorija tiek būtiski izmantota skaitļošanas sistēmās. Katra datorprogramma galu galā balstās uz sīkām fiziskām komponentēm, kas realizē loģiskos operatorus. Otrkārt, programmēšanas valodās tiek iekļauti loģiski apgalvojumi, kā tas redzams tādās konstrukcijās kā "JA ir vasara UN spīd saule, TAD jāieslēdz gaisa kondicionētājs".

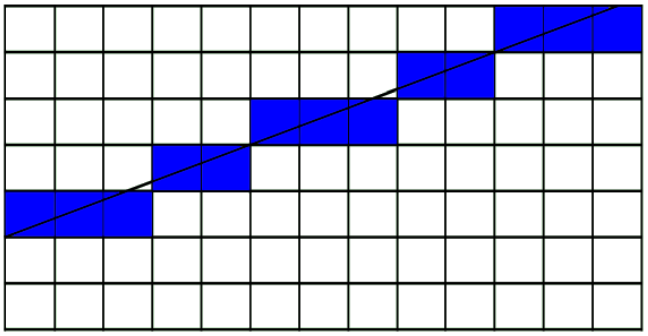
Līniju zīmēšana

Somija

Pikseļi ir mazi kvadrātiņi režģī, ko dators izmanto attēlu attēlošanai. Zīmēt horizontālas vai vertikālas līnijas ir vienkārši, jo tādā veidā blakus esošie pikseļi tiek aizpildīti.

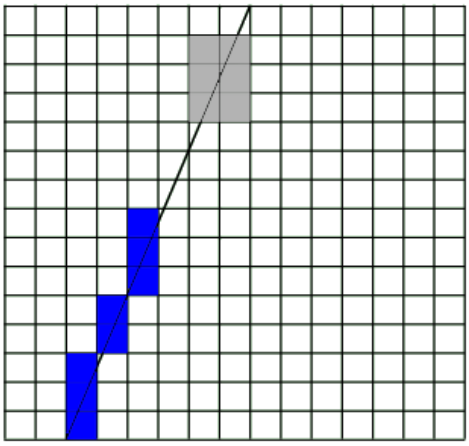


Diagonālās līnijas ir jāzīmē, izmantojot horizontālo un vertikālo pikseļu kombināciju, tāpēc tās nebūs precīzas. Šeit varat redzēt pikseļu zīmējumu, kas attēlo diagonālās līnijas daļas. Šis modelis ir horizontālo (vai vertikālo) pikseļu savienojums, kas atkārtojas pa diagonālu līniju.

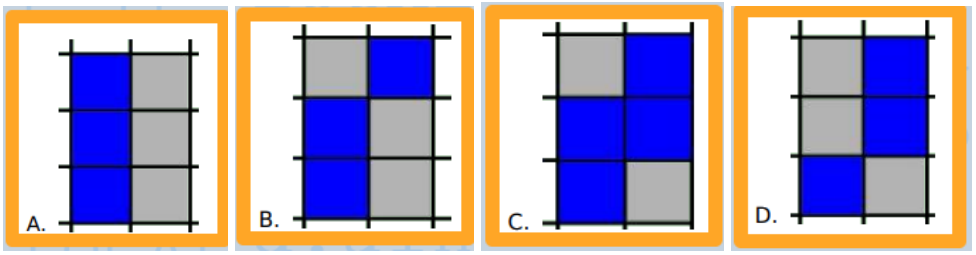


Jautājums

Kurš no pelēkajiem pikseļiem kļūs zils, lai pabeigtu zīmējumu, kas attēlo attēloto diagonālo līniju?

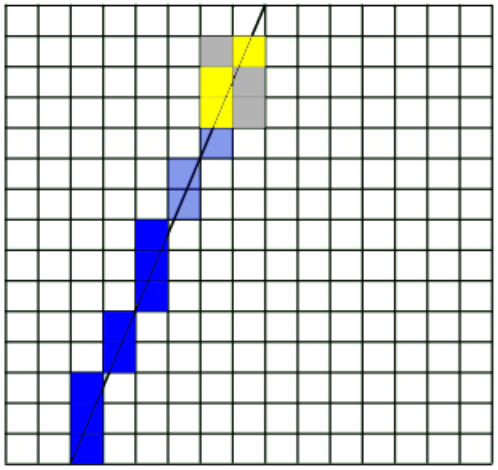


Atbilžu varianti



Skaidrojums

Pareizā atbilde ir B, kā parādīts diagrammā zemāk, kur trūkstošie pikseļi ir iekrāsoti gaiši zilā krāsā, bet pareizais atbilžu variants ir iekrāsots dzeltenā krāsā.



Tā ir informātika:

Viens no datorgrafikas attīstības virzītājspēkiem bija pēc iespējas reālistiskāku attēlu radīšana datora ekrānā. Informātikas pirmsākumos, kad 20. gadsimta 70. gados izšķirtspējas ierobežojumi bija 320 x 200 pikseļi, daudzas grafikas šķita ļoti raupjas un nereālistiskas. Lai palīdzētu izlīdzināt attēlu malas, tika izgudrotas tādas metodes kā robojumu izlīdzināšana.

Šajā uzdevumā tiek demonstrēts viens vienkāršs līniju zīmēšanas algoritms. Līniju, kas nav vertikāla vai horizontāla, var attēlot kā pikseļu virkni. Šis algoritms izmanto līnijas slīpumu, lai noteiktu, kuriem pikseļiem jābūt ieskrāsotiem, bet kuriem - nē. Pamatalgoritms, kas tika izmantots šajā uzdevumā, ir pazīstams kā Bresenhema algoritms.

Pilnveidotāki algoritmi, piemēram, Guptas-Sprūla algoritms, nodrošina daudz precīzāku attēlu un reālistiskāku konkrētas līnijas attēla iegūšanu.

Pirāts un dārgumi

Čehija

Salā kaut kur ir apglabāta dārgumu lāde. Pirāts ir saņēmis norādījumus, kā sasniegt dārgumu lādi. Instrukciju secībā ir 4 soļi, un katrā solī ir jāpārvietojas tieši vienu kilometru dienvidu (D) vai austrumu (A) virzienā. Instrukcija arī nodrošina, ka pirāts neiekritīs jūrā, kas ir pilna ar haizivīm.

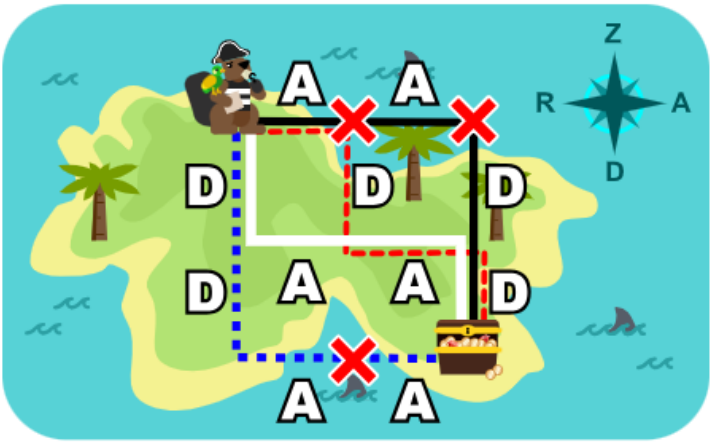
Jautājums

Kuras norādes ir saņēmis pirāts?

D, D, A, A | A, A, D, D | A, D, D, A | D, A, A, D

Skaidrojums

Pareizā atbilde ir D) D, A, A, D



Visi dažādu krāsu ceļi ir parādīti attēlā.
Pareizais maršruts (D) uz dārgumu lādi attēlā ir attēlots ar zaļu ceļu. Visi pārējie maršruti (dzeltens (A), zils (B) un sarkans (pārtrauktā līnija (C)) ved pirātu jūrā. Pirāts nokļūs jūrā, ja sekos pirmajam, otrajam vai trešajam maršrutam.
Mums nav jāzina, cik garš ir viens kilometrs attēlā. No uzdevuma formulējuma var saprast, ka visi četri soļi ir vienādā attālumā. Tā kā visos atbilžu variantos ir divi A un divi D, var uzzīmēt iedomātu 2x2 režģi, kurā pirāts un dārgumu lāde ir divi diagonāli pretēji stūri. Ir viegli pamanīt, ka tikai ejot caur centra punktu, pirāts var izvairīties no iekrišanas jūrā.

Šī ir informātika

Datorzinātnē kartes bieži tiek attēlotas kā grafi ar virsotnēm un šķautnēm, kas savieno virsotnes. Dažādu uzdevumu risināšanā bieži tiek izstrādāti algoritmi, kas balstīti uz grafa datu struktūru. Šajā uzdevumā neredzamā karte satur 9 virsotnes un 12 šķautne sar papildu ierobežojumu, ka šķautnes ir vērstas tikai horizontālā vaivertikālā virzienā. Pārvietošanās dotajā instrukcijā atbilst kustībai grafā no vienas virsotnes uz otru pa tās savienojošo šķautni.

Šis uzdevums prasa zināmu algoritmisko domāšanu, lai izlemtu, ar kādu algoritmu (instrukciju secību) sasniegt mērķi (nokļūt līdz dārgumu lādei), vienlaikus ievērojot ierobežojumu (neiekrist jūrā). Nepieciešama arī zināma dekompozīcija, jo risinātājam ir jāiztēlojas, kā atbilde, kods E, E, S, S sastāv no divām instrukcijām E, S (kas nav precīzi aprakstītas, jo mēs nezinām, cik daudz ir 1 jūdze attēlā) un kā īsti izskatās pirāta pārvietošanās.

Vārdu ķēde

Kanāda

Bebrs spēlējas ar kodiem, kas sastāv no trim simboliem. Tas veido kodu ķēdes tā, ka no viena koda uz nākamo ķēdē mainās tikai viens simbols.

Piemēram, šādi kodi var tikt sakārtoti kodu ķēdē: XUG → XUD → XED → KED

Bebrs ir izveidojis deviņus kodus: TOF, XEW, TEF, CET, COF, TEW, COT, CEF, and XEF.

Viņš tos sadala trīs kodu ķēdēs tā, lai katrs no deviņiem kodiem tiktu izmantots tikai vienā kodu ķēdē.

Jautājums

Neviena no šīm kodu ķēdēm nepārkāpj dotos noteikumus, bet viena no tām padara neiespējamu trīs kodu ķēžu izveidi, neatkārtojot nevienu koda vārdu.

Kurš variants nevarētu būt viena no Bebra kodu ķēdēm?

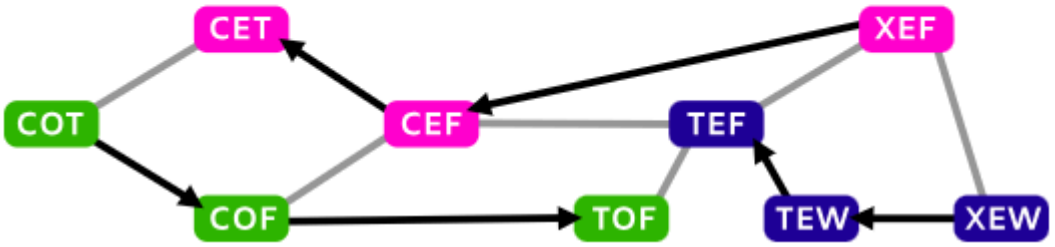
XEW → TEW → TEF | COF → COT → CET | TEF → CEF → COF | CEF → CET → COT

Skaidrojums

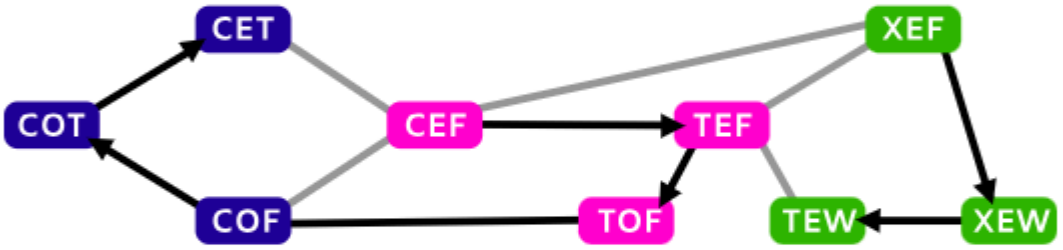
Pareizā atbilde ir TEF → CEF → COF.

Lai atrisinātu šo uzdevumu, ir lietderīgi uzzīmēt diagrammu. Šajā diagrammā divi vārdi ir savienoti ar līniju, ja tie spēj sekot viens otram vārdu ķēdē.

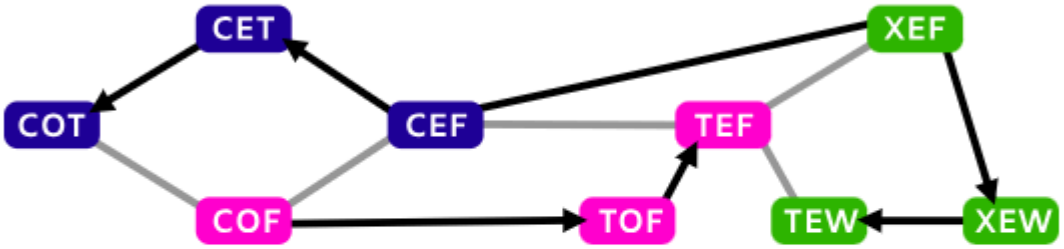
Variants XEW → TEW → TEF ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt XEF → CEF → CET un COT → COF → TOF, kā parādīts diagrammā.



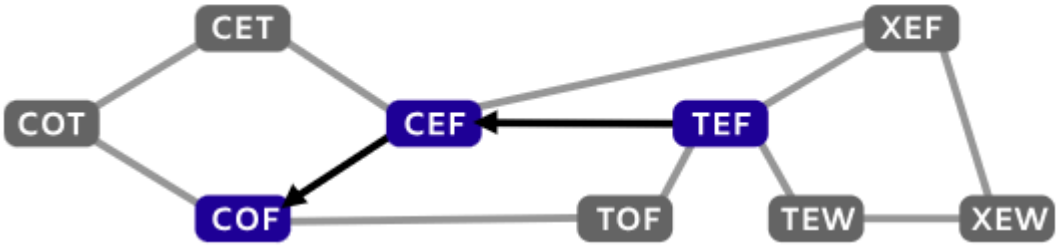
Variants COF → COT → CET ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt CEF → TEF → TOF un XEF → XEW → TEW, kā parādīts diagrammā.



Variants CEF → CET → COT ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt COF → TOF → TEF un XEF → XEW → TEW, kā parādīts diagrammā.



Variants TEF → CEF → COF nav iespējams kā vārdu ķēde, jo, piemēram, diagrammā TOF ir savienots tikai ar TEF un COF. Tādēļ TOF ir jāatrodas vienā ķēdē ar vismaz vienu no šiem vārdiem. Tā kā TEF un COF abi atrodas kodu ķēdē atbilžu variantā C, bet TOF nē, tas nozīmē, ka nav iespējams izveidot ķēdi ar TOF.



Šī ir informātika

Zīmējumos, ar kuriem mēs izskaidrojām risinājumu, redzamās konstrukcijas sauc par grafiem. Tos izmanto, lai attēlotu attiecības starp objektiem. Grafi sastāv no virsotnēm (kas atbilst objektiem) un tās savienojošām šķautnēm (kas atbilst attiecībām starp objektiem). Šajā uzdevumā katrs vārds atbilst grafa virsotnei , bet šķautne starp divām virsotnēm norāda, ka šie divi vārdi atšķiras tieši vienā pozīcijā. Šīs divas virsotnes tiek sauktas par kaimiņu virsotnēm (vai vienkārši par kaimiņiem).

Piemēram, vārdu ķēdē BAD —RAD — RAT — ROT, piemēram, BAD and RAD ir kaimiņi, bet BAD and RAT nav.ieno atbilstošās virsotnes caur to kopīgo kaimiņu RAD. No katra no deviņiem vārdiem var sasniegt visus pārējos vārdus. To var izdarīt tieši (ja attiecīgās virsotnes ir kaimiņi) vai izmantojot garāku ceļu, kas ietver vairākus kaimiņus. Lai atrisinātu šo uzdevumu, mums ir vajadzīgi trīs šādi ceļi. Katra virsotne ir jāizmanto tieši vienreiz.

Grafi ir populārs līdzeklis, ko datorzinātnieki izmanto sarežģītu uzdevumu modelēšanā. Programmās var izmantot grafus, lai atrisinātu daudzus uzdevumus.

Piemēram, navigācijas sistēmas ir programmas, kas izmanto ļoti lielus grafus, lai soli pa solim aizvestu no sākuma punkta līdz galamērķim. Var iedomāties, ka katrā solī automašīna “pārvietojas” grafā pa šķautni no vienas virsotnes uz tās kaimiņu virsotni.

Krāsu lasošais robots

Somija

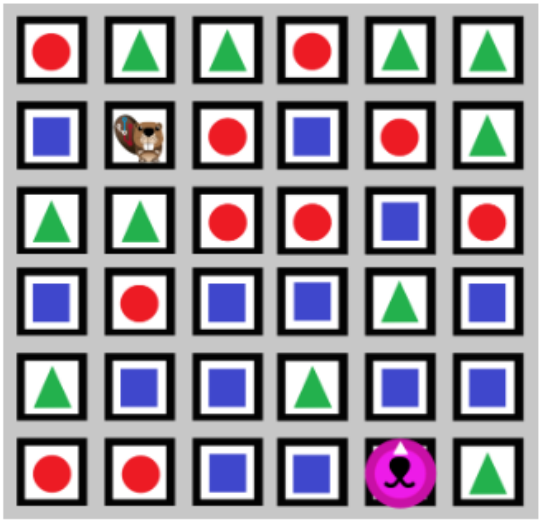
Robots  atrodas sākuma pozīcijā un vēlas sasniegt galamērķi. Tas spēj atpazīt krāsainus simbolus, uz kuriem tas nonāk. Tomēr, kad robots ir sākuma pozīcijā, mēs nevaram redzēt, uz kura simbola viņš stāv.
Robots pārvietojas ņemot vērā krāsainus simbolus un to nozīmi:

- Zils kvadrāts = solis uz priekšu
- Sarkans aplis = pagriezties pa labi un solis uz priekšu
- Zaļš trijstūris = pagriezties pa kreisi un solis uz priekšu

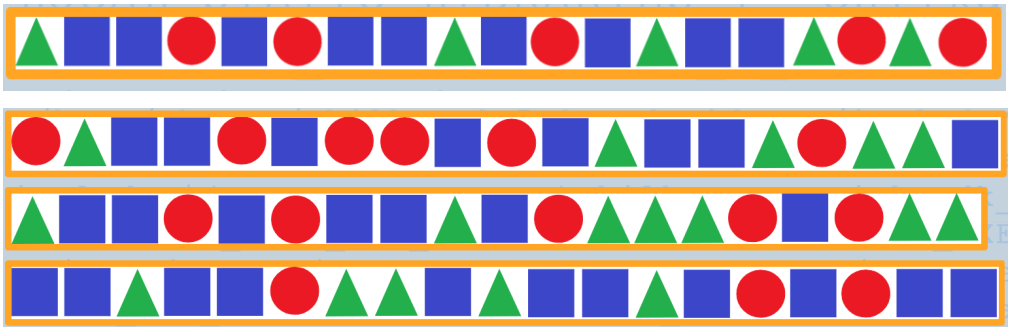
Piemērā robots pārvietojas pa šādu ceļu. Pirmajā solī tas pagriežas pa kreisi un virzās uz priekšu, tad tas pagriežas pa labi un virzās uz priekšu un beigās tas vienkārši virzās uz priekšu. Bultas parāda maršrutu, pa kuru robots pārvietojas.

Jautājums

Kuru ceļu robots izmantoja, lai sasniegtu galamērķi (nokļūt līdz bebram)?



Atbilžu varianti:



Skaidrojums

Pareizā atbilde ir C.
Zemāk esošais attēls parāda vienīgo šī uzdevuma risinājumu.

Variants A ir nepareizs, jo robots pagriezās pa labi pie sarkanā iezīmētā simbola un nevarēja turpināt virzīties uz priekšu. Nākamais solis nevar būt zils kvadrāts.
Variants B ir nepareizs, jo pēc sarkanā iezīmētā simbola robots atdūrās pret sienām un nevarēja turpināt virzīties uz priekšu.
Variants D ir nepareizs, jo pēc programmas izpildes, robots nesasniegs mērķi.

Tā ir informātika:

Programmēšana ir process, kurā tiek radītas instrukcijas programmām, lai veiktu noteiktus uzdevumus vai risinātu problēmas. Tas būtu līdzīgi tam, ka datoram tiktu iedots saraksts ar soļiem, kam sekot, lai sasniegtu konkrētu rezultātu.

Šajā uzdevumā robots darbojas saskaņā ar iepriekš definētu instrukciju vai kodu, balstoties uz krāsām un formām, ko tas atpazīst. Ja jūs iedosiet nepareizu instrukciju, robots izpildīs kaut ko citu, ko jūs nevēlētos.

Programmējot, ir ļoti svarīgi vispirms pārbaudīt savu kodu, lai saprastu, kas notiks, kad to palaistu. Piemēram, programmas palaišana robotos vai lidmašīnās, vispirms tos nepārbaudot, var būt dārga vai bīstama kļūda.

Visgarākā aproce

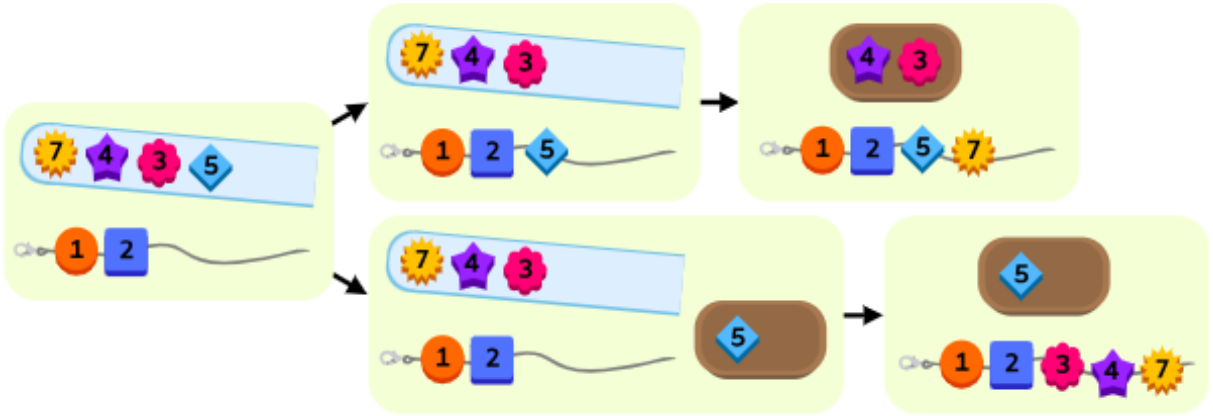
Čehija

Jānis veido aproci. Viņš no stikla tūbiņas ņem pērles uz kurām ir uzrakstīti cipari. Viņs dažas pērles izmanto, bet dažas noliek blakus un neizmanto tās. Viņam ir atļauts uzvilkt pērles uz aukliņas tikai tad, ja:

- aukla ir tukša, vai
- nākamajai pērlei ir lielāks numurs nekā pēdējai pērlei uz auklas.

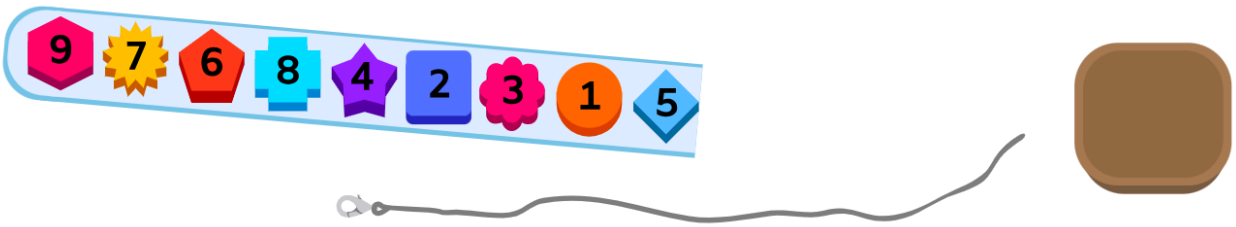
Šajā piemērā pēdējā pērle uz auklas ir 2. pērle. Tālāk Jānis drīkst uzvilkt pērli Nr. 5 no stikla tūbas. Viņš to var arī atlikt malā un neizmantot.

Ja viņš uzvelk pērlīti Nr. 5, viņš var izveidot aproci ar četrām pērlēm 1257. Ja viņš neizmanto 5, viņš var izveidot aproci ar vairāk pērlēm: 12347.



Jautājums

Jānis veido jaunu aproci no pērlēm, kas atrodas šajā stikla tūbiņā:

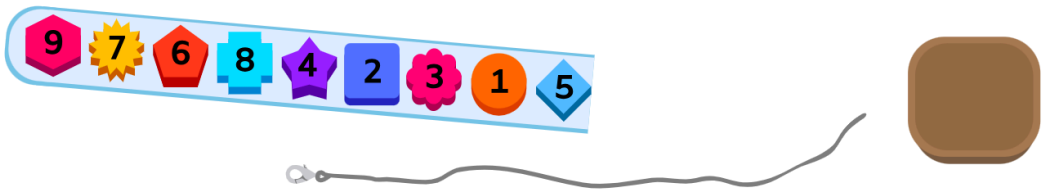


Kāds ir lielākais daudzums pērļu, kuras viņš var uzvilkt uz aproces?

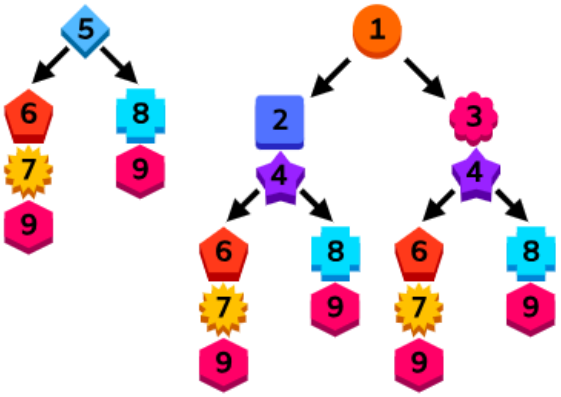
Skaidrojums

Pareizā atbilde ir 6.

Mēs varētu apskatīt visas aproces, kuras sanāktu no šīm pērlēm un saprast, kurā variantā sanāktu izveidot aproci ar vislielāko pērļu skaitu. Tomēr tas prasītu pārāk daudz laika. Tāpēc apskatīsim ciparus, kas ir uzrakstīti uz pērlēm:



Pirmās pērles cipars ir 5. Kad tā ir uzvilkta uz auklas, pēc tās ir iespējams uzvilkt tikai pērles 6, 7, 8 un 9, tāpēc iespējamā secība būtu 5679 un 589. Ja pērli 5 noliek malā un nākamā pērle 1 tiek izmantota, ir iespējamas vairāk kombinācijas. Tas ir parādīts zemāk esošajā attēlā:



Ja pērli 1 noliek malā un tiek izmantota cita pērle, iespējamais pērļu daudzums uz aproces nemainās no tā, kas parādīts attēla augstāk. Piemēram, ja jūs sākat ar pērli Nr. 2, jūs varat iegūt šādas aproces - 24679 or 2489, kurām seko šādas aproces - 124679 un 12489/

Aproces ar vislielāko skaitu pērļu sākas ar Nr. 1 pērli un sastāv no 6 pērlitem: 124679 vai 123679.

Šī ir informātika

Katra stikla tūbiņa satur noteiktā secībā sarindotas pērles. Ja Jānis seko otrajam noteikumam (drīkst likt tikai pērlīti ar lielāku skaitli nekā iepriekšējais), viņš iegūs aproci ar skaitļiem, sakārtotiem augošā secībā. Lai iegūtu visgarāko iespējamo ķēdīti, viņam ir

jānosaka visgarākā iespējamā apakšvirkne, kas ir augoša.
Lai noteiktu šo secību garām virknēm, tas prasītu ļoti daudz laika, ja mēs mēģinātu tās atrast, apskatot visus iespējamos varinatus. Piemēram, ja aprocē būtu 20 pērles, tad šis aprēķins ietvertu vairāk nekā miljons darbību.

Veiksmīgā kārtā, ir izgudroti algoritmi, kas spēj atrast visgarāko augošo virkni diezgan ātri. To var izdarīt, izmantojot tehniku, kas ir nosaukta par dinamisko programmēšanu. Tādiem algoritmiem ir jāveic mazāk par 100 darbībām, lai atrastu garāko augošu apakšvirkni, ja ir dotas 20 dažādas pērlites.

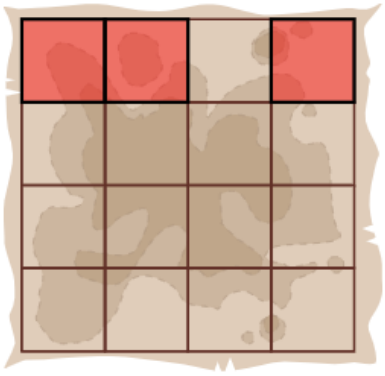
Atrodi dārgumus

Ungārija

Pirāts Pips meklē salā paslēptus dārgumus.

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

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



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

Jautājums

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

Skaidrojums

Pareiza atbilde ir 4.

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

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

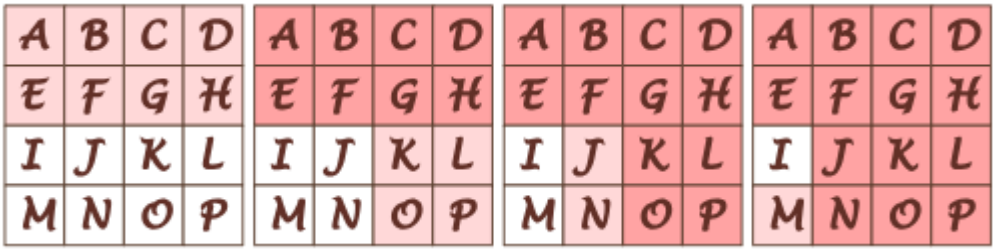
B. Ja ierīce norāda, ka dārgums nav norādītajos reģionos, viņš sadala atlikušo daļu divās daļās un vienu no tām ievada ierīcē.

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

Piemēram, ja viņš ierīcē ievada A,B,C,D,E,F,G,H un ierīce saka “nav”, viņš var ievadīt K,L,O,P (jebkuru atlikušo 4 reģionu grupu). Ja ierīce atkal atbild “nav”, viņš turpina sadalīt atlikušos

reģionus un ierīcē ievada jebkurus 2 atlikušos reģionus (piemēram, J,N). Ja ierīce atbild “nav”, dārgums ir paslēpts vienā no atlikušajiem reģioniem - I vai M. Ja viņš ievada M un ierīce atbild “nav”, viņš droši zina, ka dārgums ir I reģionā.

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



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

Šī ir informātika

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

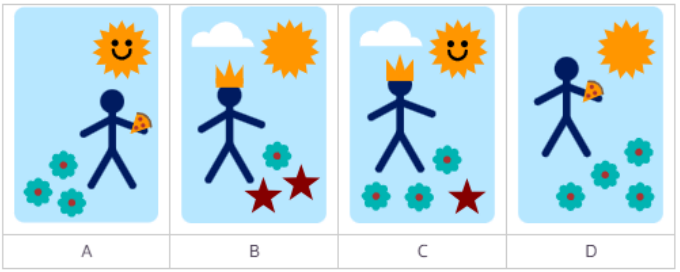
Ja ir 1 000 objektu, ir nepieciešami 10 meklēšanas soļi 2^9<1000<2^10, bet 1 000 000 objektu gadījumā - 20 soļi. Vispārīgi var teikt, ka n objektu meklēšanai ir nepieciešami aptuveni log(n) soļi; logaritma funkcija ir logaritms pie bāzes 2. Tā kā binārā meklēšana ir tik ātra, to bieži datorprogrammās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

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

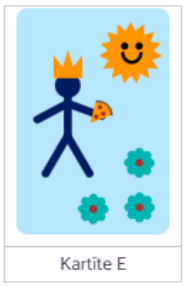
Kartītes

Indija

Jānis zīmē savas kartītes, ievērojot vienu vienkāršu slepenu noteikumu. Viņš ir izveidojis četras kartītes, kas atbilst šim noteikumam: A, B, C, D.



Kārlis apskata kartītes un izveido kartīti E, bet Jānis saka, ka tā neatbilst šim noteikumam.



Jautājums:

Kura no šīm opcijām varētu būt slepenais noteikums, ko Jānis ievēro, zīmējot kartītes?

Kad ir apmācies laiks, tad nav ziedu.

Kad ir apmācies laiks, saule nesmaida.

Ir jābūt vai nu sarkanai zvaigznei, vai picas šķēlītei.

Ja ir picas šķēle, tad nav kroņa.

Skaidrojums

Pareizā atbilde ir: Ja ir picas šķēle, tad nav kroņa.

Apskatot visas iespējas:

Kad ir apmācies laiks, tad nav ziedu: Nē, jo 3. un 4. kartiņā ir mākonis un ziedi.

Kad ir apmācies laiks, saule nesmaida: Ne, jo uz 4. kartes nav ne smaidošas saules, ne mākoņa. Tātad A un D gadījumā slepenais noteikums nebūtu patiess attiecībā uz sākotnējām 4 kartītēm.

Ir jābūt vai nu sarkanai zvaigznei, vai picas šķēlītei: Nē, jo visām kartītēm šis noteikums ir patiess, un 5. kartīte šo noteikumu nepārkāpj.

Ja ir picas šķēle, tad nav kroņa: Jā, šis varētu būt derīgs noteikums, jo nevienā no četrām kartītēm nav picas un kroņa kopā, un, kad Kārlis izveidoja šādu kartīti, Jānis saka, ka viņa noteikums nav ievērots. No piedāvātajiem variantiem šis ir vienīgais iespējamais noteikums: Ja ir picas šķēle, tad nav kroņa.

Šī ir informātika

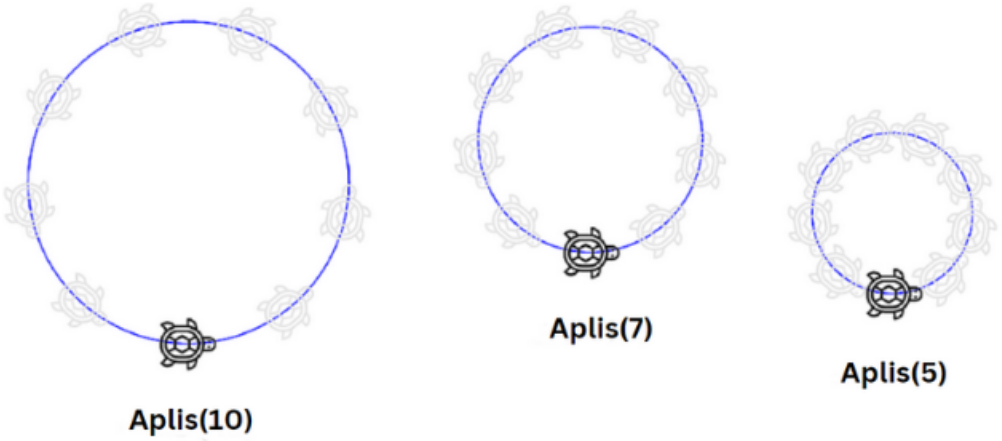
Šis uzdevums ilustrē loģikas pamatus, ko izmanto informātikā, lai mācītu, kā var attēlot un apstrādāt lēmumus un novērojumus. Datori ievēro skaidrus noteikumus, lai noteiktu rezultātus, līdzīgi tam, kā mēs ikdienā novērojam likumsakarības un izdarām izvēli. Tie izmanto pamata loģiku, lai apstrādātu nosacījumus un noteiktu darbības.

Dekompozīcija: Sarežģītu problēmu sadalīšana vienkāršākās daļās. Katrs apgalvojums izolē konkrētu nosacījumu un tā sekas.
Modeļu atpazīšana: Līdzību vai modeļu identificēšana datos. Piemēram, atpazīt, ka noteikti nosacījumi (piemēram, apmācies laiks) pastāvīgi noved pie konkrētiem rezultātiem (nav ziedu vai saule nespīd).
Abstrakcija: Koncentrēšanās uz svarīgām detaļām, ignorējot mazāk būtiskās. Katrs apgalvojums abstrahē sarežģītās reālās pasaules dinamiku un pievēršas tikai loģiskajām attiecībām starp nosacījumiem un rezultātiem.

Bruņurupuča zīmējumi

Somija

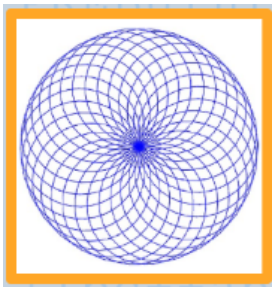
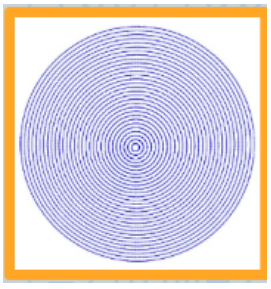
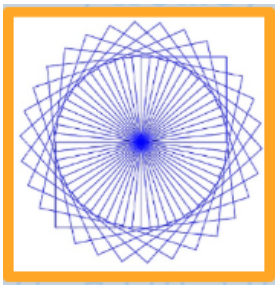
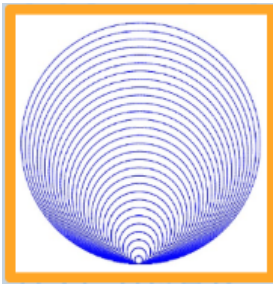
Robots, kurš ir bruņurupuča formā, uz zīmēšanas dēļa izveido zīmējumu atbilstoši instrukcijai. Kad tas saņem komandu zīmēt apli: "aplis(d)", tas uzzīmēs apli, kura izmērs mainīsies atkarībā no parametra d. Mainot parametru d, var tikt uzzīmēti dažāda izmēra apļi (skat. attēlu).



Jautājums

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

Atbilžu varianti

a) 
b) 
c) 
d) 

Pareizā atbilde: D

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

Kāpēc pārējās atbildes ir nepareizas:
A: visi uzzīmētie apļi attēlā ir vienāda izmēra, taču tiem jābūt dažāda izmēra.
B: šiem apļiem nav kopīga punkta, no kura robots sāktu tos zīmēt.
C: šis attēls nav veidots no apļiem, bet no kvadrātiem (paceļoties no centra).

Tā ir informātika:

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

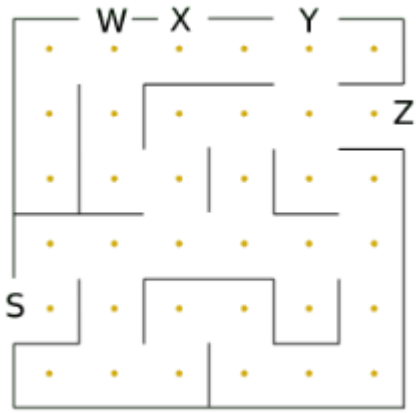
Šī ir algoritmiskā domāšana:

Šis uzdevums prasa pielietot abstrakciju, kas nozīmē atrast būtisko aprakstītajā situācijā. Piemēram, saprast, kā augošā parametra vērtība ietekmē dažāda izmēra apļus. Abstrakcija arī ietver nebūtisku lietu atstāšanu malā. Piemēram, ka apļa centram nav nozīmes un, ka koeficients ietekmē apļa izmēru. Ir arī nepieciešama algoritmizācija, jo ir aprakstīts algoritms, tāpēc ir jāsaprot, piemēram, ka pēc viena apļa uzzīmēšanas nākamais aplis jāsāk tajā pašā punktā tādā pat virzienā.

Labirints

Somija

Mēs aplūkosim šo labirintu:



Sākot no punkta S, mums jāpārvietojas caur dzeltenajiem punktiem. Atrodoties dotajā punktā, mēs varam doties uz kādu no blakus esošajiem punktiem, kas var būt tikai tieši virs, zem, pa kreisi vai pa labi. Mēs nevaram šķērsot melnās līnijas. Mūsu mērķis ir izkļūt no labirinta pa kādu no W, X, Y vai Z caurumiem.

Jautājums

Kāds ir minimālais punktu skaits, ko sanāks apmeklēt pirms iziešanas no labirinta?

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

Skaidrojums

Pareizā atbilde: D

Skaidrojums: Mēs varam izmantot tā saukto breadth-first search algorithm. Vispirms mēs atzīmējam ar “1” punktu, kas atrodas vistuvāk S. Tas ir 1. līmenis. Pēc tam ar "2" atzīmējam visus punktus, kas vēl nav atzīmēti, bet ir sasniedzami viena soļa attālumā no punkta, kas atzīmēts ar "1". Parasti, ja esam atzīmējuši kādu punktu ar "n", sākam ar visiem punktiem, kas atzīmēti ar "n", un visus punktus, kas vēl nav atzīmēti, bet ir sasniedzami viena soļa attālumā no šiem punktiem, atzīmējam ar "n + 1".

Mēs turpinām, kā aprakstīts iepriekš, līdz nav palikuši nemarķēti punkti. Pēc tam skatāmies, kurai izejai ir vismazākais skaitlis, kas norāda līmeni. Mūsu gadījumā tā ir izeja W, un šis skaitlis ir "8".

Tā ir informātika:

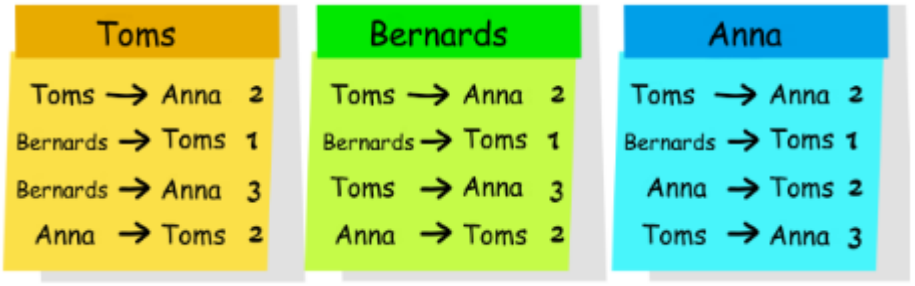
Šis ir strukturēts grafiks, kas sastāv no virsotnēm, kuras savienotas ar malām. Daudzas reālās pasaules problēmas var pārveidot grafikos.
Algoritms “Breadth-first search” (BFS) kopā ar “Depth-first search” ir viens no pamatalgoritmiem, ko izmanto, lai efektīvi un bez liekām darbībām sasniegtu visas grafika virsotnes.
BFS pārbauda virsotnes to attāluma secībā no sākuma virsotnes. Piemēram, šajā uzdevumā BFS sākas no sākumpunkta S un vispirms pēta visus punktus, kurus var sasniegt vienā solī, tad tos, kurus var sasniegt divos solī, un tā tālāk, līdz tiek atrasta izeja.

Uzskaites sistēma

Austrija

Anna, Bernards un Toms ik pa laikam cits citam aizdod bumbiņas. Lai pārliecinātos, ka neviens no draugiem nav kļūdījies, katrs uz sava papīra lapiņas uzraksta, kurš kuram ir aizdevis, cik bumbiņu.

Pēc nedēļas viņi salīdzina savas lapiņas:



Jautājums

Trīs draugiem rodas aizdomas, ka ir pieļauta kļūda. Ja tā, tad kurš ir pieļāvis šo kļūdu?

a) Anna
b) Bernards
c) Toms
d) Neviens nav pieļāvis kļūdu

Skaidrojums

Pareizā atbilde ir: Toms

Ja neviens nebūtu kļūdījies, uz katra papīra pieraksti būtu vienādā secībā. Trīs ieraksti ir vienādi uz visiem papīriem. Attēlā tie ir atzīmēti ar zaļu, dzeltenu un oranžu līniju. Ir tikai viena atšķirība: ierakstu var atrast uz divām pierakstu lapiņas un tikai uz Toma pierakstu lapiņas.

Ja pieņemam, ka kļūdījusies tikai viena persona, tad tai jābūt Tomam.

Šī ir informātika

Datorzinātnē katru atsevišķu faktu saucam par ierakstu un uzglabājam datubāzē. Bērni mūsu stāstā ir nolēmuši, ka viņiem nebūs tikai viena centrālā datubāze, bet katram būs sava "datubāze". Tagad ir daudzas fiziski atsevišķas datubāzes, kas tomēr - kā sagaidāms - atspoguļo vienu un to pašu saturu. Šajā kontekstā apzināti ir notikusi atteikšanās no vienas centrālas iestādes, kurai visi uzticas. Katrs bērns ir līdzvērtīgs citiem, un bērnu darbības notiek vienādranga tīklā, kurā katram dalībniekam ir tādas pašas tiesības un pienākumi kā visiem pārējiem.

Vienlīdzība un datu sadale ir viena no blokķēdes tehnoloģijas pamatidejām.

Uzdevumā minētā pieeja darbojas tikai tad, ja lielākā daļa dalībnieku ir godīgi. Ja rodas domstarpības, tad dalībnieki balso par to, kura no datu bāzēm ir pareiza.

Blokķēdes tehnoloģija iet soli tālāk, ieviešot konsensa algoritmu, kas, piemēram, ļaunprātīgam dalībniekam prasītu kontrolēt lielāku skaitļošanas jaudu nekā visiem pārējiem dalībniekiem kopā.

Tas de facto nav iespējams, ja tīkls ir pietiekami liels. Tomēr šis uzdevums jau parāda, kā uzticēšanos var stiprināt a priori neuzticamā vidē, sadalot jaudu starp visiem dalībniekiem.

Grand Prix

Somija

Pieci bebri Anna, Deivids, Harrijs, Janita un Raivis sacentīsies Formula E pasaules čempionāta E-Prix. Zemāk esošajā tabulā ir sniegta informācija par sacīkšu trasēm dažādās pilsētās, kurās notiks sacīkstes šajā sezonā. Sacīkšu sezona sākas janvārī un beidzas jūlijā.

Pilsēta	Mehiko	Rīga	Roma	Portlanda	Londona
Sacensību mēnesis	Janvāris	Februāris	Aprīlis	Jūnijs	Jūlijs
Līkumi	16	21	19	12	22
Garums	4.3 KM	2.49 KM	3.36 KM	3.2 KM	2.25 KM
Virziens	Pulksteņrādītāja	Pulksteņrādītāja	Pret - Pulksteņrādītāja	Pulksteņrādītāja	Pret- Pulksteņrādītāja
Slīpums (palielinās)	0	0	1	0	1
Slīpums (pazeminās)	0	0	1	0	1

Iepriekšējās sezonās dalībniekiem ir šādi rezultāti:

1. Raivim un Janitai labāk veicas trasēs, kurām ir 20 vai vairāk līkumi.
2. Harriam nepadodas braukt trasēs, kur viņam ir jābrauc pretēji pulksteņrādītāja virzienam.
3. Raivis mīl sacensties trasēs, kurās ir slīpumi.
4. Annai vislabāk veicās trasēs, kas ir garākas par 3 km.
5. Deividam vislabāk veicas trasēs ar nepāra līkumu skaitu un kas ir garākas par 3 km.
6. Anna parasti uzvar sezonas sākumā.

Jautājums

Katrā sacīkstē uzvar viens bebrs. Kurš no šiem atbilžu variantiem visticamāk ir patiess, ja tiek izpildīti visi iepriekšminētie nosacījumi?

Skairojums

Pareizā atbilde ir C.

Šo uzdevumu var atrisināt, izmantojot deduktīvo loģiku, balstoties uz iepriekš novērotajiem apbalvojumiem. Mums jāuzskata šos apgalvojumus par ierobežojumiem, kas ir jāievēro, nonāktu līdz pareizajai atbildei.

Anna uzvar sezonas sākumā, un tā kā viņai vislabāk veicas trasēs, kas ir garākas par 3 km, tikai Mehiko atbilst šiem nosacījumiem – pirmās sacīkstes un trases garums > 3 km. Lai gan Rīgā sacīkstes notiek sezonas sākumā, trases garums ir < 3 km. Romas, Portlendas un Londonas sacīkstes notiek vēlāk sezonā.

Deividam vislabāk veicas trasēs ar nepāra likumu skaitu un tajās trasēs, kuras ir garākas par 3 km. Tikai Roma atbilst šiem nosacījumiem – trasē ir nepāra līkumu skaits un trases garums > 3 km. Lai gan Rīgā ir nepāra līkumu skaits, trases garums ir < 3 km. Citu pilsētu trasēs ir pāra līkumu skaits.

Harijs nebrauc labi, kad viņam ir jābrauc pretēji pulksteņrādītāja virzienam. Tāpēc viņam var labi veikties tikai Mehiko, Rīgā un Portlendā, jo tās ir pulksteņrādītāja virzienā. Tā kā Anna, visticamāk, būs uzvarētāja Mehiko, Harrijs varētu būt uzvarētājs Rīgā vai Portlendā.

Janitai labāk veicas trasēs ar 20 vai vairāk līkumiem. Tāpēc viņš varētu būt uzvarētājs Rīgā vai Londonā.

Raivim labāk veicas trasēs ar 20 vai vairāk līkumiem un, ja trase ir slīpa. Tikai Londona atbilst šiem kritērijiem. Tāpēc Raivis, visticamāk, uzvarēs Londonā.
Tāpēc varam secināt, ka Janita uzvarēs sacīkstēs Rīgā, bet Harrijs – Portlendā.

C variants atbilst iepriekš minētajiem nosacījumiem.

Tā ir informātika:

Datorzinātnieki bieži vien nodarbojas ar to, lai izlemtu, vai konkrētais priekšlikums atbilst zināmajiem ierobežojumiem. Viņi ir apmācīti iegūt pēc iespējas vairāk informācijas no dotajiem datiem (zināmajiem faktiem). Šajā uzdevumā mums ir tabula, kurā apkopotas formulas E sacīkstēs izmantoto sacīkšu trašu īpašības. Ir sniegts noteikumu (novērojumu) kopums, kas nosaka ierobežojumus. Šajā uzdevumā jāatrod loģisks risinājums, kas atbilst

visiem ierobežojumiem. Lai savienotu braucēju ar sacīkstēm, kurās viņš uzvarēja, jānosaka, ka tās atbilst zināmiem ierobežojumiem.