

Bebr[a]s

'24

2024. gada 1. kārtas uzdevumi ar atbildēm

9.-10. klase

INFORMATĪVIE ATBALSTĪTĀJI



Izglītības un zinātnes
ministrija

start(it)

www.startit.lv

Saturs

Saturs.....	1
Atrodi dārgumus.....	2
Kartītes.....	4
Bruņurupuča zīmējumi.....	6
Labirints.....	8
Uzskaites sistēma.....	10
Grand Prix.....	12
Balonu mašīna.....	14
Ēdiena karte.....	16
Uzmini domino kauliņu.....	20
Olu krāsošana.....	22
Lielākais virknes punktu skaits.....	24
Draugi.....	28
Izpēte.....	30
Ķieģeļu siena.....	34
Superbebras.....	36

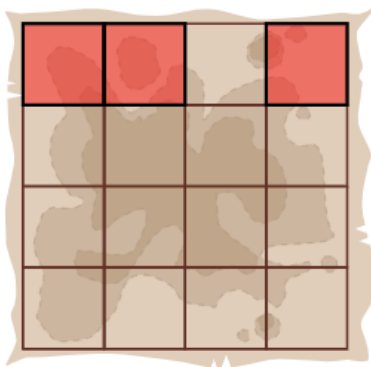
Atrodi dārgumus

Čehija

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.

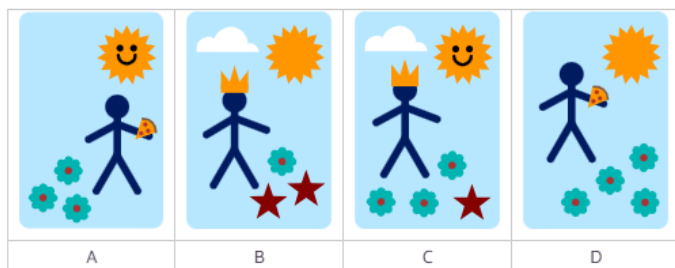
Ja ir 1 000 objektu, ir nepieciešami 10 meklēšanas soļi $2^9 < 1000 < 2^{10}$, bet 1 000 000 objektu gadījumā - 20 soļi. Vispārīgi var teikt, ka n objektu meklēšanai ir nepieciešami aptuveni $\log_2(n)$ soļi; logaritma funkcija ir logaritms pie bāzes 2. Tā kā binārā meklēšana ir tik ātra, to bieži datorprogrammās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

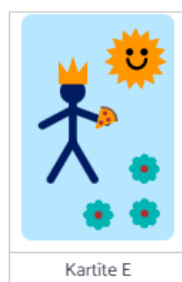
Kartītes

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.

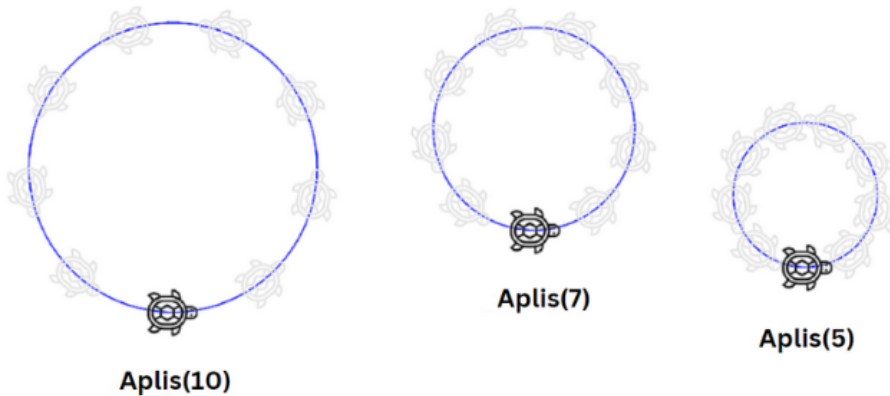
Modeļu atpazīšana: Līdzību vai modeļu identificēšana datos. Piemēram, atpazīt, ka noteikti nosacījumi (piemēram, apmācies laiks) pastāvīgi noved pie konkrētiem rezultātiem (nav ziedu vai saule nespīd).

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

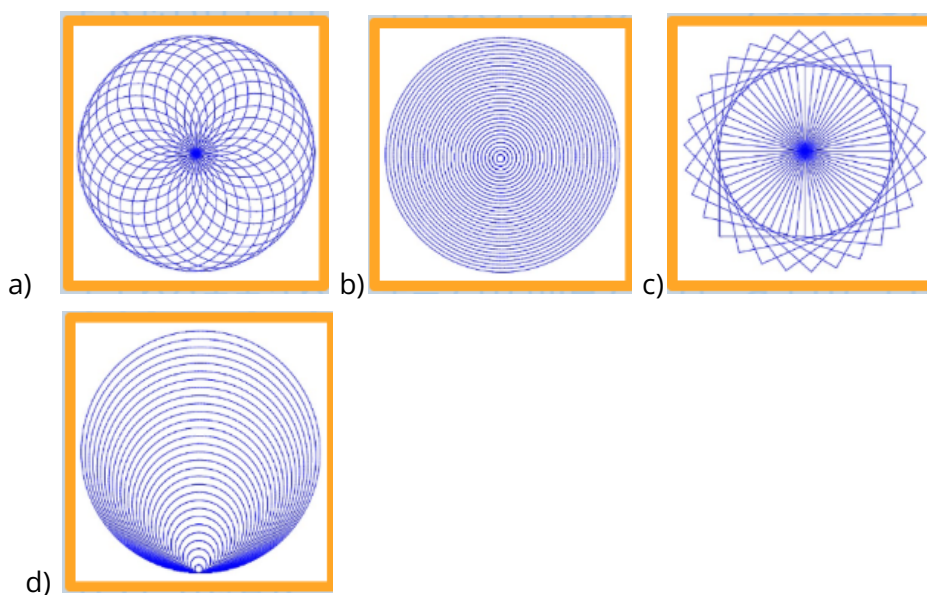
Robots, kurš ir bruņurupuča formā, uz zīmēšanas dēļa izveido zīmējumu atbilstoši instrukcijai. Kad tas saņem komandu zīmēt apli: "aplis(d)", tas uzzīmēs apli, kura izmērs mainīsies atkarībā no parametra d. Mainot parametru d, var tikt uzzīmēti dažāda izmēra apli (skat. attēlu).



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



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 apliem jāatbilst robota sākuma pozīcijai. Pēc tam, kad visi 30 apli ir uzzīmēti, visiem tiem jābūt ar vienu kopīgu sākuma punktu, jo robota sākumpunkts ir nemainīgs. Tātad gala attēlā jābūt 30 dažāda izmēra apliem, un visiem apliem jābūt vienam kopīgam punktam, kā parādīts variantā D.

Kāpēc pārējās atbildes ir nepareizas:

A: visi uzzīmētie apli attēlā ir vienāda izmēra, taču tiem jābūt dažāda izmēra.

B: šiem apliem nav kopīga punkta, no kura robots sāktu tos zīmēt.

C: šis attēls nav veidots no apliem, 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 apli 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.

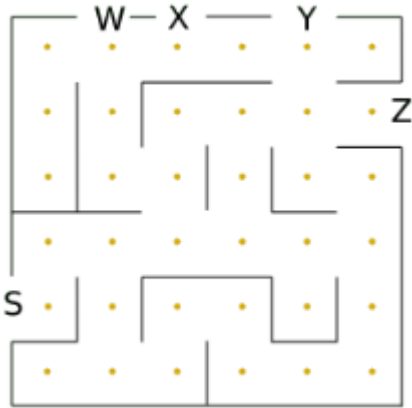
Šī ir algoritmiskā domāšana:

Šis uzdevums prasa pielietot abstrakciju, kas nozīmē atrast būtisko aprakstītajā situācijā. Piemēram, saprast, kā augošā parametra vērtība ietekmē dažāda izmēra aplis. 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 izklūt no labirinta pa kādu no W, X, Y vai Z caurumiem.

Jautājums

Kāds ir minimālais punktu skaits, ko sanāks apmeklēt pirms izešanas 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 nemarkēti 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 pārlicinātos, ka neviens no draugiem nav kļūdījies, katrs uz sava papīra lapiņas uzraksta, kurš kuram ir aizdevis, cik bumbiņu.

Pēc nedēļas viņi salīdzina savas lapiņas:

Toms	Bernards	Anna
Toms → Anna 2	Toms → Anna 2	Toms → Anna 2
Bernards → Toms 1	Bernards → Toms 1	Bernards → Toms 1
Bernards → Anna 3	Toms → Anna 3	Anna → Toms 2
Anna → Toms 2	Anna → Toms 2	Toms → Anna 3

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 blokkēdes 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.

Blokkēdes 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
Likumi	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 likumi.
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 likumu skaitu un kas ir garākas par 3 km.
6. Anna parasti uzvar sezonas sākumā.

Jautājums

Katrā sacīkstē uzvar viens bebrš. 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, Harijs 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 Harijs – 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.

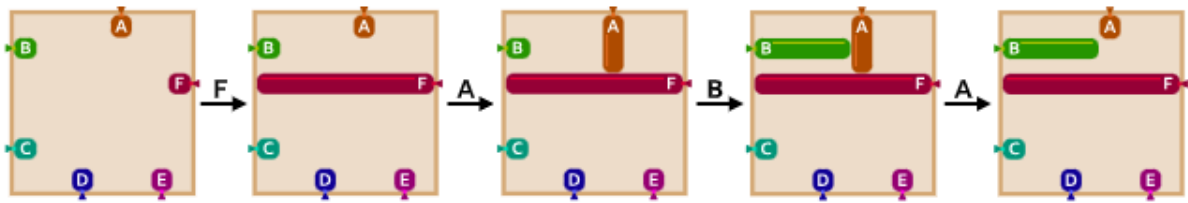
Balonu mašina

Vācija

Bebriem ir mašina, kas var radīt attēlus, piepūšot balonus kvadrātveida formā. Uz baloniem ir uzrakstīti burti A, B, C, D, E un F. Mašina 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 mašina nolasa F, A, B un pēc tam A, tā rīkojas šādi:



Uzdevums

No sākuma katrs balons tiek izpūsts, un tad mašina 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

Iespējamās 4 pareizās atbildes ar deviņiem burtiem:

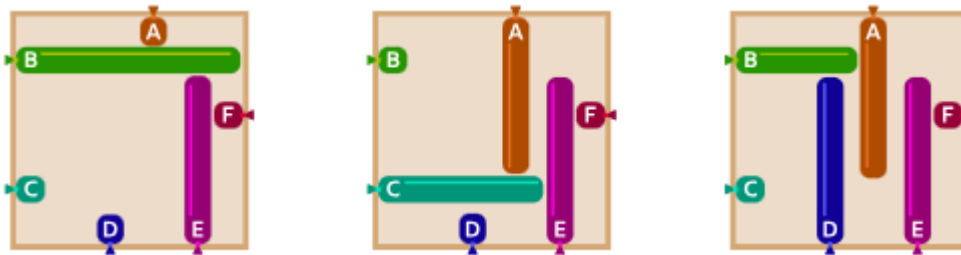
- B E B C A C B D B
- B E C B A C B D B
- B E B C A B C D B
- B E C B A B C D B

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.

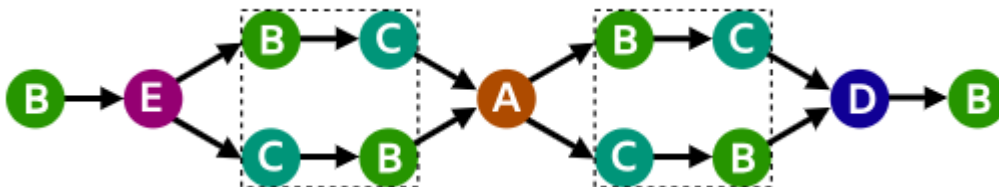
Otrajā attēlā parādīts stāvoklis pēc B E B C A. Balons C kalpo kā šķērslis A, tāpēc tas ir jāpiepūš pirms A. Turklāt balons B ir jāizpūš, pirms var piepūst A, citādi tas nerasniegs vajadzīgo garumu.

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 u.t.t. 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 orientētu aciklisku 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:

🌳	🌳	🌳	🌳	🌳	🌳	🌳
🌳	2	2	1	1	2	🌳
🌳	2				2	🌳
🌳	3	2	1		2	🌳
✖	🌳	🌳	1		1	🌳

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:

🌳	🌳	🌳	🌳
🌳	2	2	1
🌳	2		
🌳	3	2	1
✖	🌳	🌳	1

Labajā dīķa pusē ir iespējami divi varianti:

🌳	🌳	🌳	?
1	1	2	🌳
		2	🌳
		2	?
		1	🌳

🌳	🌳	🌳	🌳
1	1	2	?
		2	🌳
		2	🌳
		1	?

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:

E	F	G	H	I	J	K
D	2	2	1	1	2	L
C	2				2	M
B	3	2	1		2	N
A	 	 	1		1	O

Cipari zilajos kvadrātos satur "ierobežotājus". Skatoties pulksteņrādītāja virzienā no apakšas, pirmie 3 noteikumi (1,1,2) ir izpildīti, 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!

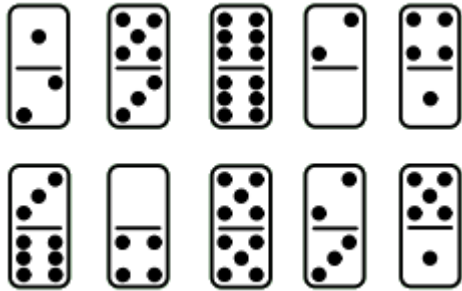
Ievērojiet, ka kopumā, ja ap dīķi ir n koku, tie atbilst 2^n iespējamām konfigurācijām, un tad parasti ir daudz vieglāk pārbaudīt, vai dotais risinājums vispār atbilst uzdevuma ierobežojumiem, nekā to atrast praktiski!

Skatoties uz šo uzdevumu mēs varam arī redzēt, cik bieži uzdevumus var risināt pakāpeniski t.i. atrisināt pa atsevišķām daļām. Nešaubīgi 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 ir 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 iespējam 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:

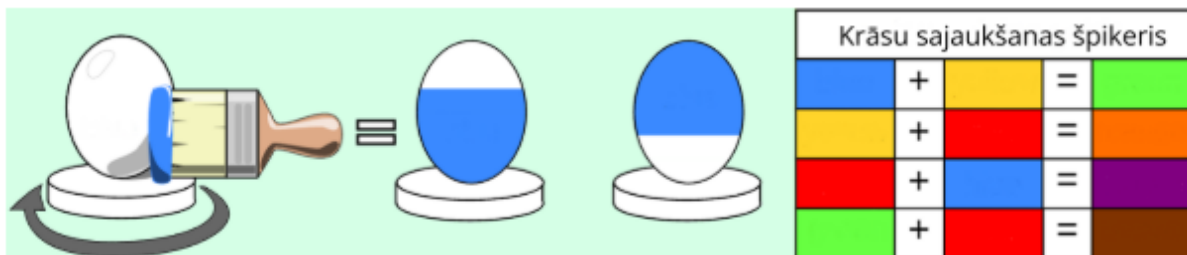
Datori glabā daudz datu, un tiem ir vajadzīgi ātri paņēmieni, kā atrast konkrētu informāciju. Tā vietā, lai aplūkotu katru datu vienību atsevišķi, mēs varam izmantot gudrākas metodes, ja datus labi sakārtojam. Piemēram, ja dati ir sakārtoti, mēs varam izmantot metodi, ko sauc par bināro meklēšanu. Tā 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.

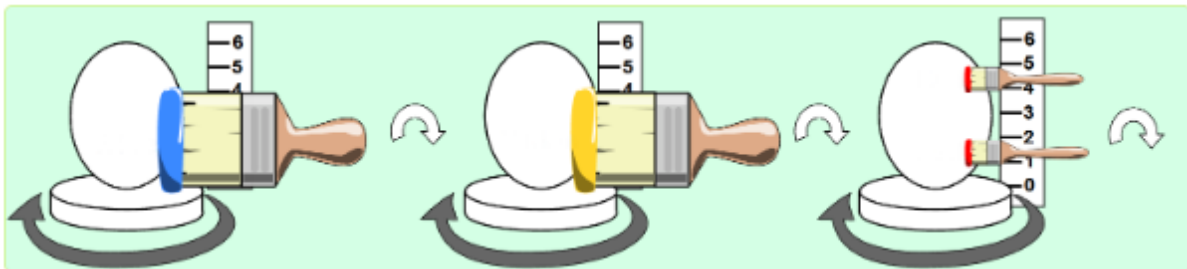
Olu krāsošana

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.

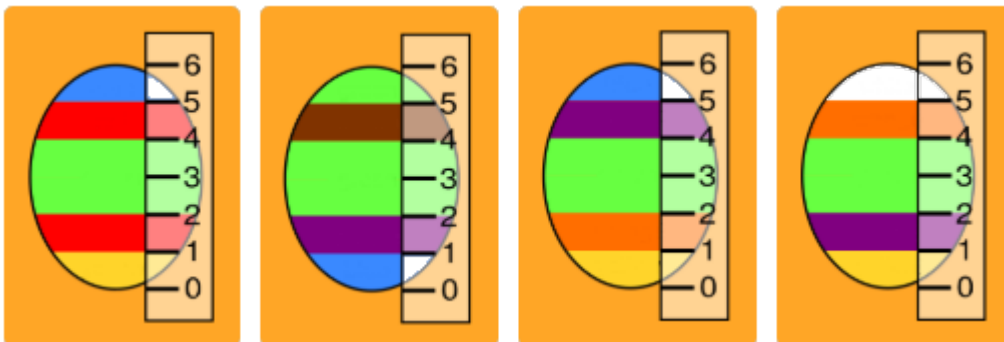


Aija nokrāso savu balto olu, izmantojot krāsu secību un otas platumu, kā parādīts nākamajā attēlā.



Jautājums

Kāda būs Aijas ola pēc krāsošanas?

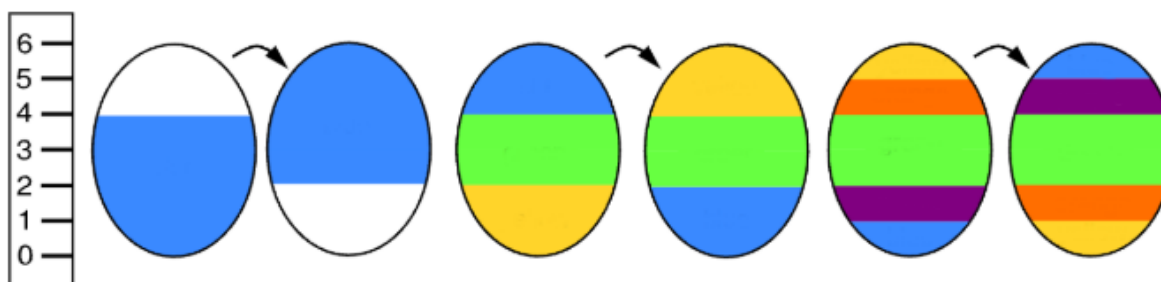


Skaidrojums

Pareizā atbilde ir C.

Šī uzdevuma risināšanā ir svarīgi pamanīt, kuras krāsas pārklāsies. Lielākas olas nokrāsos 4/6 olas, bet mazākas olas - 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 olas 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 dzelteni otru, 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 jā sajauc 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.

Lielākais virknes punktu skaits

Slovākija

Spēlē ir jāredīgē secība, kas izveidotas no kartītēm ar burtiem A, B un C. Spēlētāji cenšas iegūt pēc iespējas augstāku punktu skaitu, nomainot dažus no burtiem.

Par katru 2 vienādu burtu grupu rindā spēlētājs saņem 2 punktus. Par katru 3 vienādu burtu grupu spēlētājs saņem 3 punktus. Un tā tālāk. Piemēram, piecu pelēku kāršu virkne iegūst 0 punktu, jo blakus nav neviena vienāda burta. Taču ir pieejamas divas sarkanas kārtis, lai uzlabotu rezultātu.



Pirmo C var aizklāt ar sarkanu A karti, bet otro C - ar sarkanu B karti, kā parādīts attēlā.



Uzdevums

Tad divu "A" un divu "B" punktu grupai rezultāts ir $2 + 2 = 4$ punkti. Pirmā B ir pati par sevi, tāpēc tā neieskaita punktu skaitu.

Velciet sarkanās kārtis B, B un C, lai nosegtu jebkurus trīs burtus un izveidotu secību ar visaugstāko iespējamo punktu skaitu.



Skaidrojums

Zemāk ir parādīts risinājums:



Šajā rindā jūs iegūstat $2 + 2 + 4 + 3 = 11$ punktus; no kreisās uz labo 2 B, 2 C, 4 B and 3 A vienus pēc otra. Vairak punktus iegūt jūs nevarat.

- Šajā rindā ir 12 burti, tāpēc vairāk par par 12 punktiem jūs iegūt nevarat;
- Tomēr, burts A kreisajā malā nevar piederēt grupai, tapēc ka jums nav neviena A, lai samainītu uz B
- Visi pārējie 11 burti pieder grupai un katrs sastāda pa 1 punktam no kopejā rezultāta.
- Katrs burts var dot tikai maksimums 1 punktu.

Vai ir kādi citi veidi kā iegūt 11 punktus? Lai iegūtu maksimāli lielu punktu skaitu, jums ir jāizkārto burti B, B un C vienā līnijā, kurā izveidojas jauna burtu grupa, bet neviena veco burtu grupa netiek samazināta:

- Ja burts tiek pievienots vienai grupai, jums jāpieskaita 1 punkts.
- Tomēr, ja burts ir novietots blakus vienam burtam un izveido jaunu grupu vai šis burts savienojas ar esošu burtu grupu, jūs varat pieskaitīt 2 punktus.
- Ja burts ir novietots starp diviem citiem brīviem burtiem, tad veidojas 3 burtu grupa un tiek pieskaitīti 3 punkti.

Piemērs jums sākotnēji dod 5 punktus. Nav iespējams iegūt 3 punktus ar B, B vai C. Lai iegūtu 11 punktus, B, B un C jānovieto tādā veidā, lai start tiem katru reizi veidojas jauna grupa un to var izdarīt tikai ar s B un diviem C:



Ir tikai 1 veids kā izveidot 3 jaunas grupas ar B, B un C, nepazaudējot esošo grupu lielumu. Tikai pēdējā gadījumā jūs iegūstat 11 punktus, bet par pārējiem 10 punktu. Šeit ir šie 4 gadījumi:



Šī ir informātika

Šis piemērs skaidri parāda, kā datorspeciālisti rīkojas, meklējot optimālako risinājumu vai metodi risinājuma atrašanai.

Ideja ir atrast tādu B, B un C burtu izvietojumu, kurā iegūst visvairāk punktu. Pirmā pieeja varētu būt vienkārši ģenerēt visas iespējas, kurās trīs burti sērijā tiek aizstāti ar B, B un C. Divpadsmit burtu sērijai ir $12 * 11 * 10 = 1320$ iespējas, un katrai no tām jāaprēķina un jāsalīdzina punkti, lai noteiktu labāko risinājumu.

Risinājumu skaitu var ievērojami samazināt, ja ņem vērā, ka B, B un C saņem punktus tikai tad, ja tie atrodas blakus vienādiem burtiem. Ja ņem vērā arī to, ka esošās grupas nedrīkst samazināt, risinājumu skaits vēl vairāk samazinās. Papildus, ja ņem vērā arī to, ka atsevišķu burtu savienošana dod vairāk punktu nekā esošo grupu palielināšana, tad paliek tikai viens iespējamais izvietojums.

Iepriekš minētās 1320 iespējas nav tik daudz, lai dators tās varētu aprēķināt, Tomēr datorzinātnē vienmēr tiek uzdots jautājums, kā mainīsies problēmas atrisināšanai nepieciešamās pūles, pieaugot problēmas lielumam. Ar garākām rindām un vairāk mainīgajiem burtiem dažādu risinājuma iespējas būtu attiecīgi vairāk. Ir vērts rūpīgi analizēt problēmu un izstrādāt efektīvāko metodi.

No informātikas viedokļa tā ir optimizācijas problēma. Informātikas jomā optimizācijai ir būtiska nozīme. Runa ir par to, lai kaut ko padarītu pēc iespējas efektīvāku vai funkcionālāku ņemot vērā noteiktus ierobežojumus. Optimizācijas mērķis ir atrast labāko no visiem iespējamajiem risinājumiem. Piemēram, kad mēs runājam par koda optimizāciju kompilatorā, mēs runājam par koda pārveidošanas procesu, lai padarītu to efektīvāku un mazāku, nemainot tā rezultātus un neveidojot blakusparādības. Tas ir svarīgi, jo palīdz programmām darboties ātrāk un izmantot mazāk resursu. Piemēram, datu saspiešana ir viens no optimizācijas optimizācijas piemēriem. Tas ietver informācijas kodēšanu, izmantojot mazāk bitu nekā sākotnēji, lai ietaupītu glabāšanas vietu vai samazinātu pārsūtāmo datu apjomu. Šajā uzdevumā, mēģinot optimāli sakārtot burtus "BBC", mēs būtībā darām to pašu - cenšamies atrast visefektīvāko sakārtojumu, kas ļaus sasniegt mūsu mērķi, t. i., izveidot secību, kas dod mums visvairāk punktu. Lai to izdarītu, mēs saglabājam burtus "BBC", lai

palielinātu to pašu burtu secības garumu vai lai izveidotu jaunu apakšvirkni. Visām šādā veidā izveidotajām sekvencēm aprēķinām to punktu skaitu. Pēc tam varam izvēlamies virkni ar vislabāko rezultātu.

Draugi

Lietuva

Klasē ir septiņi bebri: Aleks, Baiba, Kira, Dāvis, Ēriks, Freds un Džeina. Daži no bebbiem viens otru pazīst, bet daži nē.

Tabulā zemāk starp diviem bebbiem ir atzīme tad, ja viņi viens otru pazīst.

	Aleks	Baiba	Kira	Dāvis	Ēriks	Freds	Džeina
Aleks	■	✓	✓				
Baiba	✓	■					✓
Kira	✓		■		✓		
Dāvis				■		✓	
Ēriks			✓		■		
Freds				✓		■	
Džeina		✓					■

Katrs bebris saņēma no citiem atšķirīgu ziņu no kāda bebra ārpus klases un dalījās ar to ar visiem savas klases bebbiem, kurus pazina. Ikreiz, kad bebris saņem jaunu ziņu no kāda ārpus klases, viņš nekavējoties dalās tajā ar visiem klasē pazīstamajiem bebbiem.

Jautājums

Kurš(-i) šīs klases bebris(-iem) kopumā saņēma vismazāk ziņojumu?

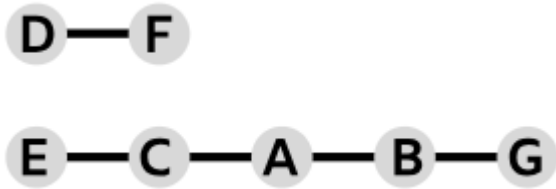


Skaidrojums

Pareizā atbilde ir Dāvis un Freds.

Mēs varam izmantot diagrammu, lai parādītu savstarpējās attiecības starp šīs klases bebbiem. Šajā diagrammā katrs bebris ir attēlots ar apli, un, ja divi bebri viens otru pazīst, tad viņu apļus savieno līnija. Dāvis pazīst tikai Fredu, un Freds pazīst tikai Dāvi, tāpēc mēs

savienojam viņu apļus kopā un nesavienojam tos ar citiem apļiem. Ēriks pazīst tikai Kiru, tāpēc mēs savienojam viņu apļus. Baiba pazīst arī Aleku, tāpēc mēs savienojam viņu apļus. Aleks pazīst arī Baibu, tāpēc mēs savienojam viņu apļus. Visbeidzot, Baiba pazīst arī Džeinu, tāpēc mēs savienojam viņu apļus.



Tagad no diagrammas redzams, ka katrs no Dāvja un Freda var saņemt tikai 2 ziņas, bet katrs no Ērika, Kiras, Aleksa, Baibas un Džeinas saņems 5 ziņas. Tāpēc Dāvis un Freds saņems vismazāk ziņu.

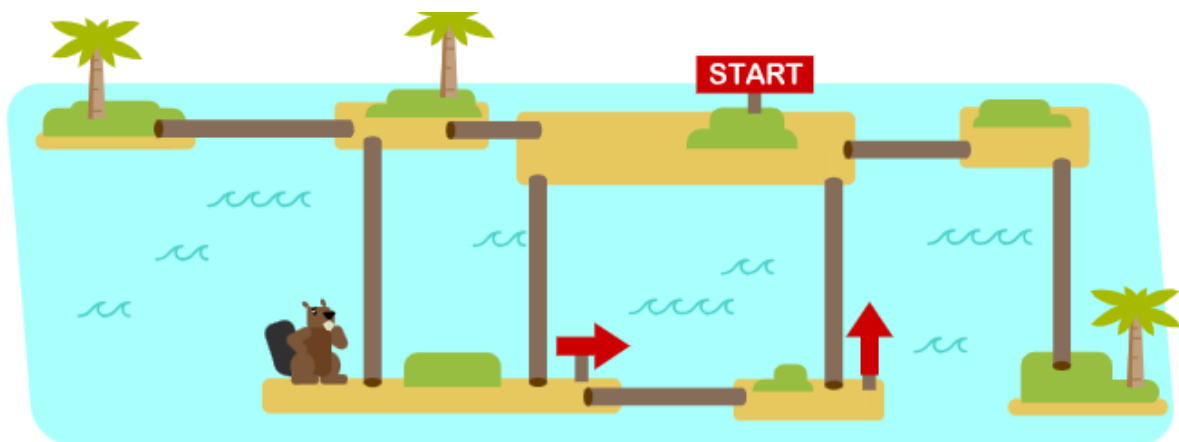
Šī ir informātika

Grafi ir svarīgs abstrakcijas rīks datorikā. Šajā uzdevumā grafs tiek attēlots kā saistību matrica. Bebru draudzības problēma ir saistīta ar grafu sakarības problēmu. Grafu teorijā un datorikā saistību matrica ir kvadrātveida matrica, ko izmanto, lai attēlotu grafu. Matricas elementi norāda, vai virsotņu pāri grafā ir savienoti ar šķauni vai nē. Šis grafs ir neorientēts (t.i., visas šķautnes ir divvirzienu), jo saistību matrica ir simetriska. Saistību matrica var tikt izmantota kā datu struktūra, lai attēlotu grafus datorprogrammās, kas paredzētas darbībām ar tiem. Galvenā alternatīva šai datu struktūrai, ko izmanto arī šajā pielietojumā, ir kaimiņu saraksti.

Izpēte

Čehija

Bebrs Harijs pēta ar balķiem savienotu salu tīklu, kāds parādīts zemāk attēlā. Viņš vēlas apmeklēt visas parādītās salas. No jebkuras salas viņš var redzēt blakus esošo salu un noteikt, vai uz tās ir zīme.



Iepriekš redzamajā attēlā viņš ir sācis no patvaļīgi izvēlētas salas un sasniedz vēl divas salas, izmantojot šādu stratēģiju:

Uzdevums

Aizpildiet tukšās vietas ar pareizajiem norādījumiem, nodrošinot ka Harijs apmeklē visas salas.

levieto pareizos teksta blokus tukšajās vietās, kas sakrīt ar teksta bloku formu. Nospiediet "saglabāt", kad esat pabiedzis

Ejiet uz pirmo salu un novietojiet START zīmi.

Katru reizi, kad apmeklējat jaunu salu:

- Ja uz salas nav zīme, novietojat bultiņu, kura norāda uz salu, no kurienes atnācāt
- Ja salai ir blakus cit sala, uz kuras , tad dodaties uz to salu
- Ja salai ir blakus cit sala, uz kuras
- Ja uz salas ir bultiņa, dodaties uz salu
- Vai paliekat šeit. Jūs esat bijuši uz visām salām vismaz vienu reizi

nav zīme
ir bultiņas zīme
ir START zīme

no kuras jūs atnācāt
uz kuru norāda bultiņa

Skaidrojums

Pareizā atbilde ir:

Ejiet uz pirmo salu un novietojiet START zīmi.

Katru reizi, kad apmeklējat jaunu salu:

- Ja uz salas nav zīme, novietojat bultiņu, kura norāda uz salu, no kurienes atnācāt
- Ja salai ir blakus cit sala, uz kuras **nav zīme**, tad dodaties uz to salu
- Ja salai ir blakus cit sala, uz kuras **nav zīme**, tad dodaties uz to salu
- Ja uz salas ir bultiņa, dodaties uz salu **no kuras jūs atnācāt**
- Vai paliekat šeit. Jūs esat bijuši uz visām salām vismaz vienu reizi

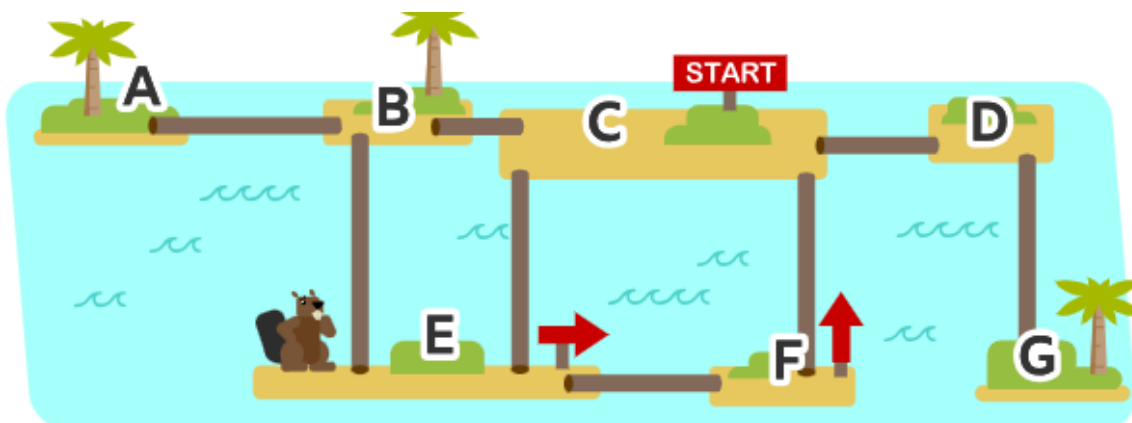
nav zīme
ir bultiņas zīme
ir START zīme

no kuras jūs atnācāt
uz kuru norāda bultiņa

Pirmais tukšais laukums:

Uz salām, kurās Harijs jau ir bijis, ir zīme, bet uz salām, kurās viņš vēl nav bijis, zīmes nav (jo viņš uzstāda zīmi, kad dodas uz salu, uz kuras tās vēl nav).

Pareizi: a) nav zīmes - dodieties uz salām, kurās viņš vēl nav bijis. Piemēram, turpmāk parādītajā situācijā viņam būtu jādodas uz blakus esošo salu bez norādes, B. Ja viņš priekšroku dotu salām, kurām jau ir norādes, viņš pārvietotos starp salām C, E un F, bet nekad nedotos uz salām, uz kurās vēl nav bijis.



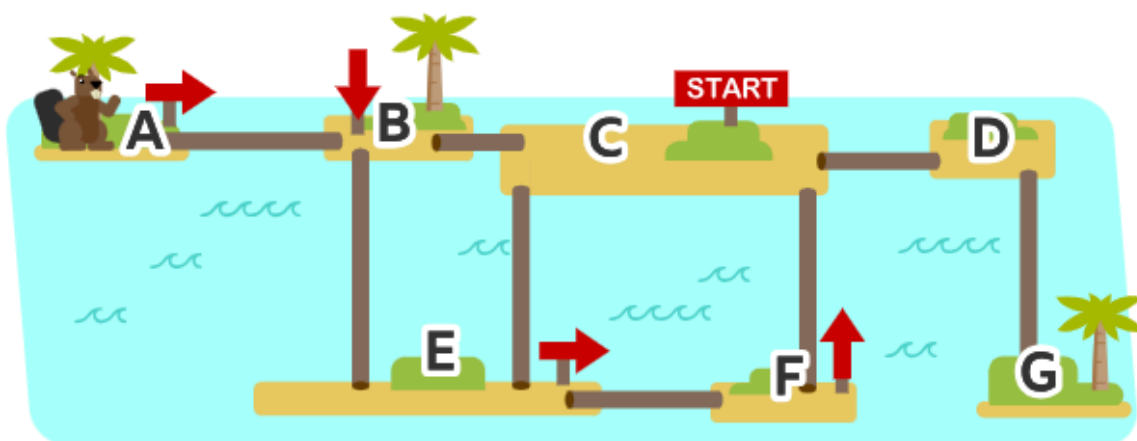
Nepareizi: c) sākuma zīme - algoritma otrās iterācijas laikā Harijs atkal atradīsies sākuma salā, un algoritms nepareizi izpildītu pēdējo "pretējā gadījumā" instrukciju.

Otrs tukšais laukums:

Pareizi: b) sala, uz kuru norāda bultiņa - atpakaļceļš uz salu, kurai vēl ir neizpētītas kaimiņu salas.

Nepareizi: a) sala, no kuras nupat nācis, - var sanākt visu laiku pārvietoties starp divām salām, kurām nav kaimiņu salas bez zīmēm.

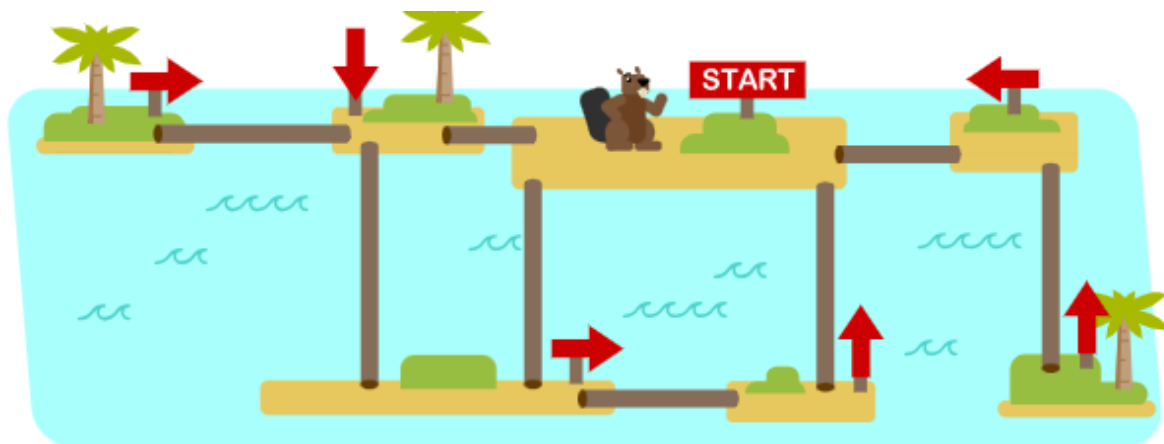
Piemēram, tālāk redzamajā situācijā viņš atrodas uz salas A, kurai vairs nav neizpētītu kaimiņu salu. Viņam nav citas izvēles, kā vien doties uz salu B. Nākamajā solī, ja viņš dotos atpakaļ uz salu, no kuras tikko bija nācis, viņš atgrieztos uz A, tad uz B, tad uz A un nekad neatstātu šo salu pāri. Tādējādi viņam jāseko bultiņām uz zīmēm, lai dotos uz E, F, C un no turienes uz salu D, kurā viņš vēl nav bijis.



Šī ir informātika

Savstarpēji savienotu salu tīklu var modelēt ar grafu palīdzību. Grafs ir datu struktūra. Tas sastāv no virsotnēm(šeit tās atbilst salām) un šķautnēm(šeit tās atbilst koku stumbriem), kas savieno virsotnes. Daudzas sistēmas realitātē var attēlot ar grafu palīdzību. Piemēram, cilvēkus saista draudzība, dzelzceļa stacijas savieno dzelzceļa līnijas, bet iespējamās pozīcijas šaha spēlē - derīgi gājieni. Dažreiz ir svarīgi pilnībā izpētīt grafu, piemēram, ja vēlaties atklāt noteiktas īpašības. Īpaši tas attiecas uz lieliem grafiem ar daudz virsotnēm un šķautnēm.

Šajā uzdevumā izmantoto stratēģiju datorzinātnē sauc par meklēšanu dziļumā jeb dziļkursiju. Kad algoritms ir pilnībā izpildīts, bebrs ir atgriezies uz salas, no kuras viņš sāka, un šajā piemērā salu grupa izskatās šādi.



Algoritms, kas bija jāizpilda, tika pierakstīts pseidokodā. Pseidokods ir dabiskās valodas un programmēšanas valodas sajaukums. Instrukcijas ir formulētas tā, lai tās būtu vispārēji saprotamas. Tiek izmantotas atkāpes (līdzīgi kā programmēšanas valodā Python), lai parādītu, kuras instrukcijas pieder cita citai.

Ķieģeļu siena

Somija

Bobs uzcēla sienu, izmantojot trīs veidu ķieģeļus: A, B un C.

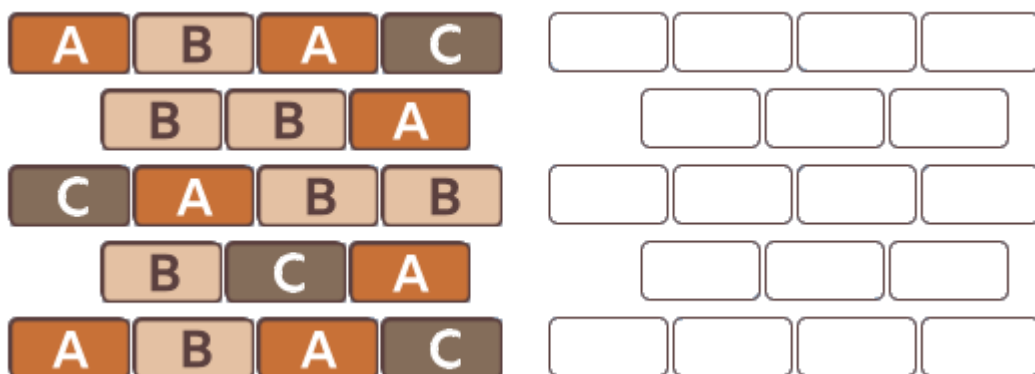
Diemžēl Bobs uzcēla sienu nepareizā vietā, un viņam tā ir jāpārceļ citur. Viņš to dara pa vienam ķieģelim pēc kārtas šādi:

- No sākotnējās sienas noņem jebkuru ķieģeli, tieši virs kura nav neviena ķieģeļa.
- Novieto šo ķieģeli jaunajā sienā vai nu uz zemes, vai uz cita ķieģeļa, bet ne zemāk par jau esošajiem ķieģeļiem.

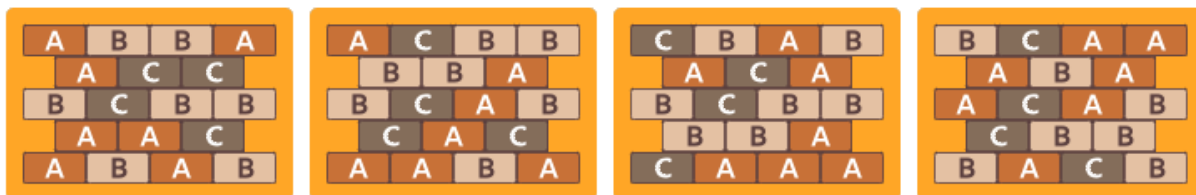
Kamēr viņš ievēro šos noteikumus, Bobam ir vienāga, kādā secībā dažādu veidu ķieģeļi tiek pārnesti no sākotnējās sienas uz jauno. Nākamajā attēlā pa kreisi redzama Boba sākotnējā siena. Jūs varat klikšķināt un vilkt ķieģeļus, lai uzbūvētu jaunu sienu labajā pusē, taču ņemiet vērā, ka netiks pārbaudīts, vai jūs esat pareizi ievērojis noteikumus. Lai sāktu no jauna, noklikšķiniet uz pogas RESET.

Jautājums

Izvēlies visas sienas, ko Bobs var uzbūvēt, sekojot šiem noteikumiem!



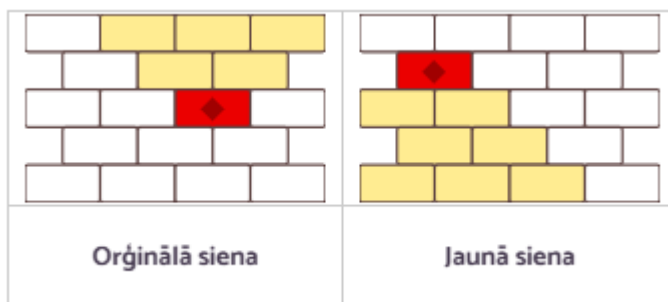
Atbilžu varianti



Skaidrojums

Ir iespējamas tikai sienas A) un C).

Aplūkojiet diagrammu kreisajā pusē, kurā redzams ķieģelis, kas sākotnējā sienā ir iezīmēts sarkanā krāsā. Tas ir otrādi apgriezta trīsstūra gals, kas sniedzas līdz augšējai rindai - dzeltenie izceltie ķieģeļi ir tie, kas noteikti jānoņem, pirms var noņemt sarkano ķieģeli. Līdzīgi, diagrammā pa labi ir attēlots sarkans ķieģelis, kas ir uz augšu vērsta trīsstūra gals (nogriezts šajā gadījumā) - pirms sarkanā ķieģeļa novietošanas ir jānovieto visi dzeltenie ķieģeļi.



Šī ir informātika

Uzdevums ir saistīts ar savstarpējo atkarības (jeb prioritātes ierobežojumu) analīzi starp objektiem. Tā ir klasiska datorzinātnes problēma. Attiecības starp objektiem - šajā Bebras uzdevumā - ķieģeļiem - parasti tiek izteiktas izmantojot attiecību grafiku. Derīga datu apstrādes secība - šajā uzdevumā tā ir secība, kādā ķieģeļus var noņemt un novietot - atbilst objektu topoloģiskai sakārtošanai.

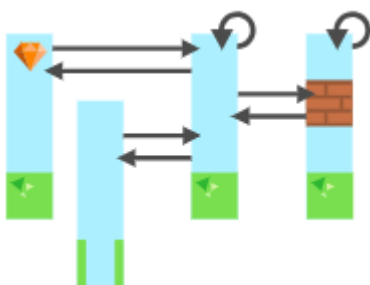
Superbebras




Vācija

Datorspēlē Superbebras spēles fons sastāv no daudzām vertikāliem blokiem. Dators pastāvīgi pievieno jaunu bloku rindas labajā pusē un vienlaikus noņem bloku kreisajā pusē. Tādējādi dators rada ilūziju, ka attēls kustās.



Dators izvēlas jaunu bloku, ko pievienot, izmantojot zemāk redzamo shēmu. Tas aplūko iepriekšējo plāksnīti, pārbauda bultiņas, kas nāk no šīs plāksnītes, un pēc nejaušības principa izvēlas vienu no blokiem, uz kuru norāda bultiņa.



Piemēram, pēc bloka  dators var izvēlēties nākamo bloku vai nu  vai šo  bloku.

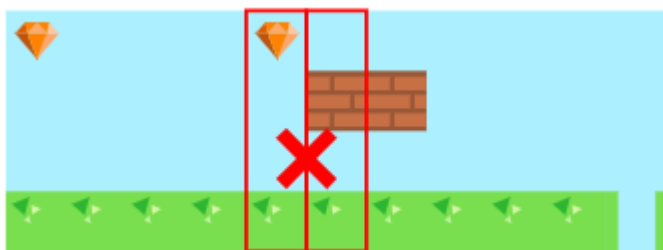
Jautājums

Viens no šiem variantiem nevar būt Superbebras fons. Kurš ir tas?



Skaidrojums

Atbildē C parādītais fons nav pilnībā nesaskan ar pirmstam parādīto shēmu.



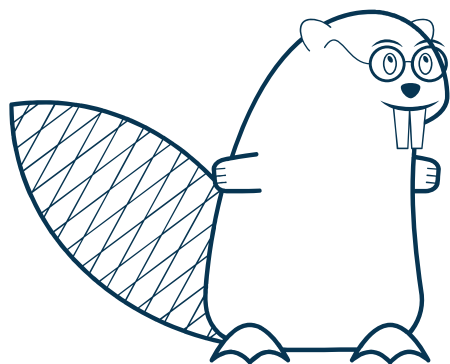
Blokam ar dimantu var sekot tikai, tikai viens specifisks bloks, kas šajā gadījumā nav tā noticis

Ir vairāki veidi, kā atrast šī uzdevuma risinājumu. Vienkāršākā stratēģija ir izmēģināt katru doto fona attēlu un pārbaudīt katru bloku pēc shēmas, lai pārlicinātos, vai tas ir pareizs. Ātrākais veids ir šāds: Jūs pārbaudāt shēmu pēc blokiem, sākot ar tiem, no kuriem iziet tikai viena bulta. Šīs bultas uzskatiet par ierobežojumiem. $A \rightarrow B$ nozīmē, ka "blokam A jāseko blokam B". Tad pārbaudiet, vai šis ierobežojums ir spēkā katram atbilstošam variantam.

Šī ir informātika

Dažās datorspēlēs, piemēram, bezgalīgās skriešanas spēlēs, fons pārvietojas horizontāli, radot ilūziju, ka spēlētājs pārvietojas uz priekšu pa iedomātu pasauli. Dažkārt fons nav nemainīgs attēls, bet to automātiski izveido dators. To sauc par procedurālo ģenerēšanu. Parasti mazāki elementi netiek nejauši apvienoti, lai radītu visdažādākos fonus.

Uzdevumā šie noteikumi tiek attēloti ar flīžu diagrammu, un šādas diagrammas, kurās elementi ir savienoti ar bultiņām, sauc par virzītiem grafiem, tos sauc arī par grafiem. Elementi tiek saukti par mezgliem, bet bultas ir norādītās malas. Virzītos grafus izmanto visiem modelēšanas veidiem.



copyright Bebras