

Bebr[a]s

'24

2024. gada 1. kārtas uzdevumi ar atbildēm

4. klase

INFORMATĪVIE ATBALSTĪTĀJI



Izglītības un zinātnes
ministrija

start(it)

www.startit.lv


Saturs

Brīnumpuķe.....	2
Rožkassprādžu kārtošana.....	5
Ricca kārtis.....	7
Bumbiņas.....	10
Virpuļojošās lapiņas.....	13
Bākas.....	15
Picas ballīte.....	17
Saulainās dienas.....	21
Līniju zīmēšana.....	23
Pirāts un dārgumi.....	25
Vārdu ķēde.....	27
Krāsu lasošais robots.....	30
Visgarākā aprobe.....	32
Atrodi dārgumus.....	35

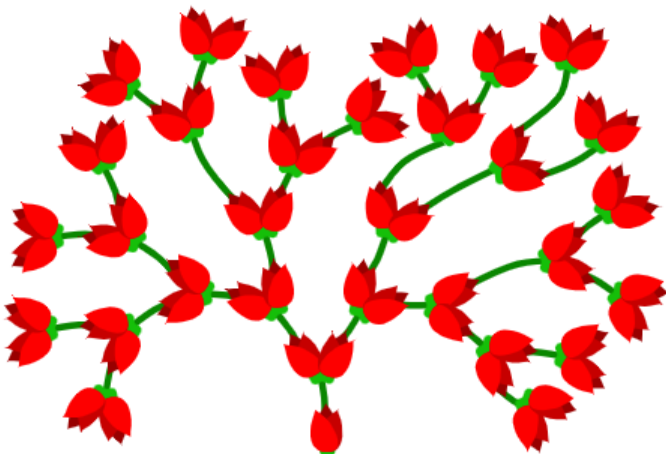
Brīnumpuķe

Vācija

Saullēktā no katra jaunā brīnumpuķes pumpura izaug viens stublājs. Stublājs turpina augt visas dienas garumā. Saulrieta laikā stublājs sazarojas divos jaunos brīnumpuķes pumpuros un beidz augt. Tā tas turpinās katru dienu, un brīnumpuķe kļūst arvien krāšņāka.

Brīnumpuķe pirms saullēkta	Brīnumpuķe pēc pirmās dienas	Brīnumpuķe pēc otrās dienas
		

Pēc vairākām dienām, brīnumpuķe izskatās šādi:



Jautājums

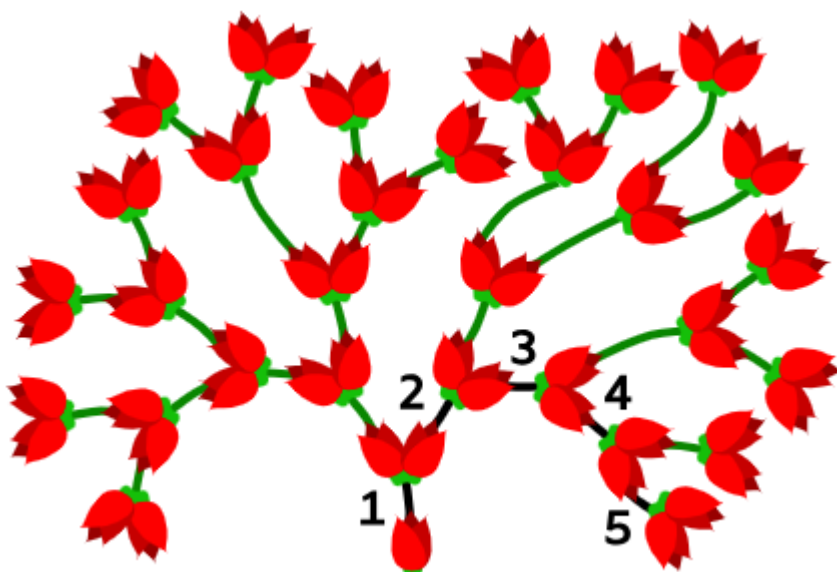
Cik dienas brīnumpuķe ir augusi?

Skaidrojums

Pareizā atbilde ir 5 dienas.

Katras dienas beigās stumbrs no diviem brīnumpuķes zieda pumpuriem ir jau izaudzis kā stublājs ar diviem jauniem pumpuriem. Principā, ir jāseko tikai vienam zaram, lai saskaitītu augšanas dienas, piemēram, malējam zaram labajā pusē.

Tas ir parādīts šajā diagrammā:



Tā ir informātika:

Brīnumpuķes pumpuri aug saskaņā tikai ar vienu noteikumu. Šis noteikums nosaka, ka pumpurs veido vienu zaru un divus jaunus pumpurus. Šādā veidā brīnumpuķe teorētiski var turpināt augt bezgalīgi. Šādus pašpietiekamus noteikumus datorzinātnē sauc par rekursīviem noteikumiem.

Programmēšanā rekursīva funkcija ir funkcija, kas izsauc pati sevi un tai nav vajadzīgi ārēji faktori. Tātad šī funkcija darbojas pati par sevi, un vislaik atkārtojas bezgalīgi, mainoties izmantoto parametru vērtībām.

Svarīgi, lai funkcijā būtu beigšanās nosacījums, t. i., funkcijai vajadzētu izsaukt pašai sevi tikai tad, ja ir izpildīts kāds nosacījums. Šādā gadījumā, pēc noteikta skaita izsaukumu, process tiks pārtraukts, kad nosacījums vairs nebūs izpildīts.

Zināmākais piemērs no matemātikas ir faktoriālā funkcija "n!": $n! = n * (n-1)!$, ja $n > 1$, un $1! = 1$.

Ja n vietā ievadīsiet skaitli 4, kā rezultātu saņemsiet 24.

$$4! = 4 * 3! = 4 * (3 * 2!) = 4 * (3 * (2 * 1!)) = 4 * (3 * (2 * 1)) = 4 * (3 * 2) = 4 * 6 = 24$$

Rekursīvās programmas var būt ļoti skaidras un viegli izprotamas, ja izdodas izprast daudzo tās izsaukumu loģiku. Tomēr, daudzo funkcijas izsaukumu dēļ, tās parasti ir lēnākas nekā iteratīvās programmas un prasa vairāk atmiņas.

Rekursīvās programmas var izmantot arī, lai izveidotu ļoti "dabīgas" grafikas, ko sauc par fraktāļiem.

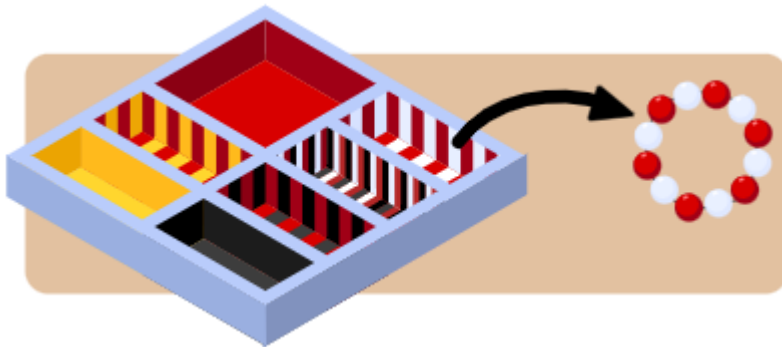
Rožkassprādžu kārtošana

Brazīlija

Viktorijai ir lādīte, kurā viņa sakārto savas rokassprādzes.

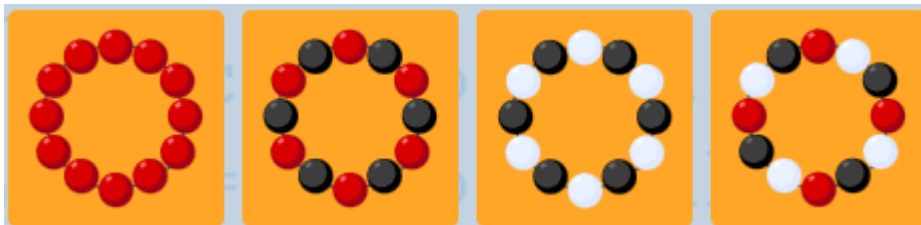
Šajā kastē ir septiņi nodalījumi, un Viktorija katru rokassprādzi liek nodalījumā ar tādu pašu krāsu rakstu.

Zemāk esošais attēls parāda, kur viņa liek katru no savām rokassprādžēm.



Jautājums

Kura no šīm rokassprādžēm NEATBILST nevienam no nodalījumiem Viktorijas kastē?

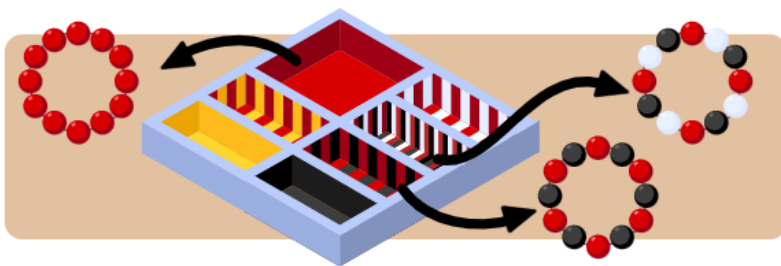


Skaidrojums



Pareizā atbilde ir

Zemāk esošais attēls parāda nodalījumus, kur Viktorija liek rokassprādzes variantos pārējos variantos. Var redzēt, ka rokassprādzes un nodalījumi ir ar tādu pašu krāsu rakstu.



Jāņem vērā, ka Viktorijas kastē nav nodalījuma ar melnbalto krāsu rakstu, tādēļ nav

nodalījuma, kurā varētu ievietot rokassprādzi no varianta



Tā ir informātika:

Bieži vien, kad cenšamies atrisināt kādu problēmu, mums ir daudz datu, ar kuriem strādāt. Tieši tāpat kā šajā uzdevumā, ir daudz mainīgie, kas saistīti ar rokassprādzēm, un tos mēs varam izmantot, lai ievietotu rokassprādzes pareizajos nodalījumos. Ir svarīgi spēt atfiltrēt nevajadzīgos datus, lai būtu vieglāk atrast svarīgos datus un nonāktu līdz risinājumam. Šajā uzdevumā mums ir jāpievērš uzmanība tikai krāsām, nav svarīgi, kāda forma ir pērlītēm, cik daudz tās ir vai kādā rakstā tās parādās. Mums pat nav jāredz visa rokassprādze, lai atrisinātu jautājumu, pietiek redzēt tikai daļu no attēla.

Šī ir algoritmiskā domāšana

Šis uzdevums izmanto abstrakciju. Lai atrastu informāciju, kas nepieciešama lai nonāktu līdz pareizajai atbildei, mums jāignorē nevajadzīgās detaļas par rokassprādzēm. Šajā uzdevumā ir svarīgi saprast, ka krāsa ir svarīgākais mainīgais un tas ir vienīgais, kas mums jāzina par rokassprādzēm, lai atrisinātu uzdevumu.

Ricca kārtis

Čehija

Barbara kolekcijā kartis, uz kurām attēloti briesmoņi ar nosaukumu "Riccas". Visiem Riccām ir tādas pazīmes kā vārds vai acu skaits. Katrs Ricca ir atšķirīgs, tāpēc pazīmju vērtības dažkārt atšķiras. Šeit ir trīs paraugkartes, kurās redzami dažas no šīm pazīmēm un to vērtībām:



Katras pazīmes vērtība tiek atzīmēta noteiktā veidā:

Pazīme	Pazīmju veidi
	vērtības tiek norādītas kā TEKSTS
	Pazīme ir norādīta kā ✘ vai ✔
	pazīmes ir norādītas kā SKAITĻI

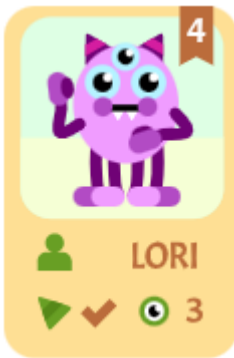
Jautājums

Uz kuras kartītes visas pazīmju vērtības ir atzīmētas pareizā veidā?



Skaidrojums

Pareizā atbilde ir



Šeit redzams kāpēc:

	Pazīmes uz kartītes			
	A)	B)	C)	D)
Vērtība				
	LORI	LORI	LORI	
		3	3	3

Tikai B kartītē katras pazīmes vērtības ir atzīmētas pareizi.

- Atbildēs A), C) un D): Monstram vārdā Lori ir 2 ragi , tāpēc informācija ir pareiza, bet nav atzīmēta pareizā veidā.
- Atbildē A) Ricca monstram Lori ir acis ,tāpēc informācija ir pareiza, bet informācija ir norādīta ar ✓ , tomēr pareizi būtu atzīmēt ar CIPARU, tāpēc vērtībai jābūt skaitlim 3, kam šajā gadījumā ir jābūt ✓ vietā.

Tā ir informātika:

Vērtības uz kartītēm raksturo Ricca monstrus. Taču attēlā ir attēlots daudz vairāk nekā šī informācija. Piemēram, jums ir jāizvēlas vairāk vērtību, ko salīdzināt. Un jums ir arī jādomā par to, kāda veida pazīmi jūs atzīmējat.

Vērtības:

Piemēram, ja jūs vēlaties uzzināt, cik daudz ragu Ricca kartītēm ir kopā uz kartes, jums tās būtu jāskaita uz attēliem atsevišķi. Citādāk ir ar acīm, kurām jūs varat saskaitīt tieši zem tām uzrakstītos skaitļus.

Atkarībā no tā kā mēs datorā uzglabājam datus, tas var veikt dažādus aprēķinus (vai darbības). Pirms mēs saglabājam datus datorā, mums ir jāpadomā, ko mēs vēlamies ar tiem darīt. Tad mēs varam izlemt, kādu datu tipu izmantot. Izvēlētais datu tips nosaka arī to, kādas vērtības tiek pieņemtas un kādas darbības var veikt ar šādiem datiem.

Bumbiņas

Bulgārija

Ir dotas sarkanās un zilās bumbiņas noteiktā secībā:



Ja mēs skaitītu zilās bumbiņas no kreisās puses uz labo, sākot no pirmās bumbiņas un pēc tam sākot no otrās bumbiņas un tā tālāk, un mēs iegūtu sekojošu secību:

3, 3, 2, 1, 1, 1.

Ja mēs ierakstītu 0, kad skaitlis ir pāra, un ierakstītu 1, kad skaitlis ir nepāra, mēs iegūtu sekojošu virkni:

110111

Uzdevums

Ja mēs vēlētos iegūt nākamo virkni ar apzīmējumiem 0 un 1 (kas apzīmē pāra un nepāra skaitļus):

01110100

Nosakiet, kuras krāsu bumbiņu virkne ir iegūta no iepriekš minētās skaitļu virknes:

Izvēlieties bumbiņas un novietojat tās pareizajā secībā zem pelēkā laukuma.

Skaidrojums

Pareizā atbilde ir:



Ir vairāki veidi kā nonākt līdz pareizās atbildes. Viens veids ir apskatīties bumbiņu rindu un loģiski noteikt kura ir zila un sarkana.

Apskatīsim bumbiņu rindu:

1. No sākuma apskatīsim pirmo bumbiņu no labās puses. Mēs pieņemam, ka bumbiņa ir sarkana, jo pirmais cipars ir 0 un tas ir nepieciešams, lai visu pārējo zilo bumbiņu skaits būtu 0.

0	1	1	1	0	1	0	0
							sarkana

2. Nākamajai bumbiņai no labās arī jābūt sarkanai, jo tikai tā zilo bumbiņu summa ir pareiza

0	1	1	1	0	1	0	0
						sarkana	sarkana

3. Nākamajai bumbiņai ir jābūt zilai, jo nākamais skaitlis ir 1 un tas nozīmē ka zillo bumbiņu summai ir jābūt nepāra skaitlim. Tā kā mēs zinām, ka pirmās divas bumbiņas ir sarkanas, 3. bumbiņai jābūt zilai, lai mēs mainītu zilo bumbiņu skaitu no pāra uz nepāra skaitu.

0	1	1	1	0	1	0	0
					zila	sarkana	sarkana

4. Nākamais skaitlis no labās puses ir 0, kuru iespējams iegūt tikai, ja 4. bumbiņa ir zila.

0	1	1	1	0	1	0	0
				zila	zila	sarkana	sarkana

5. Pēc tam seko skaitlis 1, tāpēc arī nākamajai bumbiņai ir jābūt zilai, lai mainītu summu no pāra uz nepāra.

0	1	1	1	0	1	0	0
			zila	zila	zila	sarkana	sarkana

Tas nozīmē, ka katru reizi kad skaitlis mainās no 0 uz 1 vai no 1 uz 0, ir jābūt zilai bumbiņai, kamēr, ja skaitlis nemainās, ir jāizmanto sarkanā bumbiņa.

0	1	1	1	0	1	0	0
4 zila	3 sarkana	3 sarkana	3 zila	2 zila	1 zila	0 sarkana	0 sarkana

Tādā veidā zilo bumbiņu secībai ir jābūt: 43332100.

Šī ir algoritmiskā domāšana

Šajā uzdevumā jūs trenējat savu algoritmisko domāšanu. Cerams jūs atklājāt, ka varat atjaunot kodu, ejot atpakaļ (kā parādīts atbildes izskaidrojuma daļā) – un tā ir algoritmiskā domāšana vislabākajā izpausmē. Datorzinātnieki bieži strādā ar dažāda veida algoritmiem, un viņiem jāizprot, kā tos izmantot un kā tie darbojas.

Virpuļojošās lapiņas

Īrija

Lauris nometa savu mīļāko lapiņu uz zemes un tagad tā ir pazudusi starp citām lapiņām

Šādi izskatās Laura lapiņa:



Jautājums

Kura ir Laura lapiņa?

Lapiņas var pagriezt visos virzienos. Nospiediet uz tās lapiņas, kura jums šķiet pareiza



Skaidrojums



Šī ir pareizā atbilde, jo dzeltanā daļa un brūnais pleķis uz kāta, kā arī kodums ir pareizajās vietās uz lapiņas



Nav pareiza, jo nav dzeltanā plankuma



Nav pareiza, jo nav brūnā pleķīša



Nav pareiza, jo lapiņai ir kodiens nepareizajā vietā

Šī ir informātika

Mūsdienās informācija bieži tiek meklēta ar datorprogrammām, kas meklē paternus (notikumus, kas atkārtojas). Identificējot paternus, informāciju var filtrēt, analizēt un labāk izprast. Noņemot tās detaļas, kas neatbilst pareizajam paternam, bieži vien varam iegūt labākus rezultātus.

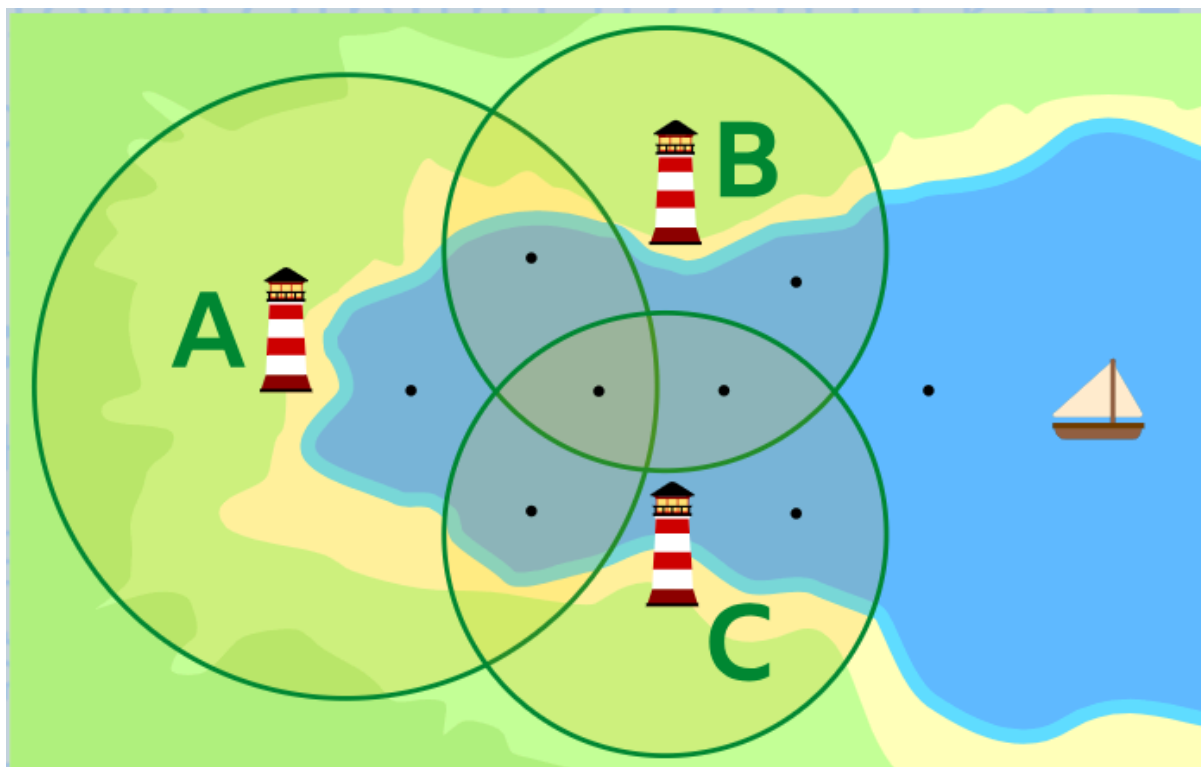
Paterni var būt sastopami jebkura veida datos, gan dabā, gan laika grafikā, kopijās. Tomēr dažreiz tie ir apslēpti (piemēram, apgriezti). Paterni var rasties arī, ja šķietami dažādi vienumi tiek analizēti. Šajā uzdevumā paterni ir nedaudz apslēpti rotācijas dēļ, lai ilustrētu, ka dati var būt jāapstrādā, lai atklātu meklēto paternu.

Abstrakcija. Šis uzdevums ilustrē līdzīgu elementu izslēgšanas konceptu jeb abstrakciju. Katru lapiņas elementu (piemēram, nelielu asaru pretī koduma zīmei) var izmantot par pazīmi, lai atrastu pareizo atbildi vairāku atbilžu sarakstā un nonāktu līdz pareizajai atbildei vai vismaz izslēgtu vienu no atbildēm no pareizo atbilžu saraksta.

Bākas

Čehija

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



Jautājums

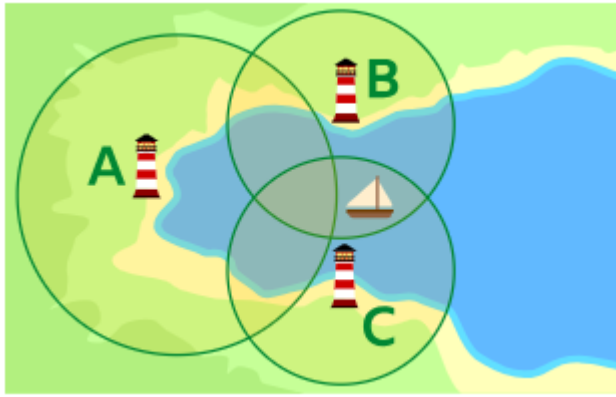
Kapteinis Bens var redzēt B un C bākas. Tomēr viņš nevar redzēt bāku A.

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 apli 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:

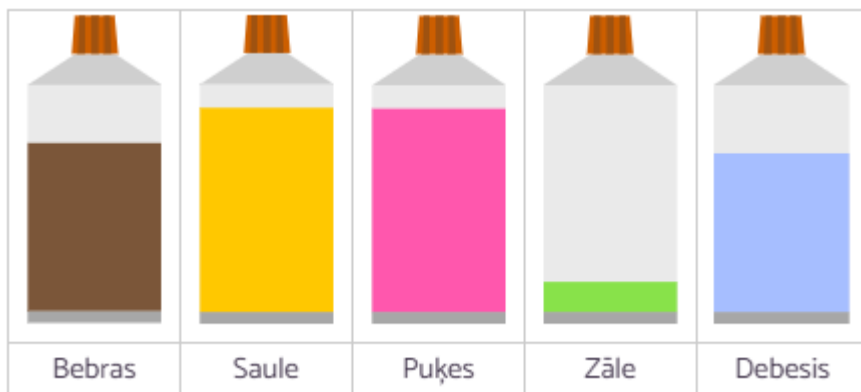
"Picas ballītes uzdevumu" vairāku iemeslu dēļ var uzskatīt par tādu informātikas uzdevumu, kas uzsver koncentrēšanos uz informācijas apstrādi, pārvaldību un analīzi (draugu iecienītākās picas piedevas), lai iegūtu vēlamu informāciju (vēlamākās picas piedevas) un pieņemtu lēmumus, lai atrisinātu uzdevumu, 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

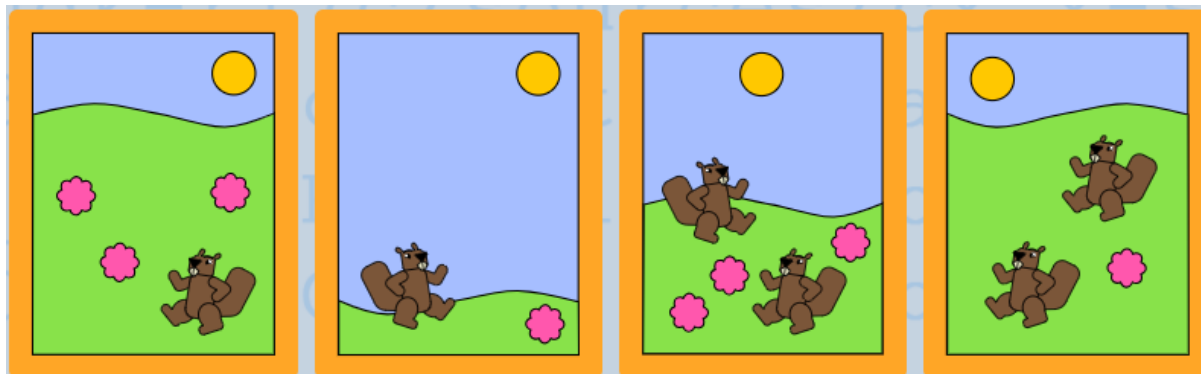
Somija

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?

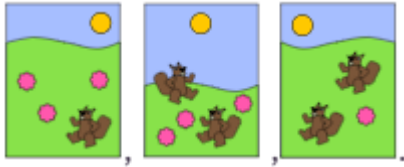


Skaidrojums



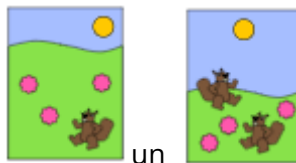
Pareizā atbilde ir:

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 tādas 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:

Pagājušajā svētdiena bija saulaina un

visi dīķi bija pilni ar peldošiem bebrim

dīķī ar ledaino ūdenskritumu nepeldēja neviens bebrs

bebrs Maikls peldēja visos dīķos

bebrs Maikls nepeldēja nevienā dīķī

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 bebrim. 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 bebrim.*" Tas nozīmē, ka katrā dīķī ir bebri, kas sakrīt ar Toma teikto.

Tā ir informātika:

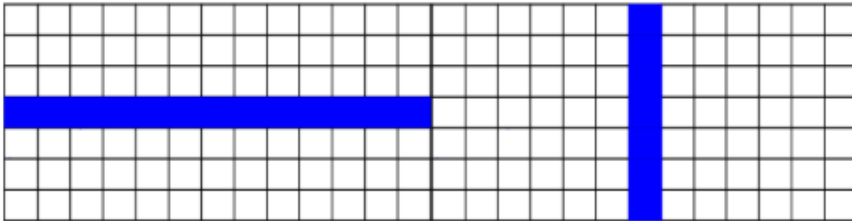
Toms un Kate apspriež Toma apgalvojuma patiesumu, izmantojot Būla loģikas teoriju, kur apgalvojumi tiek divvērtību veidā klasificēti kā patiesi vai nepatiesi. Šī teorija, ko dēvē par "loģiku", ietver sarežģītu apgalvojumu konstruēšanu, izmantojot tādus apzīmējumus kā "un"(AND), "vai"(OR), negācija (NOT), 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".

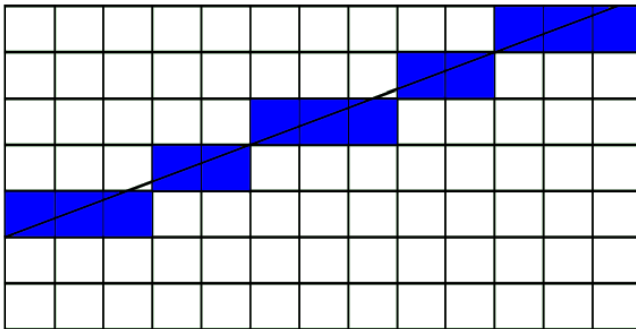
Līniju zīmēšana

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.

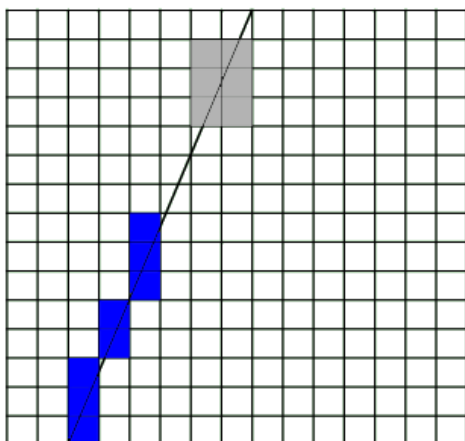


Diagonālās līnijas ir jāzīmē, izmantojot horizontālo un vertikālo pikseļu kombināciju, tāpēc tās nebūs precīzas. Šeit varat redzēt pikseļu zīmējumu, kas attēlo diagonālās līnijas daļas. Šis modelis ir horizontālo (vai vertikālo) pikseļu savienojums, kas atkārtojas pa diagonālu līniju.

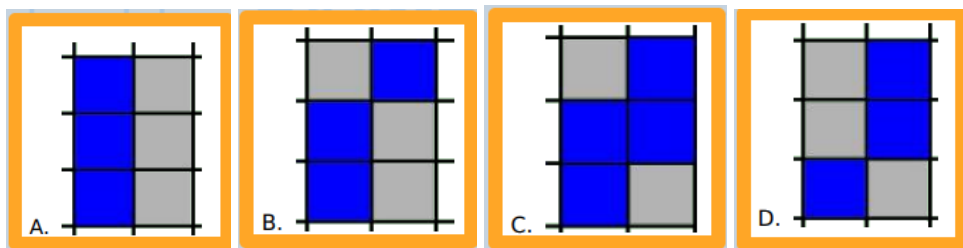


Jautājums

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

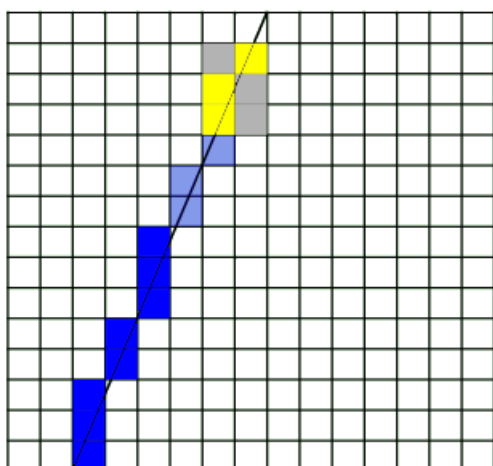


Atbilžu varianti



Skaidrojums

Pareizā atbilde ir B, kā parādīts diagrammā zemāk, kur trūkstošie pikseli 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 pikseli, daudzas grafikas šķita ļoti raupjas un nereālistiskas. Lai palīdzētu izlīdzināt attēlu malas, tika izgudrotas tādas metodes kā robojumu izlīdzināšana.

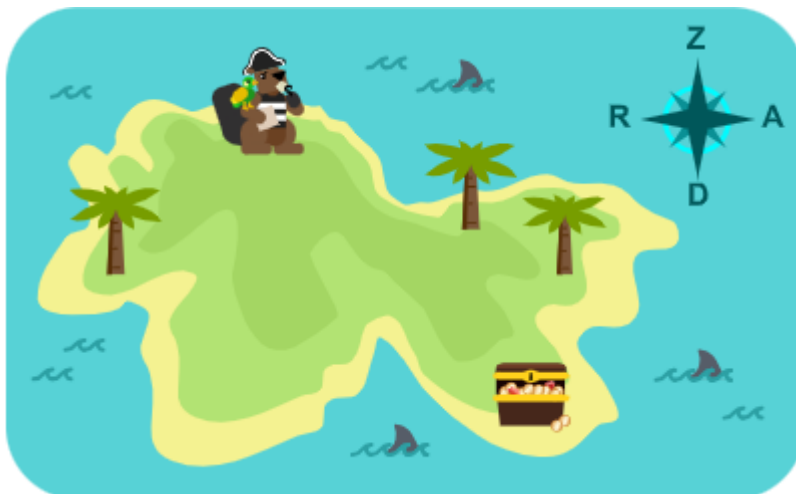
Šajā uzdevumā tiek demonstrēts viens vienkāršs līniju zīmēšanas algoritms. Līniju, kas nav vertikāla vai horizontāla, var attēlot kā pikseļu virkni. Šis algoritms izmanto līnijas slīpumu, lai noteiktu, kuriem pikseliem jābūt ieskrāsotiem, bet kuriem - nē. Pamatalgoritms, kas tika izmantots šajā uzdevumā,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

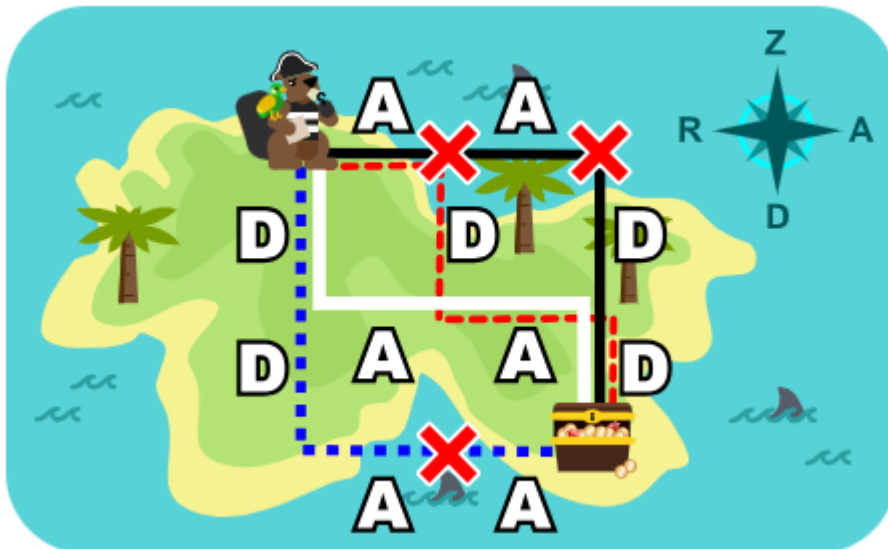
A, A, D, D

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 šķautnes ar papildu ierobežojumu, ka šķautnes ir vērstas tikai horizontālā vai vertikālā virzienā. Pārvietošanās dotajā instrukcijā atbilst kustībai grafā no vienas virsotnes uz otru pa tās savienjošo 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 nezīnām, 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

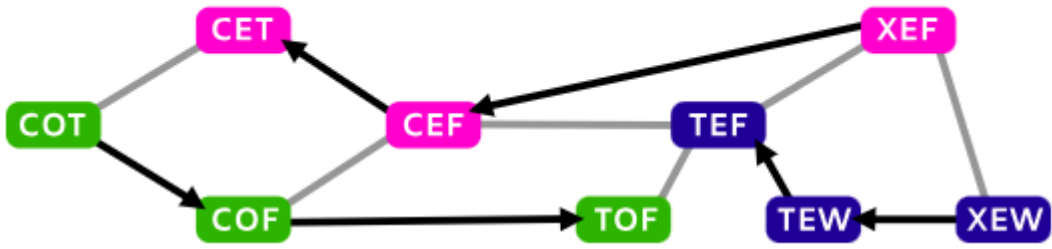
Skaidrojums

Pareizā atbilde ir TEF → CEF → COF.

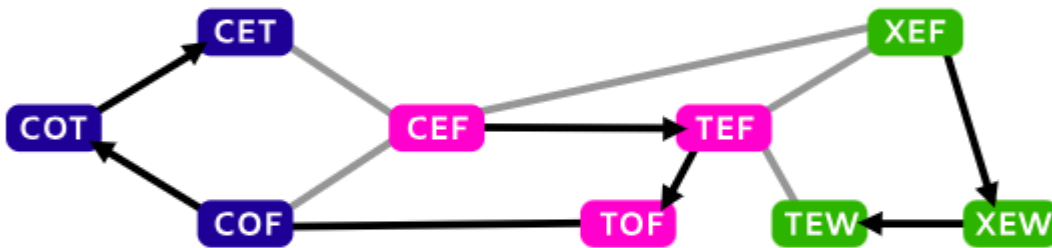
Lai atrisinātu šo uzdevumu, ir lietderīgi uzzīmēt diagrammu. Šajā diagrammā divi vārdi ir savienoti ar līniju, ja tie spēj sekot viens otram vārdu ķēdē.



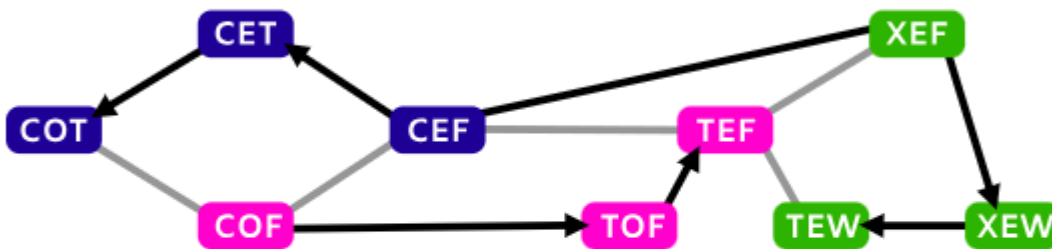
Variants XEW → TEW → TEF ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt XEF → CEF → CET un COT → COF → TOF, kā parādīts diagrammā.



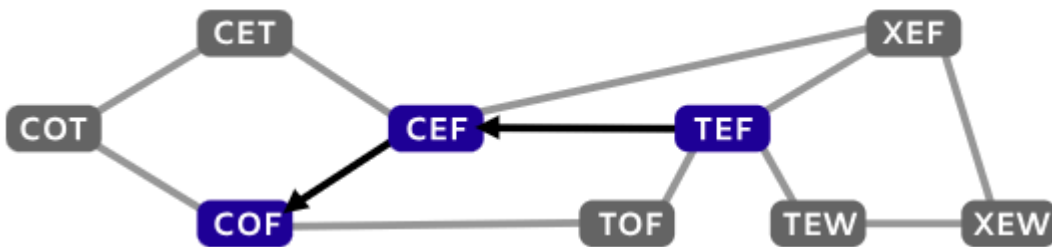
Variants $\text{COF} \rightarrow \text{COT} \rightarrow \text{CET}$ ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt $\text{CEF} \rightarrow \text{TEF} \rightarrow \text{TOF}$ un $\text{XEF} \rightarrow \text{XEW} \rightarrow \text{TEW}$, kā parādīts diagrammā.



Variants $\text{CEF} \rightarrow \text{CET} \rightarrow \text{COT}$ ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt $\text{COF} \rightarrow \text{TOF} \rightarrow \text{TEF}$ un $\text{XEF} \rightarrow \text{XEW} \rightarrow \text{TEW}$, kā parādīts diagrammā.



Variants $\text{TEF} \rightarrow \text{CEF} \rightarrow \text{COF}$ 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 atbilstoši 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.

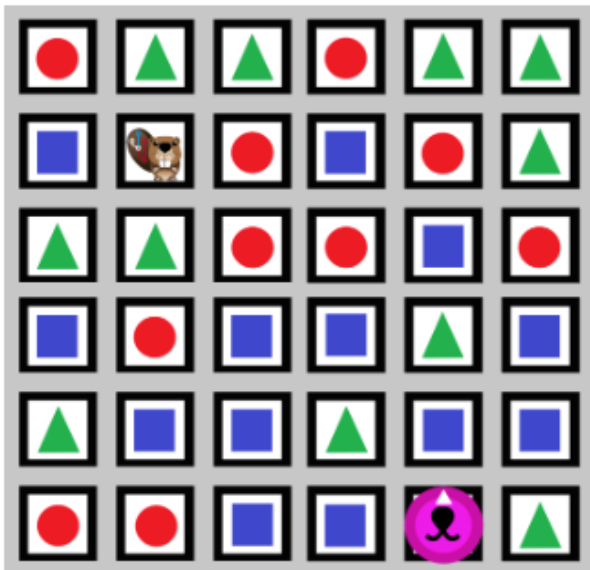
Robots pārvietojas ņemot vērā krāsainus simbolus un to nozīmi:

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

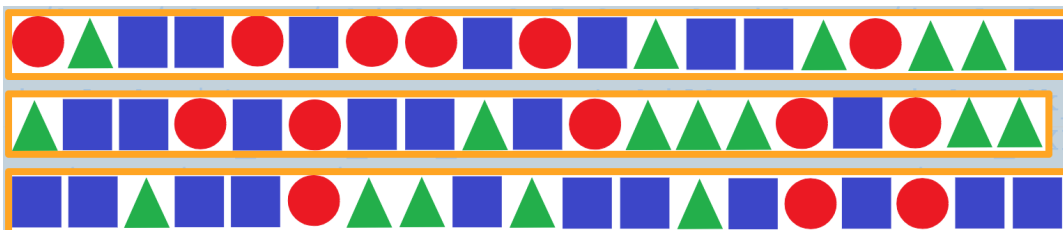
Piemērā robots pārvietojas pa šādu ceļu. Pirmajā solī tas pagriežas pa kreisi un virzās uz priekšu, tad tas pagriežas pa labi un virzās uz priekšu un beigās tas vienkārši virzās uz priekšu. Bultas parāda maršrutu, pa kuru robots pārvietojas.

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 nerasnīgs 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 aprobe

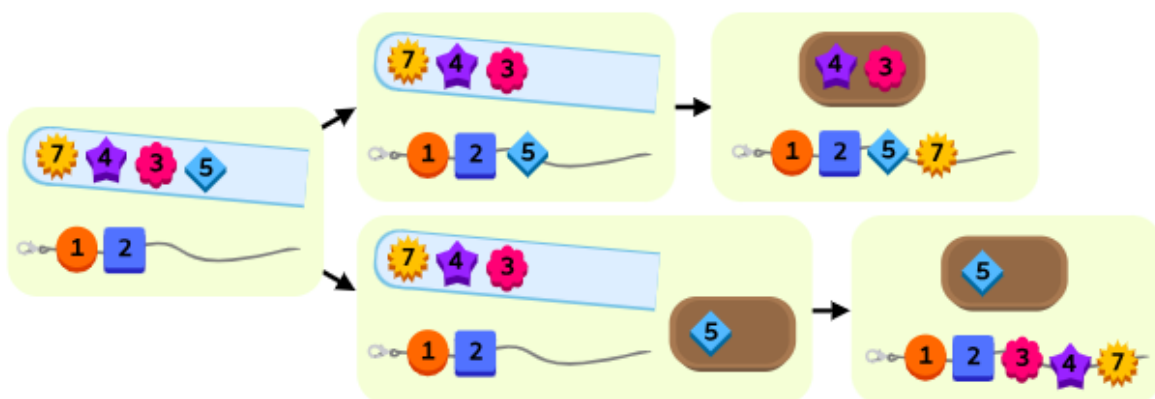
Taivāna

Jānis veido aproci. Viņš no stikla tūbiņas ņem pērles uz kurām ir uzrakstīti cipari. Viņš 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.

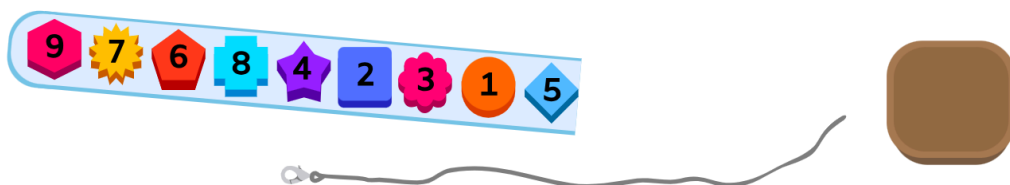
Šajā piemērā pēdējā pērle uz auklas ir 2. pērle. Tālāk Jānis drīkst uzvilkt pērli Nr. 5 no stikla tūbas. Viņš to var arī atlikt malā un neizmantot.

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:

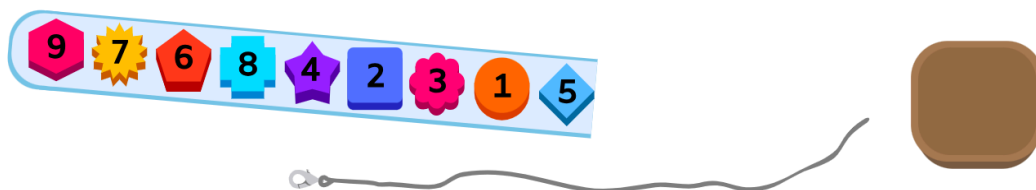


Kāds ir lielākais daudzums pērļu, kuras viņš var uzvilkt uz aproces?

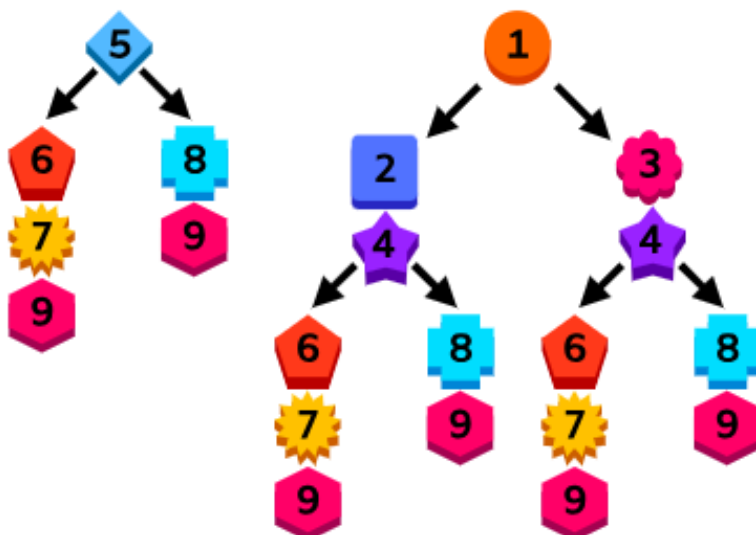
Skaidrojums

Pareizā atbilde ir 6.

Mēs varētu apskatīt visas aprocas, 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 uzvilkota 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 iespējamās 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 aprocas - 24679 or 2489, kurām seko šādas aprocas - 124679 un 12489/

Aproces ar vislielāko skaitu pērļu sākas ar Nr. 1 pērli un sastāv no 6 pērlītem: 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 iespējamus varinātus. Piemēram, ja arocē 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 pērlītes.

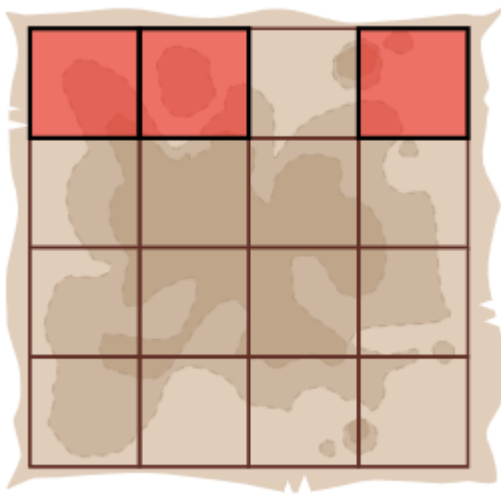
Atrodi dārgumus

Ungārija

Pirāts Pips meklē salā paslēptus dārgumus.

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

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



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

Jautājums

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

Skaidrojums

Pareiza atbilde ir 4.

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

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

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

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

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

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

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

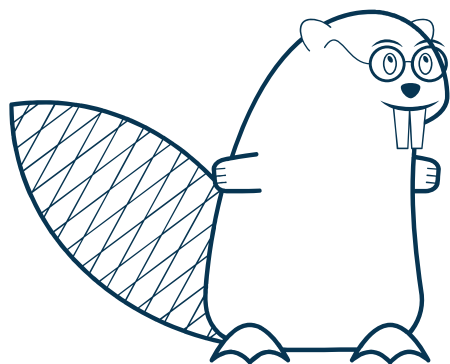
Pips nevar izmantot mazāk jautājumu. Ja viņš nesadala atlikušos kvadrātiņus divās vienādās daļās, tad dārgums varētu atrasties lielākajā daļā no tiem. Lai pajautātu par šo daļu, Pipam būtu jāuzdod tik daudz jautājumu, cik par vienu pusi.

Šī ir informātika

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

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

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



copyright Bebras