VISMA

Bebr[a]s '24

2024. gada 1. kārtas uzdevumi ar atbildēm

7.-8. klase

INFORMATĪVIE ATBALSTĪTĀJI

Izglītības un zinātnes ministrija

start(it)
www.startit.lv

Saturs

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.

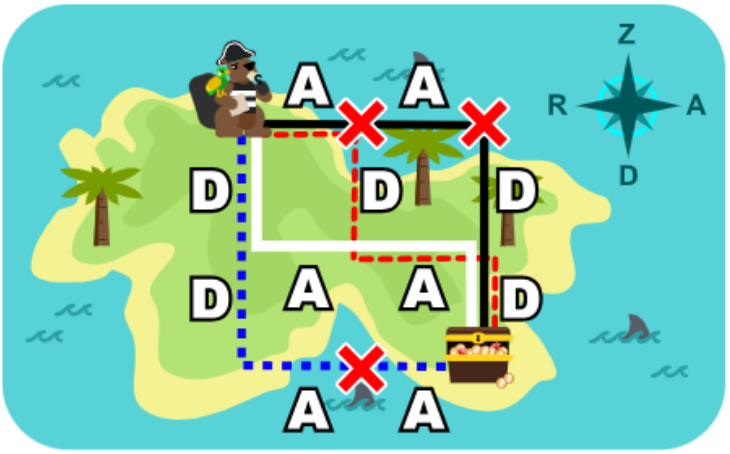
Jautājums

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?

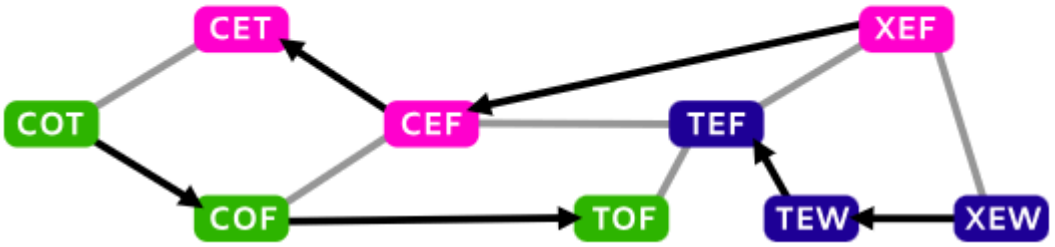
XEW → TEW → TEF | COF → COT → CET | TEF → CEF → COF | CEF → CET → COT

Skaidrojums

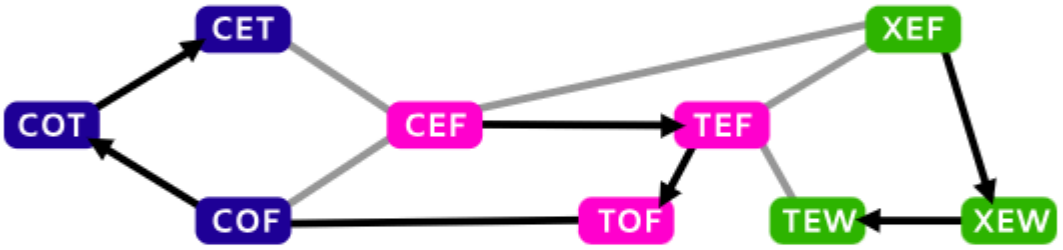
Pareizā atbilde ir TEF → CEF → COF.

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

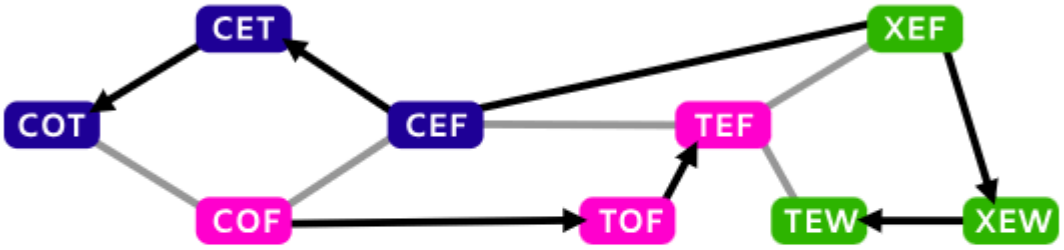
Variants XEW → TEW → TEF ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt XEF → CEF → CET un COT → COF → TOF, kā parādīts diagrammā.



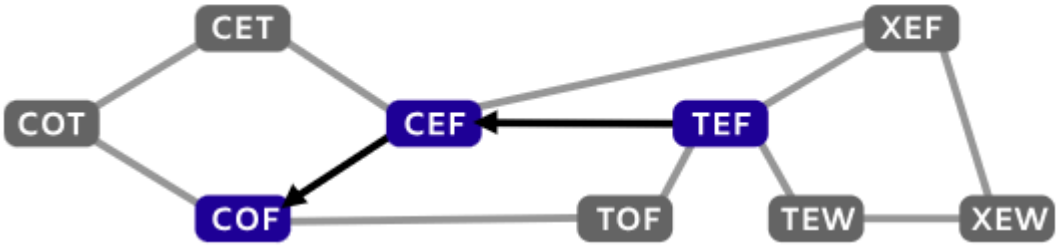
Variants COF → COT → CET ir iespējama kā vārdu ķēde. Šajā gadījumā pārējās divas ķēdes varētu būt CEF → TEF → TOF un XEF → XEW → TEW, kā parādīts diagrammā.



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

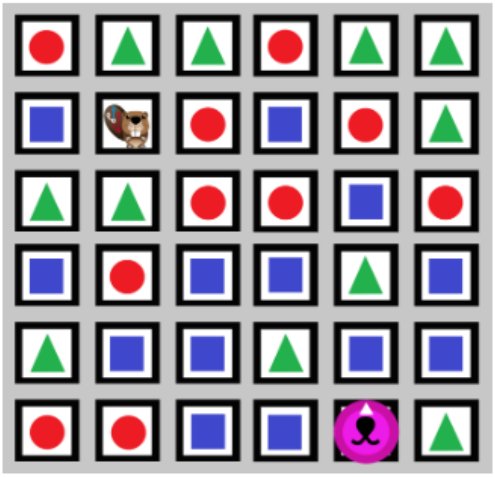
Robots  atrodas sākuma pozīcijā un vēlas sasniegt galamērķi. Tas spēj atpazīt krāsainus simbolus, uz kuriem tas nonāk. Tomēr, kad robots ir sākuma pozīcijā, mēs nevaram redzēt, uz kura simbola viņš stāv.
Robots pārvietojas ņemot vērā krāsainus simbolus un to nozīmi:

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

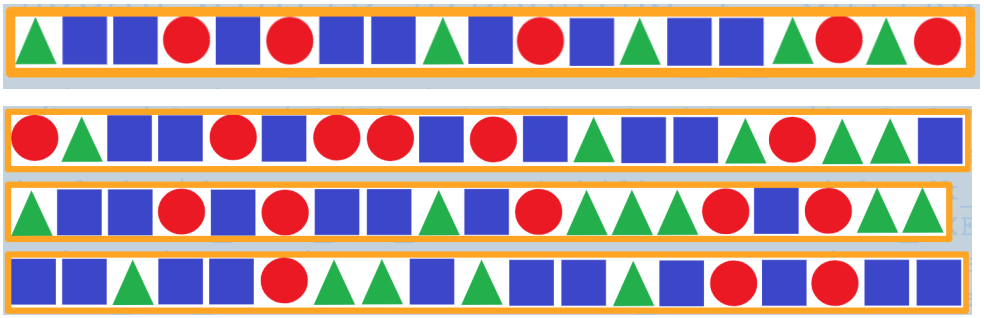
Piemērā robots pārvietojas pa šādu ceļu. Pirmajā solī tas pagriežas pa kreisi un virzās uz priekšu, tad tas pagriežas pa labi un virzās uz priekšu un beigās tas vienkārši virzās uz priekšu. Bultas parāda maršrutu, pa kuru robots pārvietojas.

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.

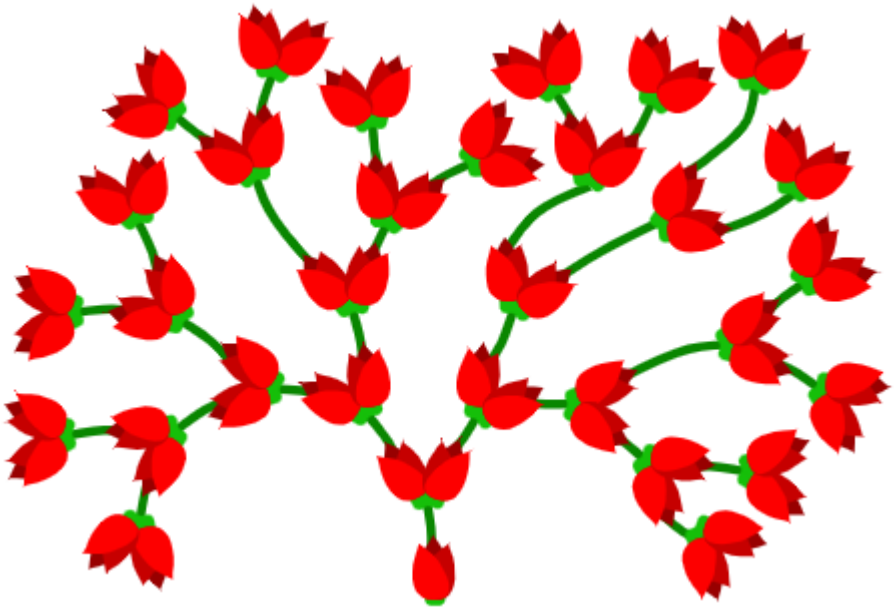
Brīnumpuķe

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

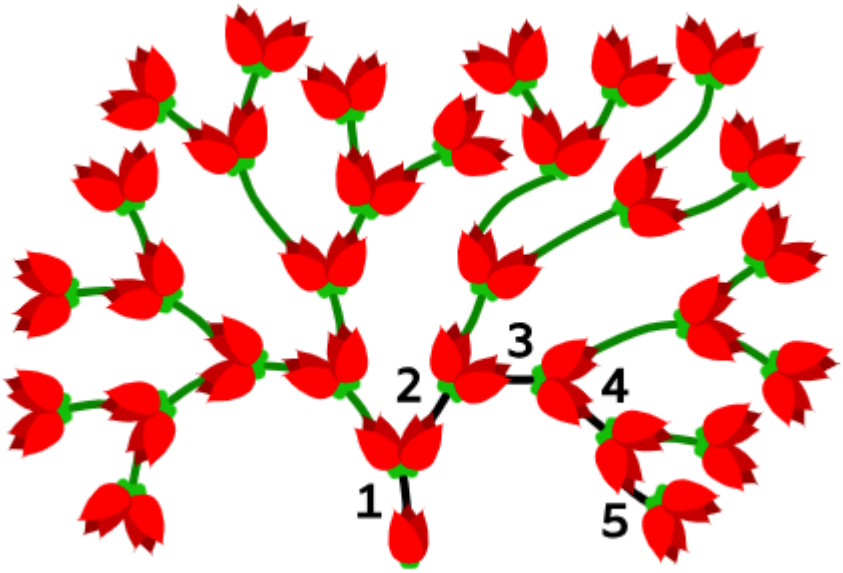
Cik dienas brīnumpuķe ir augusi?

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.

Visgarākā aproce

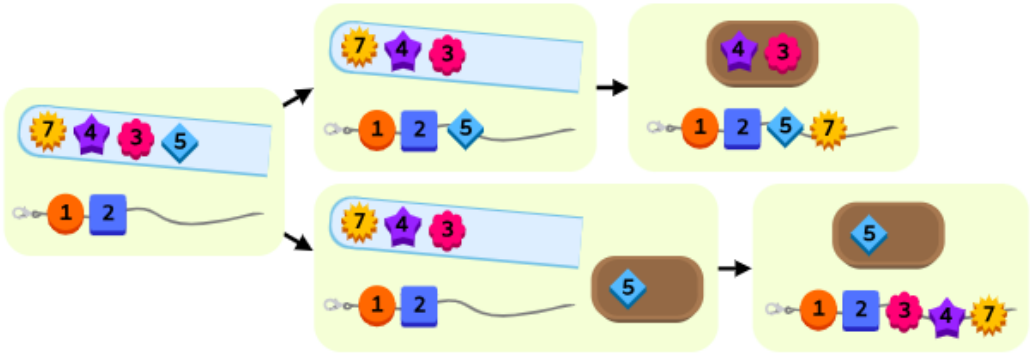
Taivāna

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.

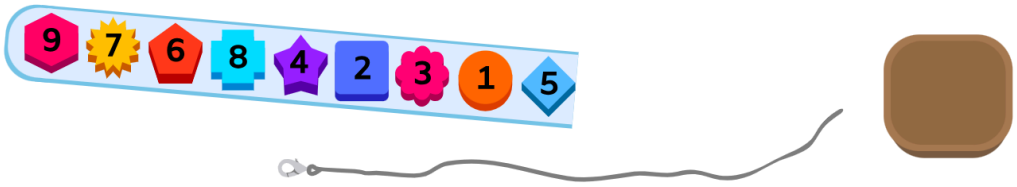
Šajā piemērā pēdējā pērle uz auklas ir 2. pērle. Tālāk Jānis drīkst uzvilkt pērli Nr. 5 no stikla tūbas. Viņš to var arī atlikt malā un neizmantot.

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:

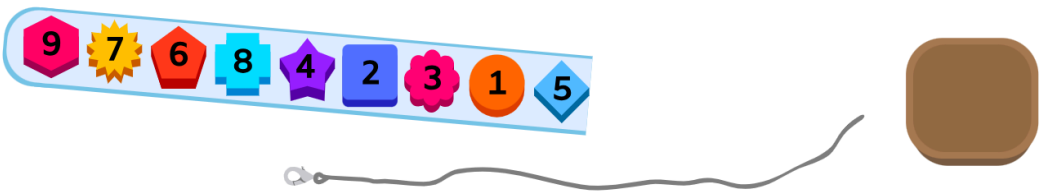


Kāds ir lielākais daudzums pērļu, kuras viņš var uzvilkt uz aproces?

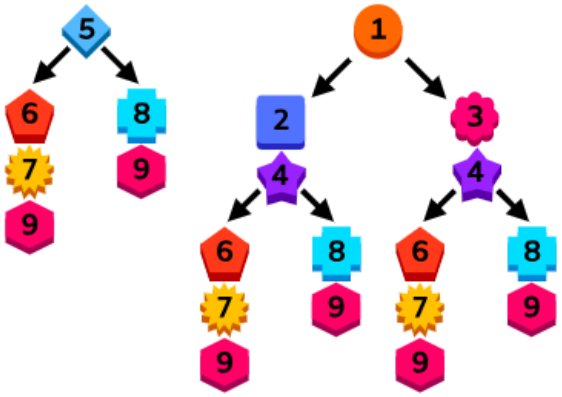
Skaidrojums

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.

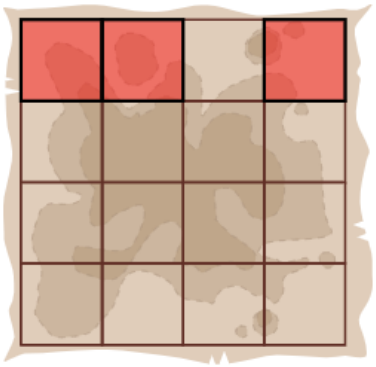
Atrodi dārgumus

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:

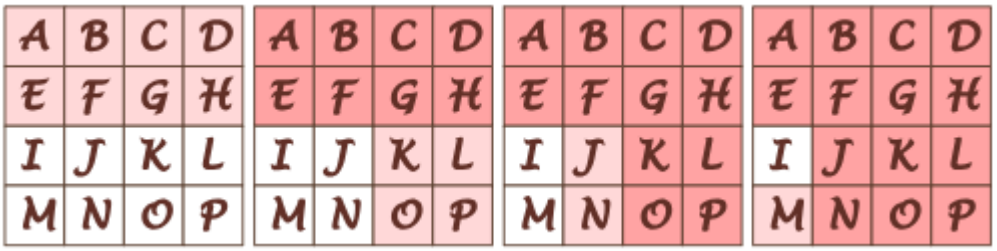
A. Ja ierīce norāda, ka dārgums atrodas norādītajos reģionos, Pips sadala reģionus uz pusēm un ierīcē ievada vienu pusi.

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.

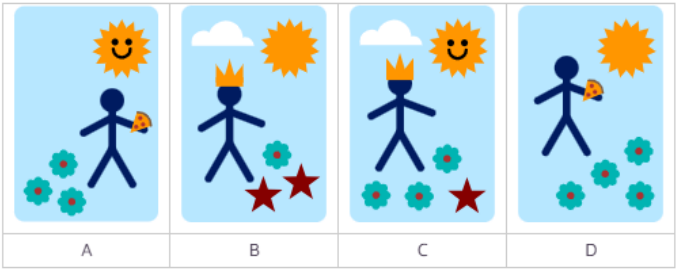
Ja ir 1 000 objektu, ir nepieciešami 10 meklēšanas soļi 2^9<1000<2^10, bet 1 000 000 objektu gadījumā - 20 soļi. Vispārīgi var teikt, ka n objektu meklēšanai ir nepieciešami aptuveni log(n) soļi; logaritma funkcija ir logaritms pie bāzes 2. Tā kā binārā meklēšana ir tik ātra, to bieži datorprogrammās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

Šajā uzdevumā meklēšanas telpa ir 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.

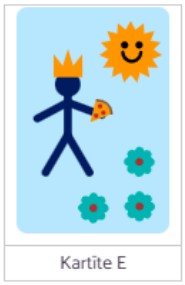
Kartītes

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

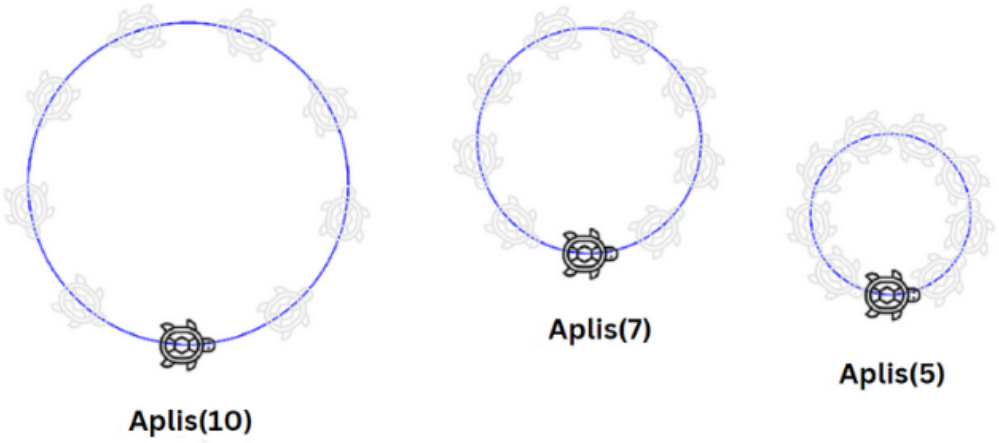
Šis uzdevums ilustrē loģikas pamatus, ko izmanto informātikā, lai mācītu, kā var attēlot un apstrādāt lēmumus un novērojumus. Datori ievēro skaidrus noteikumus, lai noteiktu rezultātus, līdzīgi tam, kā mēs ikdienā novērojam likumsakarības un izdarām izvēli. Tie izmanto pamata loģiku, lai apstrādātu nosacījumus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

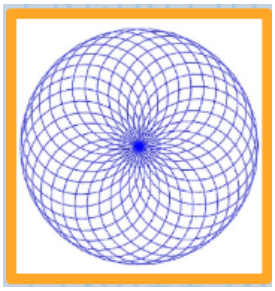
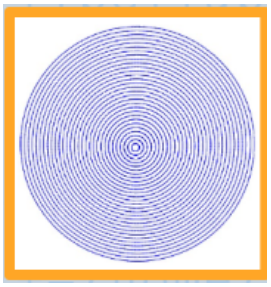
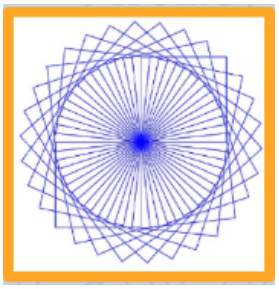
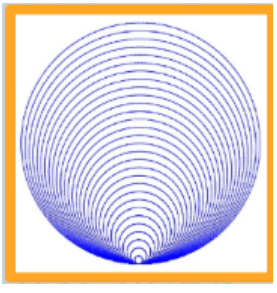
Robots, kurš ir bruņurupuča formā, uz zīmēšanas dēļa izveido zīmējumu atbilstoši instrukcijai. Kad tas saņem komandu zīmēt apli: "aplis(d)", tas uzzīmēs apli, kura izmērs mainīsies atkarībā no parametra d. Mainot parametru d, var tikt uzzīmēti dažāda izmēra apļi (skat. attēlu).



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) 

Skaidrojums

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.

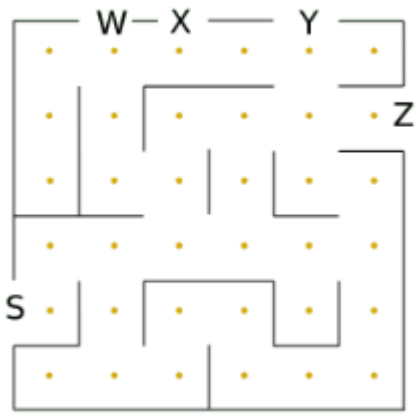
Šī ir algoritmiskā domāšana:

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:

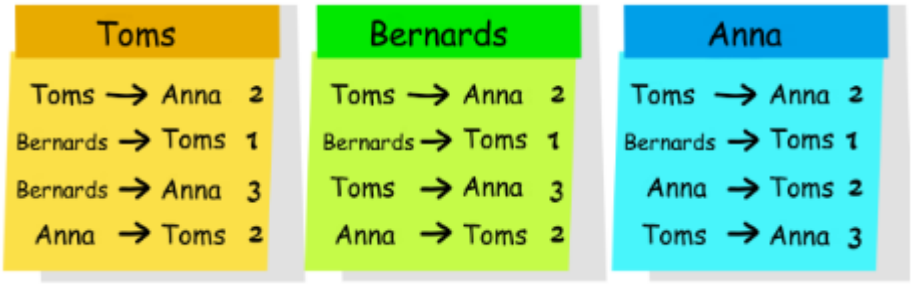
Šis ir strukturēts grafiks, kas sastāv no virsotnēm, kuras savienotas ar malām. Daudzas reālās pasaules problēmas var pārveidot grafikos.
Algoritms “Breadth-first search” (BFS) kopā ar “Depth-first search” ir viens no pamatalgoritmiem, ko izmanto, lai efektīvi un bez liekām darbībām sasniegtu visas grafika 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

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.

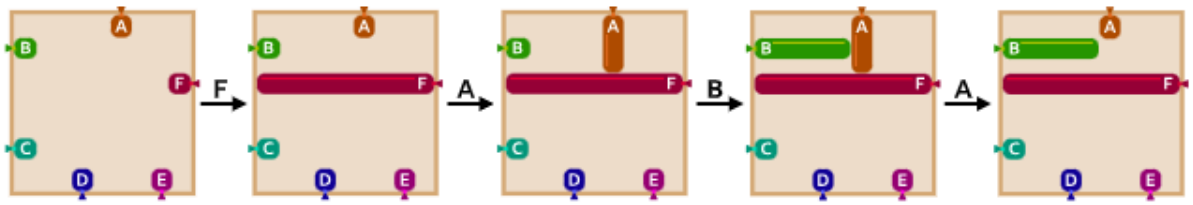
Balonu mašīna

Vācija

Bebriem ir mašīna, kas var radīt attēlus, piepūšot balonus kvadrātveida formā. Uz baloniem ir uzrakstīti burti A, B, C, D, E un F. Mašīna nolasa burtus pa vienam. Kad tā nolasa burtu:

1. Ja balons, kas apzīmēts ar šo burtu, ir tukšs, tas tiek piepūsts, līdz tas pieskaras citam balonam vai formas pretējai malai.
2. Pretējā gadījumā balons, kas apzīmēts ar šo burtu, tiek izpūsts.

Piemēram, ja sākumā visi baloni ir izpūsti un mašīna nolasa F, A, B un pēc tam A, tā rīkojas šādi:



Uzdevums

No sākuma katrs balons tiek izpūsts, un tad mašīna nolasa deviņu burtu secību. Kā rezultāts tiek iegūts turpmāk redzamais attēls. Ievadiet šo deviņu burtu secību.

a) BEBCACBDB
b) BECBACBDB
c) BEBCABCDB
d) BECBABCDB

Skaidrojums

Iespējamas 4 pareizās atbildes ar deviņiem burtiem:

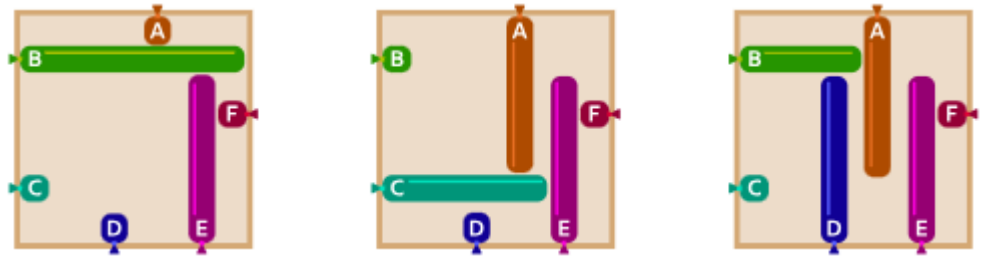
- B E B C A C B D B
- B E C B A C B D B
- B E B C A B C D B
- B E C B A B C D B

Mēs varam aplūkot detalizētu skaidrojumu vienai no pareizajām atbildēm: B E B C A C B D B. Attēlos zemāk tiek parādīts šis uzdevums, kurš ir sadalīts 3 daļās. Pirmajā attēlā redzams

stāvoklis pēc B E izpildes. Var redzēt, ka pirms E ir nepieciešams uzpūst B, lai B kļūtu par piemērotu šķērsli E uzpūšanai.

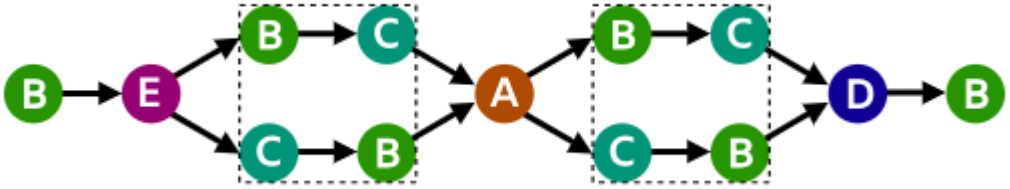
Otrajā attēlā parādīts stāvoklis pēc B E B C A. Balons C kalpo kā šķērslis A, tāpēc tas ir jāpiepūš pirms A. Turklāt balons B ir jāizpūš, pirms var piepūst A, citādi tas nesasniegs vajadzīgo garumu.

Trešajā attēlā redzams stāvoklis pēc B E B C A C B D. B kalpo kā šķērslis D, un C ir jāizpūš, pirms D tiek piepūsts.



Lai atrastu visas pareizās atbildes, ir lietderīgi problēmu attēlot, izmantojot virzienu grafiku. Burti attēlo balonus. Bultiņa no B uz E norāda, ka balons B ir jāpiepūš pirms balona E utt. Ir divi gadījumi, kad B un C secība var mainīties (norādīts ar pārtrauktu līniju).

Katra pareizā atbilde (piepūšamo balonu secība) ir šī grafika topoloģiskā secība. Šajā gadījumā ir četri topoloģiskie sakārtojumi, tāpēc ir četri iespējamie pareizie risinājumi.



Piepūstiem baloniem jāparādās nepāra reizes → A, D un E jāparādās nepāra reizes. Izpūstiem baloniem jāparādās pāra reizes (ieskaitot 0)→ B, C un F.

Šī ir informātika

Burtu secība ir datorprogramma, kas vada balonu mašīnu. Katrs burts strādā kā komanda, kas liek mašīnai piepūst balonu vai izlaist gaisu no tā. Tāpat kā lielākajā daļā šāda veida datorprogrammu, būtiska ir komandu secība. Piemēram, secība B E liek mašīnai izveidot citu attēlu nekā secība E B.

Atbildes skaidrojumā izmantotajam orientētajam grafam nav ciklu, tāpēc to sauc par orientētuaciklisku grafu (angliski - directed acyclic graph (DAG)). Jebkuram DAG ir vismaz viens topoloģiskais sakārtojums, un ir zināmi ātri algoritmi topoloģiskā sakārtojuma konstruēšanai. DAG ir daudz pielietojumu informātikā, piemēram, uzdevumu plānošanā.

Ēdiena karte

Itālija

Bebrs Niks ir paslēpis savas pārtikas krājumus zem 9 no 17 kokiem, kas ieskauj dīķi. Niks ir izveidojis karti un uzrakstījis uz kvadrātiem cik koku ar pārtikas krājumiem pieskaras šim kvadrātam.

Piemēram, kvadrāts uz kura ir uzrakstīts 3 nozīmē, ka pārtikas krājumi ir paslēpti tieši zem trim kokiem, kas pieskaras šim kvadrātam.

Tomēr šī ideja neliekas pareiza viņa draudzenei Bellai un viņai ir taisnība.
Viņa saka, ka Niks varēs noteikt tikai 7 no 9 kokiem, zem kuriem ir paslēpts ēdiens.

Uzdevums

Izvēlies šos 7 kokus

Skaidrojums

7 koki zem kuriem noteikti ir paslēpts ēdiens ir šie:

2 koki un viens tukšais (x) var būt identificēti, sākot analizēt no apakšas:

Skatoties no apakšējā labā stūra uz kreiso, zem pirmā koka noteikti ir jābūt ēdienam, lai sakristu ar zemāko skaitli 1. Arī zem otrā koka ir jābūt ēdienam, lai sakristu ar skaitli 2. Tomēr, ja ēdiens būtu zem apakšējā kreisā koka, tad nebūtu iespējams apmierināt skaitli 2, ko aptver tik mazs loks.

Tādā veidā ir iespējams noteikt visus kokus kreisajā pusē, zem kuriem atrodas ēdiens:

Labajā dīķa pusē ir iespējami divi varianti:

Tieši tāpēc ir iespējams pievienot tikai vienu sarkano zvaigzni.

Kā vēl labāk noformulēt šo problēmu? Kad mēs esam noteikuši apakšējos kvadrātus, zem kuriem atrodas ēdiens, mēs varam iedot nosakumus burtu veidā arī pārējiem kvadrātiem:

Cipari zilajos kvadrātos satur "ierobežotājus". Skatoties pulksteņrādītāja virzienā no apakšas, pirmie 3 noteikumi (1,1,2) ir izpilditi, bet, lai izpildītu 3. ierobežojumu, ir jāizvēlas vēl tieši viens elements no kopas {A, B, C}, un tā tālāk. Tad šo uzdevumu var pārveidot šādi:

No {A, B, C} jāizvēlas tieši viens elements,
No {B, C, D} izvēlēties tieši 2 elementus,
No {C, D, E, F, G} izvēlēties tieši 2 elementus,
Nav {F, G, H} izvēlas tieši 2 elementus,
Nē {G, H, I} izvēlas tieši vienu elementu,
Nav {H, I, J} izvēlas tieši vienu elementu,
Nav {I, J, K, L, M} izvēlas tieši 2 elementus,
Nē {L, M, N} izvēlas tieši 2 elementus,
Nē {M, N, O} izvēlas tieši 2 elementus,
Nē {N, O} izvēlas tieši vienu elementu.

Apskatot visus variantus (no pirmās kopas nevar izvēlēties A, jo tad no otrās kopas būtu jāizvēlas B vai C; ja no pirmās kopas izvēlas B, tad no otrās kopas jāizvēlas D utt.), var pārbaudīt, ka ir divi risinājumi:

{B, D, F, H, K, M, N}, {B, D, F, H, L, M, O}.

Elementi, kurus noteikti var atzīmēt ar sarkanu zvaigzni, ir B, D, F, H, M (kas veido abu iegūto kopu krustpunktu), kas saskan ar iepriekš iegūto.

Šī ir informātika

Šis uzdevums ir iedvesmots no slavenās loģikas puzles - videospēles, kura ir pazīstama ar dažādiem nosaukumiem (Minesweeper, Flower Field). Pat tad, kad ir atklātas plaši laukumi, var rasties situācijas, kad nav pieejams pietiekami daudz informācijas, lai droši turpinātu spēli, tāpēc dažreiz ir jāpaļaujas uz veiksmi. Tieši šāds gadījums rodas mūsu uzdevumā: dīķa labo pusi nevar atrisināt tikai balstoties uz doto informāciju!

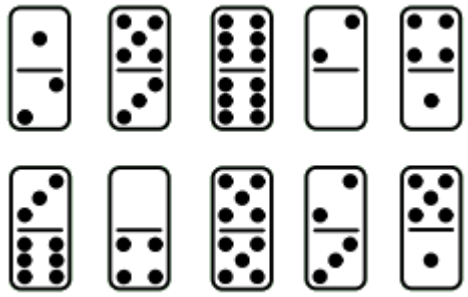
Ievērojiet, ka kopumā, ja ap dīķi ir n koku, tie atbilst 2^n iespējamām konfigurācijām, un tad parasti ir daudz vieglāk pārbaudīt, vai dotais risinājums vispār atbilst uzdevuma ierobežojumiem, nekā to atrast praktiski!

Skatoties uz šo uzdevumu mēs varam arī redzēt, cik bieži uzdevumus var risināt pakāpeniski t.i. atrisināt pa atsevišķām daļām. Nešaubigi interesantākais aspekts no informātikas viedokļa ir problēmas formalizācija, izmantojot atbilstošu datu struktūru (kā mēs to darījām šim uzdevumam skaidrojuma pēdējā daļā), kas norāda, ka šī problēma ir skaitliski sarežģīta, kā arī tai ir vairāki risinājumi.

Uzmini domino kauliņu

Somija

Alise un Bobs spēlē spēli. Viņiem uz galda ir 10 domino kauliņi:



Bobs izvēlas slepenu domino figūru, kas zināma tikai viņam. Pēc tam Alise var uzdot Bobam jautājumus “jā” vai “nē”, lai noskaidrotu, kuru domino kauliņu viņš izvēlējies. Katram jautājumam jābūt ar atbildi jā vai nē.

Alisei jautājumi jāveido tā, lai neatkarīgi no Boba atbildes viņai būtu iespēja pēc iespējas precīzāk noteikt, kurš ir slapenais domino kauliņš

Jautājums

Kuru jautājumu Alisei vajadzētu uzdot kā pirmo?

A. Vai punktiņu summa uz figūras ir lielāka vai vienāda ar 7?

B. Vai punktiņu skaits figūras galā, kurā ir lielāks punktu skaits, ir lielāks vai vienāds ar 4?

C. Vai punktiņu skaits figūras galā, kurā ir mazāks punktu skaits, ir lielāks vai vienāds ar 2?

D. Vai abos figūras galos ir vienāds punktu skaits?

Skaidrojums

Pareizā atbilde ir C, jo šis jautājums sadala kauliņu komplektu tieši divās grupās pa pieciem domino kauliņiem (pieciem domino kauliņiem viena puse it tāda, kas lielāka vai vienāda ar 2, tāpēc pārējiem pieciem ir viena puse tāda, kas ir ar mazāk nekā 2 punktiem.), tāpēc

neatkarīgi no Boba atbildes viņam paliks piecas vēl derīgas domino kauliņu kandidāti. Šādi rīkojoties, Alise katru reizi izslēdz pusi no iespējām neatkarīgi no Boba atbildes.

Citos variantos kauliņu grupas nav sadalītas tik vienmērīgi, un viena no iespējamām atbildēm atstās vairāk nekā piecus domino kauliņus kā atlikušos kandidātus.

Nākamajā attēlā katram iespējamajam jautājumam parādīts, uz kuru domino kauliņu kopu norādītu "jā" (pelēkā krāsā) vai "nē" atbildi.

Tā ir informātika:

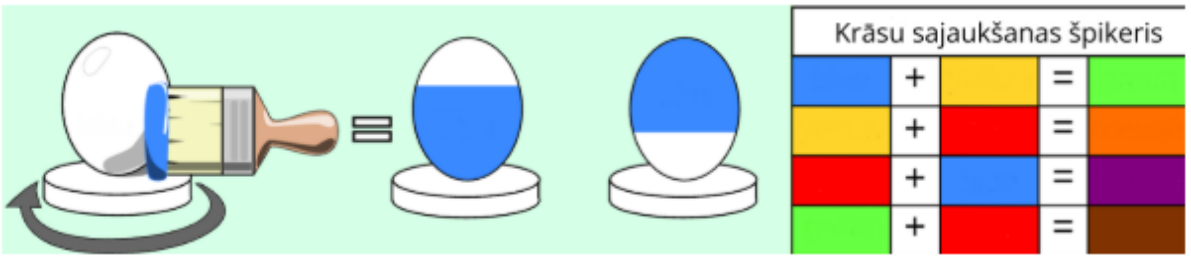
Datori glabā daudz datu, un tiem ir vajadzīgi ātri paņēmieni, kā atrast konkrētu informāciju. Tā vietā, lai aplūkotu katru datu vienību atsevišķi, mēs varam izmantot gudrākas metodes, ja datus labi sakārtojam. Piemēram, ja dati ir sakārtoti, mēs varam izmantot metodi, ko sauc par bināro meklēšanu. Tā darbojas šādi: mēs sākam ar vidējā elementa pārbaudi. Ja tas, ko meklējam, ir lielāks vai mazāks par meklēto, mēs varam ignorēt pusi datu un pievērsties tikai pārējiem. Mēs turpinām to darīt, katru reizi samazinot datu apjomu uz pusi. Tātad, ja mums ir 1 miljons elementu, mēs varam atrast meklēto tikai 20 soļos!

Šajā uzdevumā mums ir tikai jā/nē tipa jautājumi, tāpēc labākais pirmais uzdotais jautājums sadalītu iespējas divās daļās, līdzīgi kā pirmais solis binārajā meklēšanā. Ja mums būtu nepieciešams turpināt uzdot jautājumus, lai atrastu slepeno domino, mēs varētu izveidot visu iespējamo jautājumu plānu, kas līdzinās lēmumu koka izveidei.

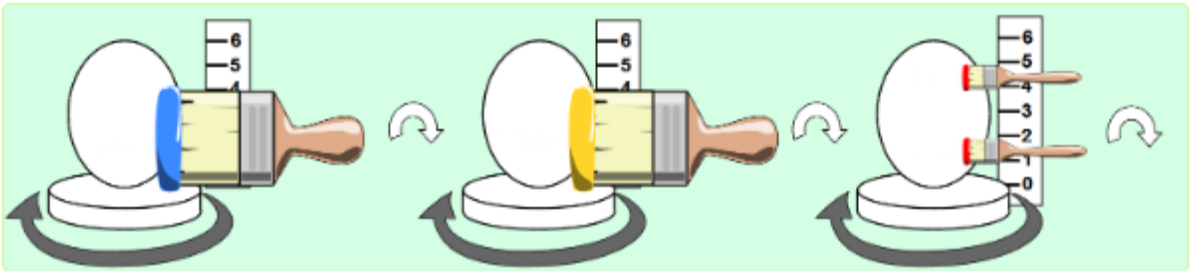
Olu krāsošana

Īrija

Aija grib nokrāsot olas Lieldienām. Kad viņa uzliek baltu olu uz krāsojamā aparāta, kas griežas ap savu asi un tur otas galu pie olas, olas tiek nokrāsotas ļoti ātri. Viņa parasti neizkustina otu, kamēr ola griežas aparātā, taču vienmēr apgriež olu otrādi pēc katras krāsas uzklāšanas, kā parādīts zemāk esošajā attēlā. Krāsas sajaucas kopā, kad tās pārklājas. Zemāk esošajā tabulā ir parādīti krāsu saikumi pēc divu dažādu krāsu sajaukšanas.

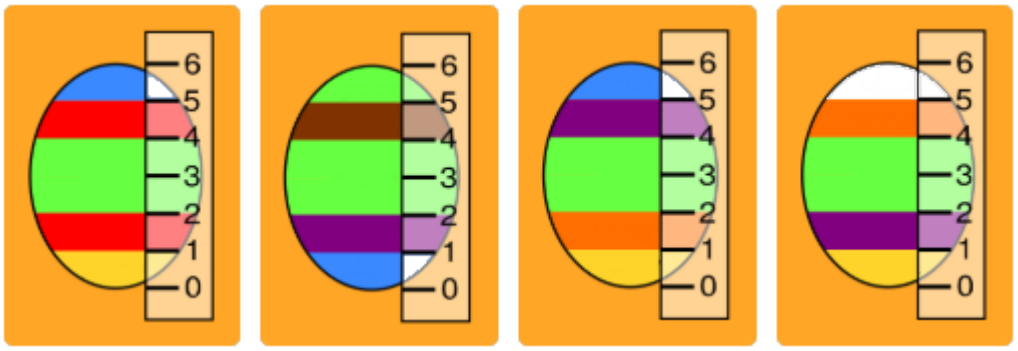


Aija nokrāso savu balto olu, izmantojot krāsu secību un otas platumu, kā parādīts nākamajā attēlā.



Jautājums

Kāda būs Aijas ola pēc krāsošanas?

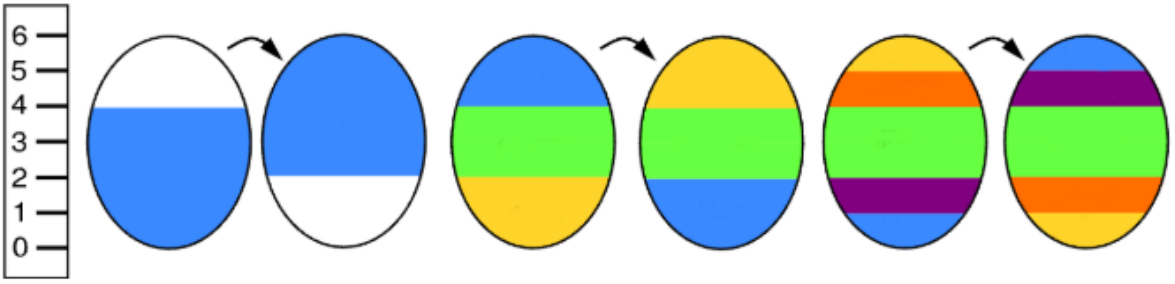


Skaidrojums

Pareizā atbilde ir C.

Šī uzdevuma risināšanā ir svarīgi pamanīt, kuras krāsas pārklāsies. Lielākas otas nokrāsos 4/6 olas, bet mazākas otas - 1/6 olas.

Zemāk ir ilustrācija, kurā redzamas olas krāsas pēc katras krāsošanas un apgriešanas darbības.



C ir pareizā atbilde, jo vidējā 1/3 ir zaļa krāsas josla, kurā sajaucas zilā un dzeltenā krāsa. Abas mazākās otas ir iestatītas dažādos augstumos: tur, kur sajaucas sarkanā un zilā krāsa, parādās violeta josla, bet tur, kur sajaucas dzeltenā un sarkanā krāsa, parādās oranža josla.

A nav pareiza, jo sarkanā krāsa nesajaucas ar zem tās esošo krāsu.

B nav pareiza, jo visa ola būtu jānokrāso zilā krāsā, lai vienā galā būtu zila, bet otrā - zaļa, bet patiesībā tikai 2/3 olas ir nokrāsota zilā krāsā.

D nav pareiza, jo vienīgais veids, kā vienā galā iegūt baltu krāsu, ir neapgriezt olu starp zilo un dzelteno otu, bet mēs zinām, ka olu apgriež pēc katras krāsošanas darbības.

Šī ir informātika

Šajā uzdevumā ola iziet cauri krāsošanas konveieram. To var uzskatīt par iteratīvu algoritmu, kas apstrādā savus “datus”, proti, olas. Katrā iterācijā olu a) krāsošanas laikā pagriež par 360 grādiem un b) apgriež otrādi. Turklāt būtu vajadzīga datu struktūra, lai uzglabātu apstrādājamās olas strīpiņu pašreizējo stāvokli. Šajā gadījumā piemērota datu struktūra varētu būt viendimensiju masīvs, kas sastāv no sešām šūnām: pa vienai katrai joslai. Krāsošanas operācijas laikā šūnu saturs būtu jāpārbauda un jāsajauc ar otiņas krāsu, un iegūtā krāsa aizvietotu veco krāsu šūnā. Iegūto krāsu aprēķina saskaņā ar sniegtajiem rādītājiem. Ja katru olas joslu uzskata par patstāvīgi krāsojamu olas zonu, tad laba datu struktūra būtu sistoliskais masīvs: šādā sistoliskajā masīvā dati iziet cauri viens otram līdzīgu skaitļošanas soļu montāžas līnijai, katram no tiem nedaudz pārveidojot datus, līdz beigās parādās pilnībā apstrādāti dati.