



Concurs Bebras Romania

DRAFT

Acest document este adresat doar
profesorilor! Va rugam sa nu il distribuiti
elevilor si nu il postati pe Internet!

2016

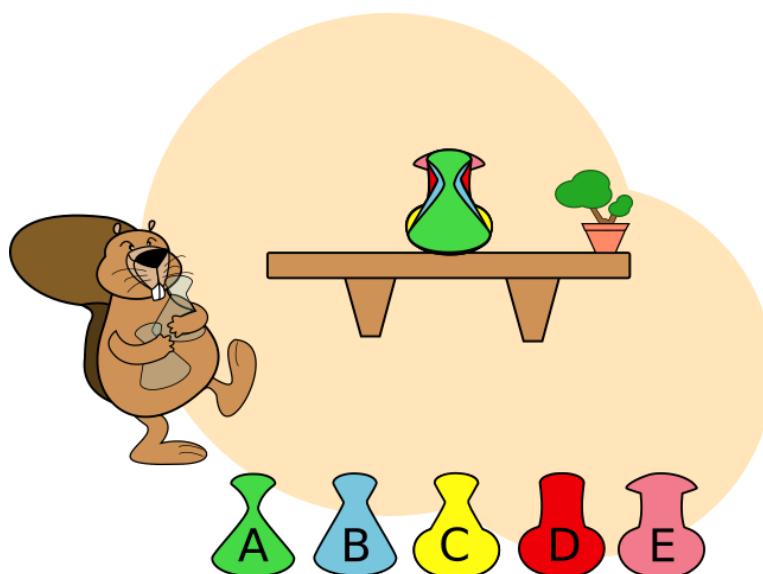
Continut

Sticle	pagina 3		Cel mai scurt drum pe bicicleta	pagina 33
Discutii intre prieteni	pagina 4		Pesterile	pagina 35
Tuburile	pagina 6		Triunghiuri	pagina 37
Codul castorilor	pagina 7		Carti de joc si conuri	pagina 39
Jocul de fotbal	pagina 9		Bilele rosii si albastre	pagina 41
Labirint	pagina 10		Pictura	pagina 43
Robotul	pagina 11		Aplicatia de posta electronica	pagina 45
Drumul cu masina	pagina 12		Scanner	pagina 47
Banner pentru petrecere	pagina 13		Codul Kix	pagina 49
Reteta secreta	pagina 14		Enigma	pagina 51
Picteaza in negru	pagina 15		Jocul	pagina 53
Culoarea florilor	pagina 16		Veveritele egoiste	pagina 56
Potiuni magice	pagina 18		Maimuta	pagina 58
Spitalele	pagina 20		Jocul in L	pagina 60
Hurling	pagina 22			
Directii concurente	pagina 24			
Statia de autobuz	pagina 26			
Gaseste hotul	pagina 28			
Rafting	pagina 30			
Segway	pagina 31			

Un castor pune 5 sticle pe o masa.

El le plaseaza astfel incat sa fie vizibil macar un pic din fiecare sticla.

El plaseaza prima sticla in partea din spate a mesei si apoi pune fiecare noua sticla in fata celei anterior asezate.



Intrebare:

In ce ordine sunt plasate sticlele astfel incat ele sa se vada ca in imagine?

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

Raspuns corect:

E D C B A

Explicatie:

Aceasta intrebare poate fi rezolvata in mai multe moduri. Daca va dati seama ca vaza subtire trebuie sa fie in fata (pentru ca altfel nu ar fi vizibila din cauza celorlaltor sticle), atunci deja ati determinat ca sticla A este cea din prim plan. Puteti incerca apoi acelasi rationament cu fiecare sticla in parte pana rezolvati intrebarea. De asemenea, puteti verifica care dintre sticle este mai larga in partea de sus sau in mijloc intrucat aceea este zona prin care se diferentiaza sticlele. Sticlele mai mici (inguste) trebuie sa fie plasate mai in fata pentru a putea fi vizibile.

Relationarea cu gandirea computationala:

Competente - Abstractizare (AB), Evaluare (EV)

Domeniu - Date, structuri si reprezentari de date

Aceasta este in esenta o problema legata de sortare. Intrebarea cere sa sortati sticlele intr-un anumit mod. De aceea, formele si dimensiunile sunt importante. Ca urmare, stabilirea ordinii trebuie realizata in conformitate cu aceste proprietati.



Discutii intre prieteni

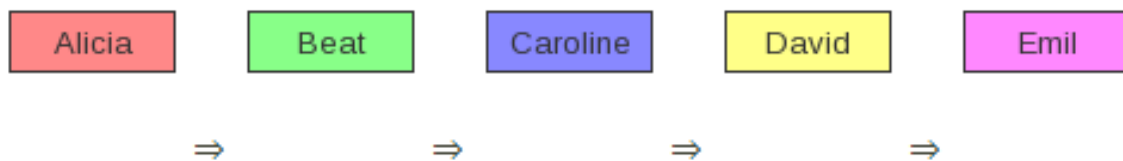
Pentru a organiza o petrecere surpriza, castorul Sara trebuie sa discute cu 5 prieteni: Alicia, Beat, Caroline, David si Emil.

Sara poate vorbi cu Emil imediat. Totusi, pentru a discuta cu ceilalti prieteni, ea trebuie sa ia in considerare anumite aspecte:

- 1- Inainte de a vorbi cu David, trebuie sa discute mai intai cu Alicia.
- 2- Inainte de a vorbi cu Beat, trebuie sa discute mai intai cu Emil.
- 3- Inainte de a vorbi cu Caroline, trebuie sa discute mai intai cu Beat si David.
- 4- Inainte de a vorbi cu Alicia, trebuie sa discute mai intai cu Beat si Emil.

Intrebare:

In ce ordine ar trebui Sara sa vorbeasca cu prietenii ei? Plasati casetele in ordinea corecta, intre sageti.



Discutii intre prieteni

Raspuns corect:

Emil $\Rightarrow$ Beat $\Rightarrow$ Alicia $\Rightarrow$ David $\Rightarrow$ Caroline

Explicatie:

Aceasta intrebare are legatura cu dependentele.

- Emil este singurul prieten care nu depinde nimeni altcineva, deci el trebuie sa fie primul.
 - Beat depinde doar de Emil, asa ca el este al doilea.
 - Alicia depinde de Beat si Emil, asa ca ea va fi a treia.
 - David depinde de Alicia, asa ca el va fi urmatorul,
 - In final, Caroline depinde de Alicia si David, asa ca ea va fi ultima.
- Nici o alta ordine nu indeplineste dependentele definite.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Descompunere in factori primi (DE)

Domeniu – Algoritmi si programare

Notiuni – Dependente functionale, graf

Respectarea dependentelor este o problema computationala intalnita frecvent in viata reala.

Aceasta problema de dependenta si ordonare poate fi modelata cu ajutorului unui graf. Un graf este compus din varfuri (reprezentand in acest caz prietenii) si arce care leaga unele varfuri de altele (in cazul nostru va exista un arc de la X la Y in cazul in care X trebuie sa apara dupa Y). In cazul de fata, graful este unul special: avand in vedere ca exista o anumita ordine, el nu va contine cicluri. Un ciclu reprezinta o traiectorie care incepe intr-un varf, parcurge apoi o serie de arce si in final se reintoarce in varful din care a pornit. Graful din aceasta intrebare este un graf aciclic direct (DAG – Directed Acyclic Graph). Aceasta problema poarta denumirea de sortare topologica: sortarea varfurilor se realizeaza in raport cu relatia de dependenta.

O metoda de a realiza acest lucru o reprezinta selectarea tuturor varfurilor din care nu iese niciun arc (ca urmare, acele sarcini pot fi indeplinite primele) si plasarea lor la inceputul raspunsului dvs. Apoi, aceste varfuri pot fi eliminate din graf si se poate continua cu un nou graf.

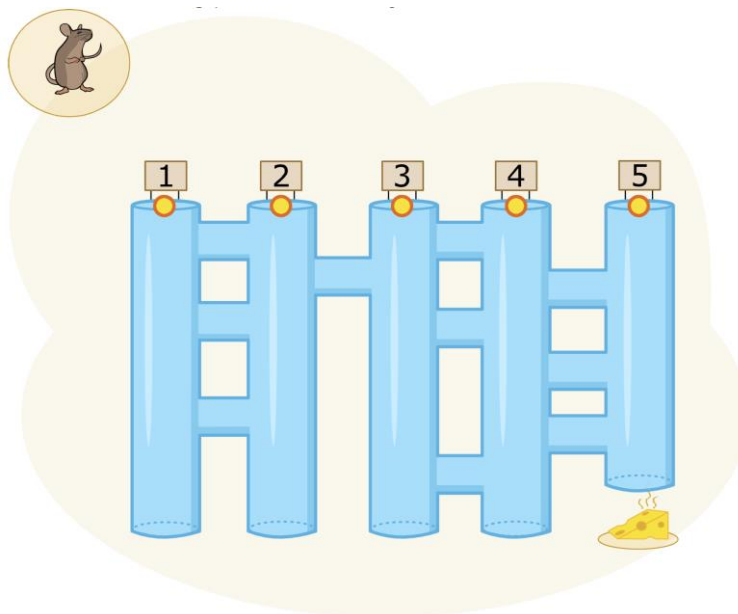
Existenta unui varf din care nu iese niciun arc este garantata de faptul ca acest graf este aciclic.

Tuburile

Un soricel se afla la intrarea intr-un sistem de tuburi si doreste sa ajunga la cascavalul aflat la finalul tubului 5.

Soricelul foloseste mereu urmatoarele 2 comenzi:

1. merge in jos pana cand ajunge la o rascruce;
2. cand ajunge la o rascruce, trece prin tubul vertical, apoi merge la comanda 1.



Intrebare:

Din ce tub trebuie soricelul sa porneasca pentru a ajunge la cascavalul de la finalul tubului 5?

1 2 3 4 sau 5

Raspuns corect:

3

Explicatie:

Pornind din tubul 1 soricelul va ajunge mereu in tubul 3.

Pornind din tubul 2 el ajunge in tubul 1.

Pornind din tubul 4 el ajunge in tubul 2.

Pornind din tubul 5 el ajunge in tubul 4.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL)

Domeniu – Algoritmi si programare

Majoritatea robotilor sunt programati sa execute anumite comenzi exacte. Acest soricel face acelasi lucru: executa la nesfarsit comenzile ‘mergi in jos’ si ‘schimba directia la urmatoarea rascruce’. Aceste comenzi depind de alegerea tubului de intrare, determinand traseul soricelului in sistemul de tuburi. Majoritatea computerelor respecta regula: daca utilizatorul introduce mereu aceleasi date de intrare, computerul va realiza aceleasi calcule si va returna mereu acelasi rezultat.

Codul castorilor

Barbara a primit 2 timbre. Cu unul poate produce o mica floare, iar cu celalalt un mic soare.

Fiind o fata desteapta, ea gaseste o modalitate de a-si scrie propriul nume utilizand codul de mai jos:



Litera	B	A	R	E	Y
Cod		 	  	   	   

Astfel, numele “Barbara” devine:

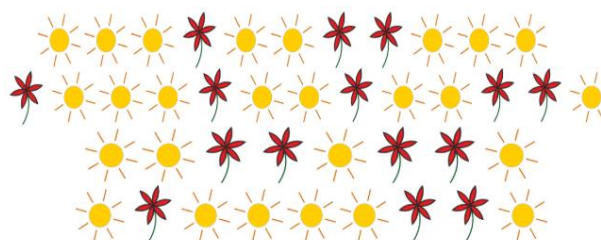


Apoi, ea scrie si numele prietenilor sai. Din pacate, acestea se amesteca intre ele.

Intrebare:

Trageti de codurile din dreapta si plasati-le in dreptul numelor celor 4 prieteni in tabelul din partea stanga

Abby	
Arya	
Barry	
Ray	



Raspuns corect:

Abby	       
Arya	          
Barry	            
Ray	        

Explicatie:

Aceasta problema se poate rezolva foarte usor prin eliminare. Numele Abby incepe cu literele A si B si ca urmare veti cauta un cod care incepe cu 2 simboluri de soare si o floare. Este un singur cod care respecta aceasta ordine, deci asignarea este rapida. Apoi, numele Arya incepe cu 3 simboluri de soare si o floare, identificarea facandu-se extrem de usor intrucat un singur cod respecta modelul. Continuand in acelasi mod, toate codurile pot fi rapid asignate numelor corecte.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Descompunere in factori primi (DE), Generalizare (GE)

Domeniu - Date, structuri de date si reprezentari

De multe ori in Informatica si Stiinta Computationala, in locul stocarii datelor intr-o maniera simpla si directa, se poate concepe o schema de stocare a datelor intr-un mod mai eficient, utilizand mai putin spatiu de stocare.

De exemplu, computerele stocheaza informatiile cu privire la caracterele tastate la tastatura folosind codul ASCII. Fiecarei litere ii corespunde o secventa diferita de 8 biti (reprezentati prin valorile 0 si 1). In codul ASCII, fiecare caracter ocupa acelasi spatiu.

Cu toate acestea, literele au diferite frecvente de aparitie in cadrul unor texte (de exemplu, litera “E” este cea mai intalnita litera in cadrul cuvintelor din limba Engleza), aceste frecvente putand fi luate in considerare pentru a imbunatati modul de codificare.

In mod specific, literele frecvente sunt codificate folosind coduri mai scurte: in aceasta intrebare, B ar trebui sa fie o litera frecventa si este reprezentat printr-un singur simbol; litera A este reprezentata prin 2 simboluri, iar celelalte litere sunt reprezentate prin mai mult de 2 simboluri. Exista un algoritm faimos si utilizat pe scara larga pentru codificarea textelor, denumit codificarea Huffman. Nu se poate folosi totusi orice tip de codificare se doreste, ci numai una care sa fie perfect lipsita de ambiguitate. De exemplu, sa presupunem ca acest cod ar fi fost urmatorul: litera B era reprezentata printr-o floare, litera A prin 2 flori. Ce ar fi insemnat atunci un cod reprezentat prin 2 flori? Ar fi putut si BB sau A, dar nu exista o metoda sigura de a determina codul corect. O metoda de a asigura lipsa de ambiguitate o reprezinta asigurarea faptului ca un cod nu are prefix; cu alte cuvinte, daca se analizeaza codul unei litere, acesta nu reprezinta inceputul unui alt cod.

De asemenea, intrucat codul Huffman folosit depinde de textul insusi (adica de frecventa de aparitie a literelor), se necesita stocarea corespondentelor dintre cod si literele efective. Acest lucru implica ocuparea unui spatiu mai mare de stocare, dar acesta este neglijabil pentru textele lungi,

Jocul de fotbal

Castorii de la echipa Beaver Rangers joaca un joc de fotbal impotriva echipei Forest Raiders.
Urmatoarea lista arata numele jucatorilor care au inscris un gol:

minutul 1: Anna

minutul 10: Dick

minutul 35: Bernard

minutul 47: Smithy

minutul 73: Backy

minutul 89: Richard



Intrebare:

Daca se stie ca doar o echipa a reusit sa inscrie doua goluri la rand, atunci care dintre urmatoarele nu poate fi scorul final al jocului?

3-3 5-1 2-4 sau 4-2

Raspuns corect:

5-1

Explicatie:

Exista 3 posibile scoruri finale ale acestui joc. Daca o echipa inscrie prima si apoi cealalta echipa inscrie 2 goluri la rand, scorul final va fi 3-3. Daca prima echipa inscrie prima si apoi inscrie inca 2 goluri la rand, scorul final va fi 4-2. Daca cea de a doua echipa inscrie prima si apoi inscrie inca 2 goluri la rand, scorul final va fi 2-4.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Evaluare (EV)

Domeniu – Algoritmi si programare

Notiuni – Conditii IF

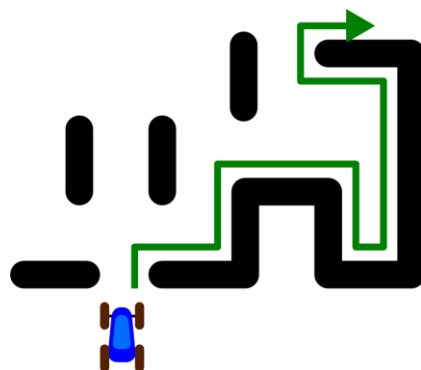
Solutia acestei probleme este o functie IF imbricata. Prima functie IF verifica ce echipa inscrie primul gol. A doua functie IF verifica daca o echipa a inscris doua goluri la rand.

Labirint

O masinuta robotizata utilizeaza o regula simpla de a merge prin labirint:

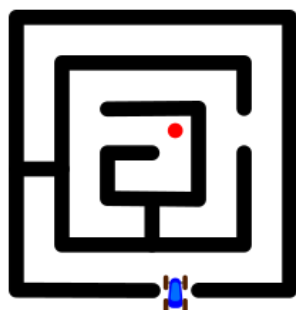
va coti la dreapta de fiecare data cand va putea.

Imaginea din dreapta va ofera un exemplu despre cum merge masinuta prin labirint.

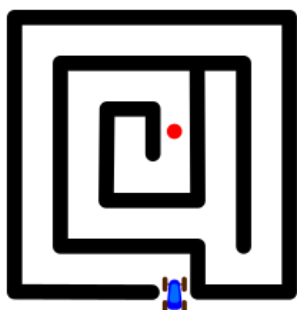


Intrebare:

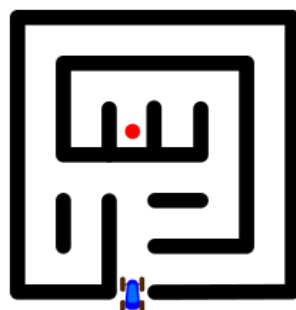
Utilizand aceasta metoda, in cate din urmatoarele labirinturi va reusi masinuta sa ajunga la punctul rosu?



Labirint A



Labirint B



Labirint C



Labirint D

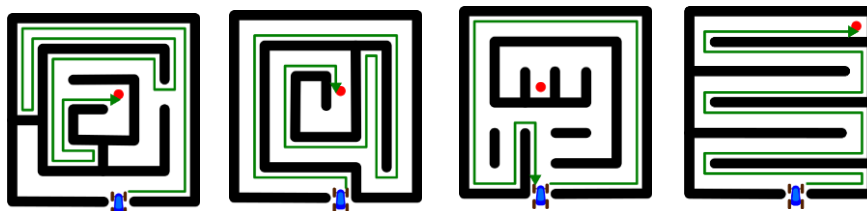
Alegeti dintre variantele: 0 1 2 3 sau 4

Raspuns corect:

3

Explicatie:

In imaginile de mai jos, sageata verde indica drumul parcurs de masinuta. In labirintul C, toata partea centrala a labirintului nu este parcursa si ca urmare punctul rosu nu este atins. In toate celelalte cazuri, punctul rosu este atins.



Relationarea cu gandirea computationala:

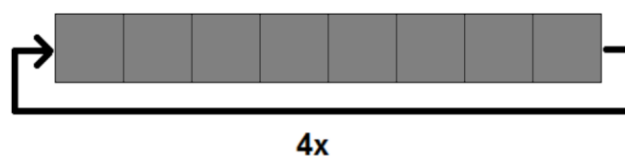
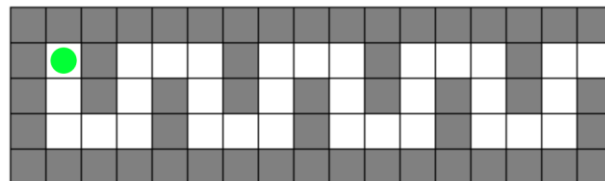
Competente – Gandire algoritmica (AL)

Domeniu – Algoritmi si programare

Metoda utilizata poarta numele de “wall follower”. Este unul din cei mai simpli algoritmi de rezolvare a unui labirint a carui schema nu se cunoaste dinainte. Prin respectarea acestei reguli, nu te vei pierde niciodata: vei reveni mereu, in cele din urma, in punctul din care ai plecat, Totusi, asa cum este exemplificat si mai sus, acest algoritm nu garanteaza parcurgerea intregului labirint.

Robotul

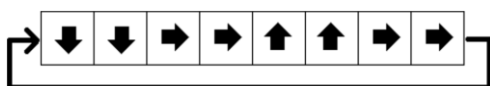
Ajuta robotul verde sa iasa din labirint.
Robotul va repeta aceste instructiuni de 4 ori.



Intrebare:

Trage de sageti pentru a forma un set de instructiuni.

Raspuns corect:



Explicatie:

In robotica, rezolvarea problemelor de tip labirint este una dintre cele mai uzuale. Pentru rezolvarea acestei probleme, este utilizat un robot autonom. Labirinturile pot fi de diferite tipuri: cu bucle sau fara bucle, cu sau fara caroi (grid). In acest algoritm de tip labirint cu bucla, robotul trebuie sa urmeze o serie de instructiuni legate de directie.

Relationarea cu gandirea computationala:

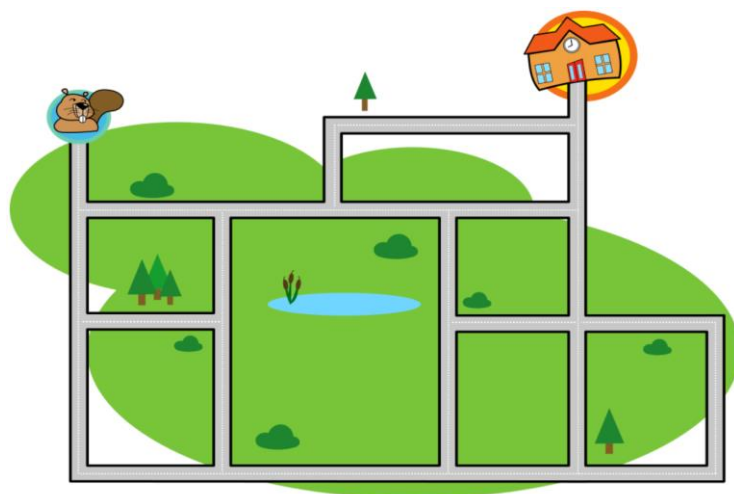
Competente – Gandire algoritmica (AL)

Domeniu – Algoritmi si programare

In programare, o bucla reprezinta o secventa de instructiuni care se repeta continuu pana cand este indeplinita o anumita conditie. Se evalueaza expresia din instructiune, iar in urma acestei evaluari se poate lua decizia de a incepe o noua iteratie sau de a trece la instructiunea imediat urmatoare buclei.

O bucla reprezinta o idee fundamentala in programare, utilizata in mod curent in realizarea aplicatiilor software.

Drumul cu masina



O masina care se conduce singura trebuie sa-l duca pe castor la scoala.

Masina este programata astfel incat sa utilizeze doar 3 instructiuni:

Inainte: merge inainte pana cand intalneste un obstacol

Stanga: vireaza la stanga cu 90°

Dreapta: vireaza la dreapta cu 90°

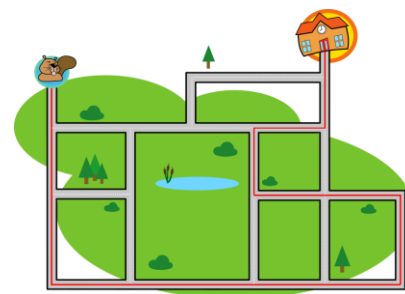


Intrebare:

Scriti un set de instructiuni care sa il duca pe castor la scoala.

Puteti face acest lucru tragand de cele 3 butoane portocalii in zona albastra din dreptul masinii.

Raspuns corect:



Explicatie:

Cel mai important lucru care trebuie retinut este ca, dupa virarea la dreapta cu 90 de grade, masina nu merge inainte. Ca urmare, instructiunea "Inainte" trebuie adaugata dupa fiecare viraj.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Descompunere in factori primi (DE)

Domeniu – Algoritmi si programare

Aceasta intrebare este similara cu realizarea unor instructiuni pentru un robot: scrierea unui program de computer necesita stabilirea unor instructiuni ce trebuie executate pas cu pas. Programele sunt esentiale in utilizarea unui computer: ele ii precizeaza computerului secventa operatiilor pe care acesta trebuie sa le execute. Computerele si robotii sunt foarte rapizi in executarea calculelor si efectuarea unor actiuni repetitive, insa nu au capacitatea de a gandi si ca urmare necesita instructiuni precise de realizare a anumitor sarcini. Asa cum este prezentat mai sus, ordinea operatiilor este foarte importanta: setul corect de instructiuni puse intr-o ordine gresita nu va genera rezultatul scontat.

Banner pentru petrecere

Pentru petrecerea pe care o organizezi, ai o rola lunga de hartie colorata.

Hartia are 3 culori (galben, rosu, albastru), intr-un model care se repeta in mod regulat.

Sora ta a taiat o portiune din hartie, asa cum este aratat in desenul de mai jos.



Sora ta iti va inapoia bucata de hartie lipsa (reprezentata in desenul de mai sus cu ...) daca poti ghici in mod corect dimensiunea hartiei pe care a taiat-o.

Intrebare:

Sora ta spune ca bucata de hartie pe care a taiat-o are una din urmatoarele dimensiuni. Care dintre ele este cea corecta?

31 32 33 sau 34

Raspuns:

31

Explicatie:

Cunoastem faptul ca modelul se termina in YRR, ceea ce inseamna ca s-a taiat cel putin un B. Dupa aceea, au fost taiate mai multe secvente de 4 litere (YRRB). Apoi, partea din dreapta a hartiei trebuie sa aiba YR, intrucat a doua parte incepe cu RB. Astfel, lungimea hartiei taiate este 1 (pentru B) + $4 \cdot X$ (unde X reprezinta numarul modelelor YRRB repetitive) + 2 (pentru YR). Ca urmare, lungimea hartiei taiate este $4X + 3$.

Verificand cele 4 raspunsuri corecte posibile, se constata ca $31/4$ da rest 3: adica $31 = 4 \cdot 7 + 3$. Asadar, ecuatia este rezolvata in conditiile in care $X = 7$. Niciunul din celelalte 3 raspunsuri nu poate fi descompus in $4X + 3$.

Relationarea cu gandirea computationala:

Competente – Abstractizare (AB), Evaluare (EV), Generalizare (GE)

Domeniu – Algoritmi si programare

Gasirea unui sablon in cadrul unor informatii este esentiala in rezolvarea multor probleme. Spre exemplu, secventa AND-ului este compusa din sabloane, iar identificarea unor sectiuni repetitive sau subsiruri care satisfac o anumita proprietate reprezinta o zona importanta a cercetarilor desfasurate in domeniul geneticii si medicinei. Pentru rezolvarea unor astfel de probleme, se folosesc algoritmi de procesare a textului si programe de potrivire a sabloanelor pentru a determina daca anumite siruri apar intr-o secventa de text.

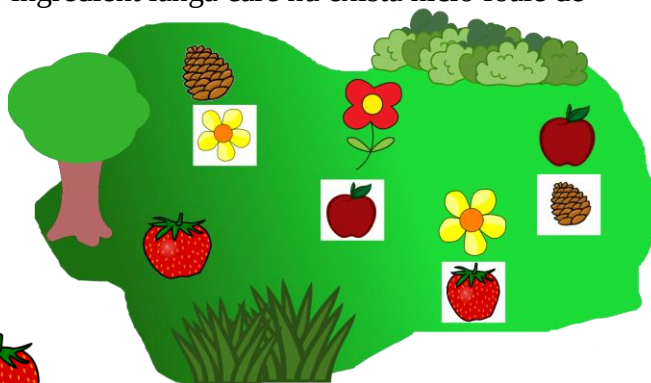
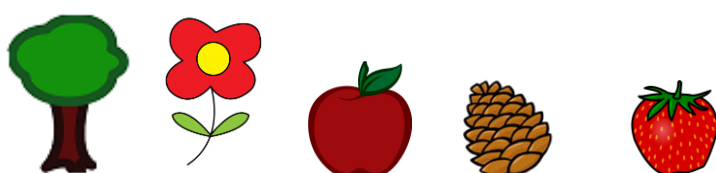
Aceasta problema implica de asemenea notiuni legate de abstractizare si generalizare. Se ia o secventa de informatii si se generalizeaza prin includerea intr-o formula sau ecuatie care poate fi rezolvata. Pentru rezolvarea problemelor, oamenii de stiinta din domeniul informaticii iau o explicatie si o convertesc in ceva concret, standardizat si matematic astfel incat sa poata crea un program care sa rezolve acea problema.

Reteta secreta

Castorii se pregatesc pentru Festivalul Culinar si ar dori sa gateasca o Prajitura Crunchy Cake. Insa, bucatarul lor este in concediu. Kate a promis ca va gati ea o prajitura, insa ea stie ca este extrem de important sa adauge cele 5 ingrediente esentiale in ordinea corecta. Atunci cand ajunge in gradina, ea isi da seama ca langa fiecare ingredient este o foaie de hartie ce arata imaginea ingredientului ce trebuie adaugat urmatorul in reteta. Exista un singur ingredient langa care nu exista nicio foaie de hartie. Gradina arata astfel:

Intrebare:

Ce ingredient trebuie adaugat primul in reteta?



Raspuns: **Explicatie:**



Daca Kate incepe cu floarea rosie, ea poate adauga toate cele 5 ingrediente in ordinea corecta. Primul ingredient adaugat trebuie sa fie cel care nu are nicio foaie de hartie langa el.

Daca alege capsuna, nu putea continua la urmatorul ingredient intrucat nu exista nicio foaie de hartie care sa indice spre acesta. Daca incepea cu marul, ar fi sarit peste floarea rosie. Daca ar fi inceput cu conul de pin, ar fi sarit peste floarea rosie si peste mar.

Relationarea cu gandirea computationala:

Competente - *Gandire algoritmica (AL), Descomponere in factori primi (DE)*

Domeniu - Date, structuri de date si reprezentari

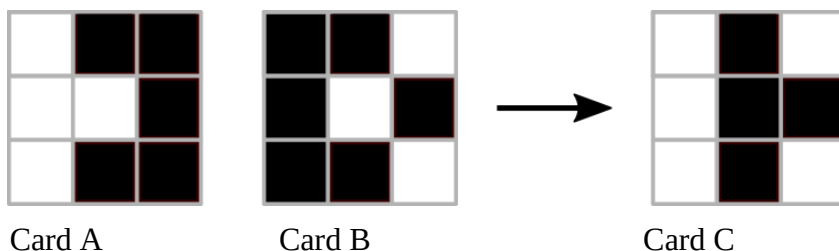
Structura de date utilizata in cadrul acestei intrebari este denumita o “lista legata” ce poate contine un numar arbitrar de elemente. O lista legata reprezinta o colectie liniara de date constand intr-un element si un punct de referinta (legatura) catre urmatorul element. Primul element din lista este foarte important intrucat intreaga lista porneste de la el si este singurul punct care face legatura cu intreaga lista.

Reteta din aceasta intrebare reprezinta o lista legata. Ingredientele reprezinta elementele listei si fiecare foaie de hartie reprezinta legatura catre urmatorul element din lista. Primul element reprezinta ingredientul la care nu se face referire in nicio foaie de hartie, dar care contine langa el o foaie de hartie.

Beneficiul unei astfel de liste consta in faptul ca ea poate stoca elemente de diverse tipuri si dimensiuni.

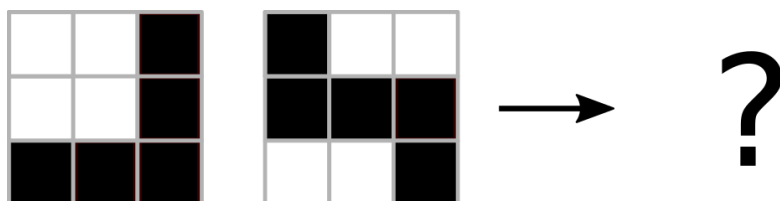
Picteaza in negru

Prin combinarea cardului A cu cardul B, se obtine cardul C.



Intrebare:

Cate celule negre va avea cardul obtinut din combinarea celor 2 carduri de mai jos?



Raspuns:

3

Explicatie:

Combinarea cardurilor respecta urmatoarea regula: atunci cand culoarea celulelor corespondente este aceeași, culoarea rezultata este negru. In caz contrar, culoarea rezultata este alb.



Relationarea cu gandirea computatională:

Competente - Abstractizare (AB), Gandire algoritmica (AL), Evaluare (EV)

Domeniu - Algoritmi si programare

Notiuni – Algebra booleana

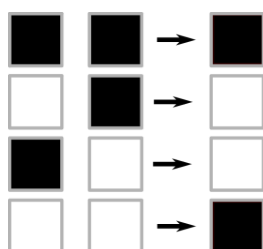
Un circuit boolean este un exemplu de model matematic computational. O echivalenta reprezinta una din operatiile booleene de baza. Daca celula alba este interpretata ca 0 sau FALSE si celula neagra este interpretata ca 1 sau TRUE, aceasta operatie poate fi descrisa astfel:

$1 \oplus 1 \rightarrow 0$

$0 \oplus 1 \rightarrow 1$

$1 \oplus 0 \rightarrow 1$

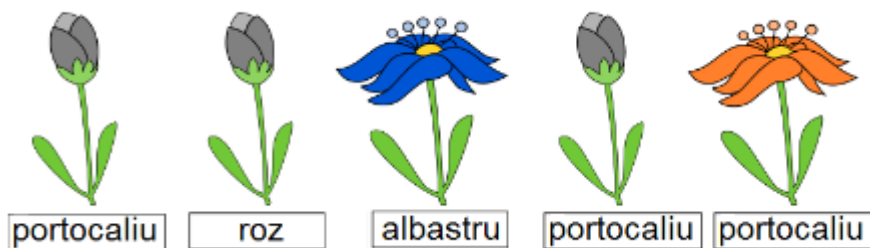
$0 \oplus 0 \rightarrow 0$



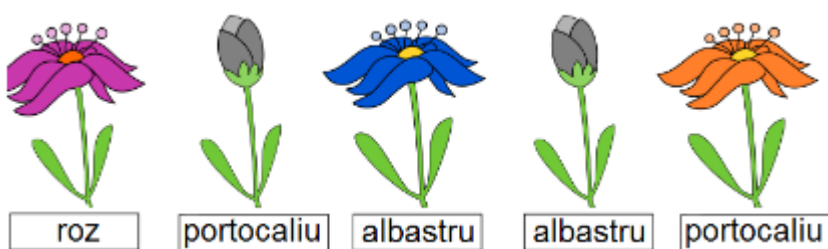
Culoarea florilor

Jane se joaca un joc pe computer. Computerul a selectat in mod secret culorile pentru 5 muguri de floare. Culorile disponibile sunt albastru, portocaliu si roz. Aceste culori nu se vor modifica in timpul jocului.

Jane a ales culoarea pentru fiecare mugur de floare dintr-o lista derulanta si a apasat butonul *Infloreste*. Mugurii de floare, a caror culoare a fost ghicita in mod corect, s-au deschis. Alti muguri insa nu s-au deschis (vezi imaginea de mai jos).



Jane a schimbat apoi culoarea anumitor flori si a apasat din nou butonul *Infloreste*. Rezultatul este afisat in imaginea de mai jos.



Intrebare:

Ce culori a ales computerul pentru flori?

- A. albastru roz albastru portocaliu portocaliu
- B. roz albastru albastru albastru portocaliu
- C. roz albastru albastru roz portocaliu
- D. roz roz albastru roz portocaliu

Culoarea florilor

Raspuns corect:

C. roz albastru albastru roz portocaliu

Explicatie:

Dupa cele 2 incercari ale lui Jane, sunt 3 flori inflorite. Ca urmare, se poate vedea deja culoarea aleasa de computer pentru prima, a treia si a cincea floare. Culoarea primei flori este roz, ca urmare raspunsul A) nu poate fi corect.

Pentru cea de a doua floare, Jane a ales roz pentru prima incercare si aceasta nu a inflorit; apoi a ales portocaliu si nici atunci nu a inflorit. Avand in vedere ca exista doar trei culori disponibile, culoarea celei de a doua flori este albastru. Acest lucru elimina si raspunsul D).

In mod similar, Jane a ales portocaliu si albastru pentru a patra floare si aceasta tot nu a inforit, deci culoarea acesteia este roz. Acest lucru elimina si raspunsul B)

Ca urmare, raspunsul C) este cel corect.

Relationarea cu gandirea computationala:

Competente - Evaluare (EV), Generalizare (GE)

Domeniu - Algoritmi si programare

Stabilirea consecintelor unor evenimente care au avut sau nu loc este o abilitate importanta pentru rezolvarea diferitelor tipuri de probleme. Aceasta intrebare este o versiune simplificata a jocului Mastermind. Este o versiune simplificata intrucat, dupa fiecare incercare, jucatorul primeste informatii complete despre toate florile. Daca la fiecare incercare jucatorul alege o culoare diferita pentru florile care nu au inflorit inca, atunci in a treia incercare jucatorul poate ghici intotdeauna culoarea tuturor florilor.

Potiuni magice

Castorul Betaro a descoperit 5 tipuri de potiuni magice noi cu urmatoarele efecte:

- o potiune face urechile mai lungi;
- alta potiune face dintii mai lungi;
- alta potiune increteste mustatile;
- alta potiune transforma nasul in alb;
- ultima potiune tranforma ochii in alb.

Castorul Betaro pune fiecare potiune magica intr-un recipient separat. Mai exista un recipient suplimentar ce contine apa pura. Ca urmare, exista 6 recipiente etichetate de la A la F. Cu toate acestea, castorul a uitat sa noteze ce potiune magica contine fiecare recipient.

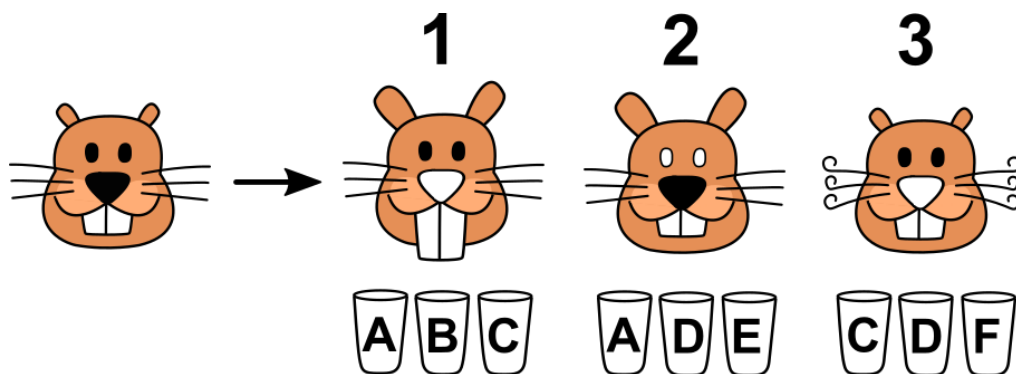


El a realizat urmatoarele experimente pentru a identifica potiunea magica din fiecare recipient.

Experimentul 1: Daca combina continutul recipientelor A, B si C, obtine efectul din Figura 1.

Experimentul 2: Daca combina continutul recipientelor A, D si E, obtine efectul din Figura 2.

Experimentul 3: Daca combina continutul recipientelor C, D si F, obtine efectul din Figura 3.



Intrebare:

Care recipient contine apa pura?

Raspuns corect:

Recipientul D

Explicatie:

Solutia 1:

Prin Experimentul 1, se constata ca niciunul dintre recipientele A, B sau C nu contine apa pura intrucat toate cele 3 schimbari au avut efect asupra castorului.

Prin Experimentul 2, recipientele D si E contin fie apa pura, fie potiunea magica ce ii transforma nasul in alb intrucat recipientul A nu contine apa pura, asa cum s-a stabilit la experimentul 1.

Prin Experimentul 3, D si F contin fie apa pura, fie potiunea magica care increteste mustatile intrucat recipientul C nu contine apa pura, asa cum s-a stabilit la experimentul 1.

Ca urmare, recipientul D contine apa pura.

Solutia 2:

Experimentul 1 are 3 efecte, Experimentele 2 si 3 au fiecare cate 2 efecte. Ca urmare, nu exista apa pura in experimentul 1 si exista exact un singur recipient cu apa in Experimentele 2 si 3. Singurul recipient comun intre Experimentele 2 si 3 este recipientul D. Ca urmare, recipientul D contine apa pura.

Relationarea cu gandirea computationala:

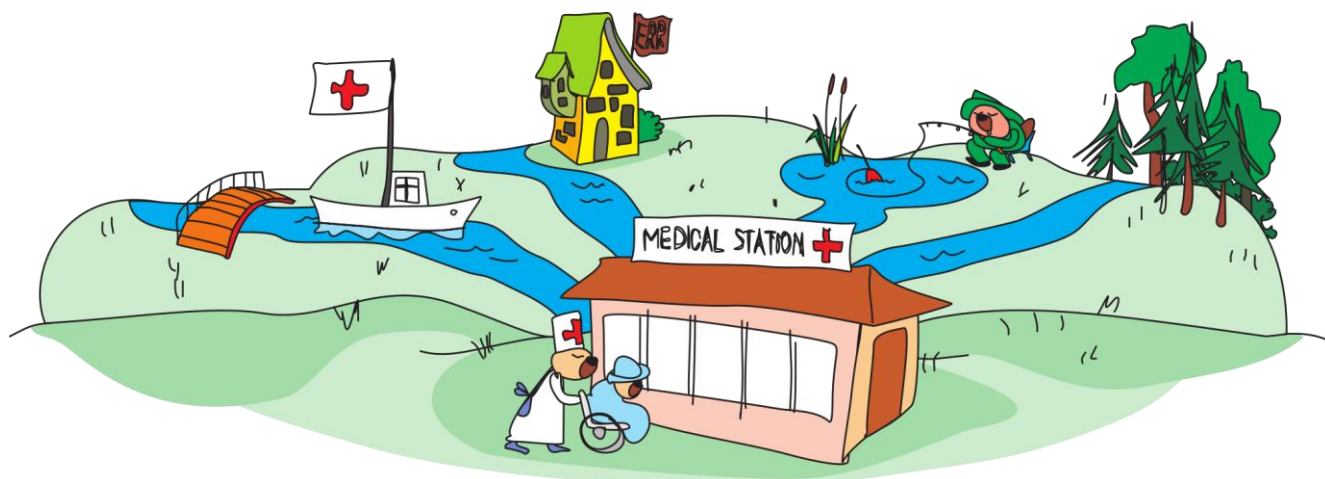
Competente – Gandire algoritmica (AL), Evaluare (EV)

Domeniu - Algoritmi si programare

In aceasta intrebare, exista o colectie de fapte din care trebuie extrase noi informatii. Acest lucru poate fi realizat utilizand rationamente logice. Rationamentele logice joaca un rol extrem de important in Informatica. Cea mai mica unitate cu care lucreaza un computer se numeste bit si are valorile 1 (true) sau 0 (false). Toate celelalte informatii din computer sunt stocate utilizand o combinatie specifica de biti. Computerul utilizeaza rationamente logice pentru a determina ce decizii ar trebui sa ia si fiecare decizie este bazata pe faptul ca anumiti biti sunt setati la valoarea 1 sau 0.

Aceasta intrebare exploreaza de asemenea notiunile de baza legate de multimi. Se cauta un element din multime care nu se afla in multimea utilizata in Experimentul 1, ceea ce inseamna ca este complementar al A,B,C. Ca urmare, se identifica intersectia Experimentelor 2 si 3 pentru a determina elementul lor comun.

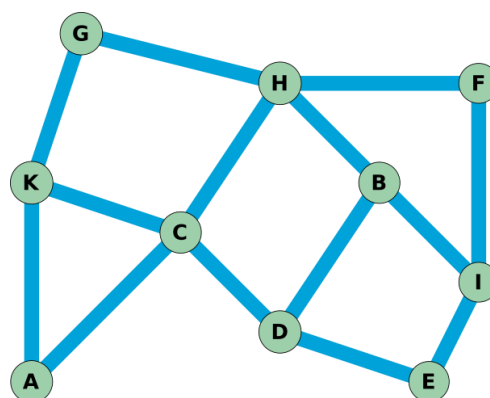
Spitalele



Doctorul Hamid doreste sa construiasca 3 spitale pentru castori.
Spitalele pot fi construite doar in locatiile marcate in harta de mai jos.
Pentru a ajunge la un spital, castorii nu trebuie sa inoate prin mai mult de un canal de la oricare din locatiile de pe harta.

Intrebare:

Alegeti 3 locatii in care Doctorul Hamid isi poate construi spitalele.



Raspuns corect:

Sunt mai multe solutii corecte, una de exemplu fiind locatiile E, H si K:

- Pentru locatiile D, E si I castorii pot inota catre E.
- Pentru locatiile B, C, F, G si H castorii pot inota catre H.
- Pentru locatiile A, C, G si K castorii pot inota catre K.

Alte solutii corecte sunt: A E H, C G I, C H I, C I K, D F K, B I K si C E H.

Explicatie:

Solutiile corecte pot fi gasite prin plasarea unei locatii intr-o pozitie aleatoare si apoi marcarea tuturor locatiilor care sunt accesibile dintr-un singur pas. Apoi, se pozitioneaza urmatoarea locatie si tot asa. Odata ce toate 3 locatiile sunt pozitionate, exista 2 posibilitati: fie este o solutie corecta, fie una sau mai multe locatii nu au fost marcate. In acest ultim caz, eliminati ultima locatie fixata, plasati alta in locul ei si reverificati. Folosind aceasta metoda, puteti determina toate solutiile posibile.

Relationarea cu gandirea computationala:

Competente - Abstractizare (AB), Evaluare (EV)

Domeniu - Date, structuri de date si reprezentari

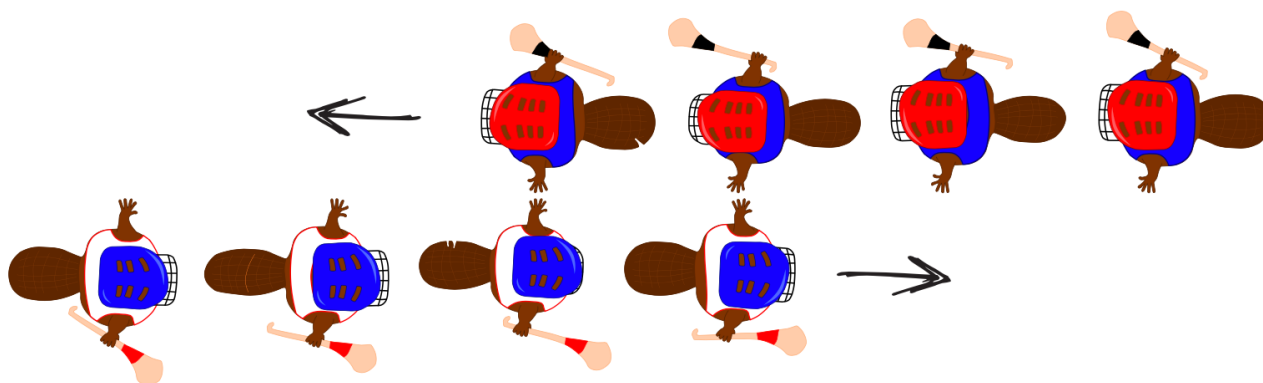
In informatica, acest sistem este generalizat sub forma unui concept numit graf, constand in varfuri (reprezentate de intersectii) si muchii (reprezentate de canalele de apa). Problema generala este de a determina in cadrul unui graf ceea ce se numeste “arbore de acoperire”. Acesta reprezinta un subset de varfuri care acopera intreg graful. Ori de câte ori toate varfurile vecine acestui subgrup sunt adăugate împreună, ele vor acoperi toate varfurile din graf. De obicei, un numar minim de astfel de varfuri este cel mai eficient din punct de vedere al costurilor. In grafuri mai complexe, este foarte dificil de gasit acele subseturi de varfuri care sa fie eficiente din punct de vedere al costurilor. Este necesar un algoritm computerizat pentru a gasi o solutie.

Metoda de plasare a locatiilor descrisa in explicatia de mai sus poarta denumirea de “backtracking”. Se incearca o solutie si daca nu este corecta, atunci se anuleaza ultimul pas si se incearca altul, in mod sistematic, pana cand se epuizeaza toate posibilitatile. Apoi, se anuleaza penultimul pas, se incearca toate solutiile si tot asa pana cand se gaseste solutia corecta. Aceasta metoda nu este foarte eficienta, insa, pentru acest tip de probleme este destul de fiabila.

■ ■ Hurling

Castorii sunt foarte buni la un joc irlandez numit hurling. De fiecare data cand un joc de hurling se termina, castorii din fiecare din cele 2 echipe se aseaza in linie, unul in spatele celuiilalt si trec pe langa echipa adversa, isi strang mainile si se felicita, spunand "Multumesc pentru acest joc!"

La inceput, doar primul jucator dintr-o echipa da mana cu primul jucator din cealalta echipa. Apoi, primii 2 jucatori isi dau mana (vezi imaginea de mai jos). Acest ritual continua pana cand fiecare jucator dintr-o echipa a dat mana cu fiecare jucator din cealalta echipa.



Intrebare:

In cadrul unui joc de hurling, sunt 15 jucatori in fiecare echipa. Daca fiecarui jucator ii ia o secunda sa dea mana cu adversarul si sa se deplaseze apoi la urmatorul jucator, in cate secunde vor reusi toti jucatorii sa dea mana unii cu altii?

Raspuns corect:

29

Explicatie:

Numarul de strangeri de maini este egal cu lungimea unei linii de jucatori + lungimea celeilalte linii de jucatori – 1.

Daca in fiecare echipa ar exista un singur jucator, atunci dupa 1 secunda ar termina sa isi stranga mainile. Daca fiecare echipa ar fi formata din 2 jucatori, in prima secunda si-ar strange mainile primii jucatori din fiecare echipa. In a doua secunda, primul jucator al fiecarei echipe si-ar strange mana cu al doilea jucator al echipei adverse, iar in a treia secunda, si-ar strange mana al doilea jucator din prima echipa cu al doilea jucator din a doua echipa. Asta inseamna in total 3 secunde..

Avand 15 jucatori in fiecare echipa, numarul total de secunde necesar este $15 + 15 - 1 = 29$.

Relationarea cu gandirea computationala:

Compente – Gandire algoritmica (AL)

Domeniu – Procese computerizate si hardware

Notiuni – Programare paralela

Aceasta intrebare reprezinta o ilustrare a paradigmei de programare paralela, cunoscuta sub numele de “pipeline processing”. Programarea paralela este o metoda extrem de eficienta de a utiliza mai multe procesoare sau computere pentru a lucra impreuna la rezolvarea rapida a unei sarcini comune, insa acest proces poate dura destul de mult, nefiind eficient (in problema de mai sus, jucatorii din coada fiecarei linii trebuie sa astepte pana le vine randul sa dea si ei mana cu jucatorii adversi).

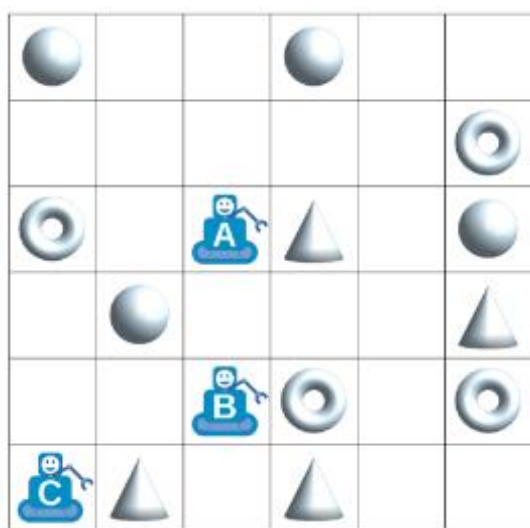
Analiza timpului de rulare a unui algoritm reprezinta o parte sofisticata a informaticii, denumita teoria complexitatii. La aceasta intrebare, fiecare echipa este formata din 15 jucatori si se poate deduce ca timpul de “rulare” al algoritmului de strangere a mainilor este de 29 de secunde. Cu toate acestea, in teoria complexitatii, ar trebui sa se masoare timpul de rulare al algoritmului, indiferent de dimensiunea echipei. Ca urmare, se poate concluziona ca algoritmul de strangere a mainilor dureaza $2N-1$ secunde, unde N reprezinta numarul de jucatori (orice numar natural mai mare sau egal cu 1).

Directii concurente

Intr-un depozit, trei roboti lucreaza mereu ca o echipa.

Atunci cand echipa obtine un simbol de directie (N, S, E, V), toti robotii se muta cu o casuta in acea directie, toti in acelasi timp. Dupa ce se deplaseaza conform unei liste de simboluri de directie, fiecare robot culege orice obiect gaseste in casuta destinatie.

De exemplu, daca dam echipei lista N, N, S, S, E, atunci robotul A va culege un con, robotul B un inel si robotul C un con.



Intrebare:

Ce lista de directii poate fi trimisa echipei astfel incat aceasta sa culeaga exact o sfera, un con si un inel?

- A. N, E, E, E
- B. N, E, E, S, E
- C. N, N, S, E, N
- D. N, E, E, S, V

Directii concurente

Raspuns corect:

B. N, E, E, S, E

Explicatie:

A. Daca lista de directii este N, E, E, E atunci robotul A va culege un inel, robotul B va culege un con si robotul C va culege un inel. Nu a fost culeasa nicio sfera, deci acesta este un raspuns incorect.

B. Daca lista de directii este N, E, E, S, E atunci robotul A va culege o sfera, robotul B va culege un inel si robotul C va culege un con. Avand in vedere ca exista un obiect din fiecare tip, acesta este raspunsul corect.

C. Daca lista de directii este N, N, S, E, N atunci robotul A va culege o sfera, robotul B va culege un con si robotul C va culege o sfera. Nu a fost cules niciun inel, deci acesta este un raspuns incorect.

D. Daca lista de directii este N, E, E, S, V atunci robotul A va culege un con, robotul B va culege un inel si robotul C va culege un con. Nu a fost culeasa nicio sfera, deci acesta este un raspuns incorect.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Evaluare (EV)

Domeniu – Procese computerizate si hardware

Notiuni – Programare paralela

Atunci cand robotii sau computerele lucreaza impreuna in acelasi timp la indeplinirea unei sarcini, se spune ca lucreaza in paralel.

Daca exista un numar mic de roboti sau computere care lucreaza in paralel, i se poate da fiecaruia o lista diferita de instructiuni. Cu toate acestea, daca exista mii sau milioane de computere ce lucreaza impreuna, nu este realist sa se scrie instructiuni separate pentru fiecare computer in parte. Este necesar sa se dea aceleasi instructiuni unor grupuri mari de computere. De exemplu, supercomputerul Tianhe-2 are un procesor cu 3 milioane de nuclee separate care pot fi utilizate in paralel la rezolvarea unei singure probleme complicate.

In aceasta intrebare, robotii lucreaza in acelasi spatiu si instructiunile lor trebuie realizate cu grija astfel incat acestia sa nu se ciocneasca intre ei sau sa se blocheze unul pe celalalt. Pregatirea instructiunilor paralele intr-o astfel de situatie reprezinta un aspect extrem de dificil al informaticii, purtand denumirea de programare concurenta.

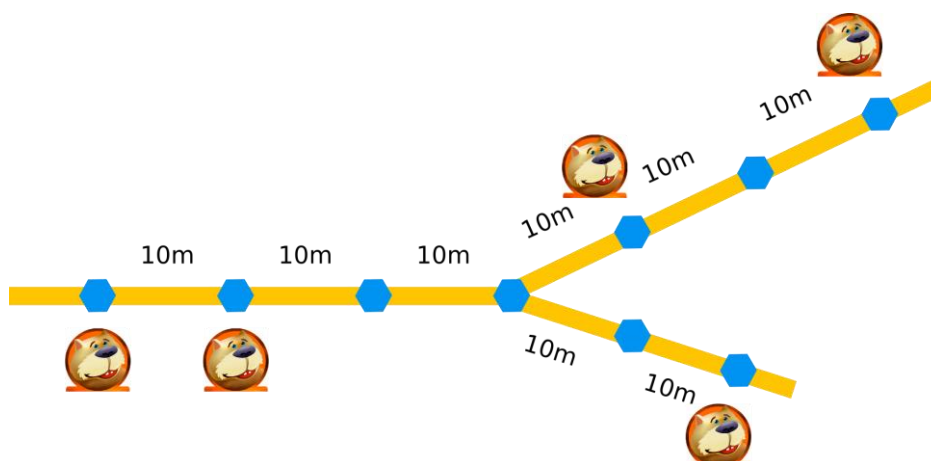
Statia de autobuz

Vizuinele celor 5 castori sunt prezentate in harta de mai jos.

Castorii vor sa plaseze o statie de autobuz in unul din locurile marcate pe harta cu hexagoane albastre.

Toate hexagoanele sunt la distanta de 10m unul de celalalt.

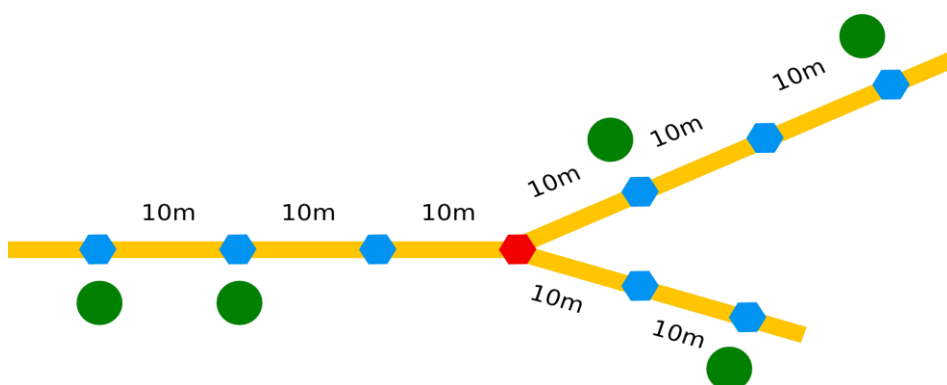
Castorii decid ca suma distantelor de la vizuinile lor si pana la statia de autobuz sa fie minima.



Intrebare:

Executati click pe cea mai buna locatie in care ar trebui plasata statia de autobuz.

Raspuns corect:



Statia de autobuz

Explicatie:

O metoda de rezolvare a acestei intrebari ar fi sa se incerce toate cele 9 posibile locatii si sa se calculeze suma distantelor, insa acest lucru implica efectuarea unui volum mare de calcule. Pentru a reduce volumul calculelor efectuate, putem generaliza solutia.

Daca se presupune ca se plaseaza statia de autobuz la intersectia celor 3 drumuri, suma tuturor distantelor de la viziune la statia de autobuz este:

$$30 + 20 + 10 + 30 + 20 = 110$$

Atunci cand statia de autobuz este mutata cu x metri spre stanga, primele 2 distante se vor reduce cu x metri, iar urmatoarele 3 distante vor creste cu x metri:

$$(30-x) + (20-x) + (x + 10) + (30 + x) + (x + 20) = 110 + x$$

Acelasi rezultat se va obtine daca statia de autobuz va fi pozitionata la o distanta de x metri spre dreapta, pe drumul care se ramifica din intersectie spre dreapta-sus:

$$(30 + x) + (20 + x) + (x + 10) + (30 - x) + (-x + 20) = 110 + x$$

Daca statia de autobuz va fi pozitionata la o distanta de x metri spre dreapta, pe drumul care se ramifica din intersectie spre dreapta-jos:

$$(30 + x) + (x + 20) + (10 + x) + (x + 30) + (20 - x) = 110 + 3x$$

Relationarea cu gandirea computationala:

Competente - Descompunere in factori primi (DE), Evaluare (EV)

Domeniu - Date, structuri de date si reprezentari

Aceasta este o problema clasica de determinare a centrului unui arbore. Este de asemenea si o problema de optimizare. In domeniul informaticii, multe probleme implica determinarea costului minim/maxim sau valorii minime/maxime ale unei functii pentru a economisi bani, pentru a utiliza un nivel minim de resurse sau pentru a determina varianta care este cel mai putin consumatoare de timp.

Aceasta problema se intalneste in viata reala in cazul sistemelor de transport in comun: controlorii de trafic urban trebuie sa ia astfel de decizii pentru optimizarea sistemului de transport (metrou, autobuze, trenuri). In informatica, se aplica diferiti algoritmi pentru determinarea raspunsului optim. Astfel, se poate utiliza unul din cei 3 algoritmi de mai jos:

- 1) O cercetare exhaustiva a tuturor optiunilor (pentru retele mici formate dintr-un numar mic de varfuri). Pentru retele mai mari (care contin mii de locatii) aceasta solutie nu este fezabila
- 2) Daca locatiile formeaza un singur rand, se calculeaza locatia mediana, care va fi optima.
- 3) Nodul optim are proprietatea ca, daca se masoara distanta de la acesta pana la fiecare din vecinii lui, se constata ca pe aceasta distanta locuiesc mai putin de jumatate din numarul total de castori. Pentru optimizarea acestui criteriu, se poate utiliza o solutie de programare dinamica pentru identificarea raspunsului corect intr-un timp cat mai scurt.

Gaseste hotul

O, NU! Faimosul Diamant Albastru a fost furat astazi din muzeu: un hot l-a inlocuit cu o imitatie ieftina de culoare verde.



Au fost 2000 de persoane care au vizitat astazi expozitia. Ei au intrat in camera diamantului pe rand, unul cate unul.

Inspectorul Bebro trebuie sa gaseasca hotul interogand cativa vizitatori.

El are lista cu toti cei 2000 de vizitatori, in ordinea in care au intrat in camera.

El va pune fiecarei persoane aceeasi intrebare: *Diamantul avea culoarea verde sau albastra cand l-ai vazut?*

Fiecare persoana va spune adevarul, cu exceptia hotului, care va spune ca diamantul era deja verde.

Intrebare:

Inspectorul Bebro este foarte inteligent si va folosi o strategie in care numarul de persoane interogate sa fie cat mai mic posibil.

Care dintre urmatoarele afirmatii poate fi facuta de inspector, fara ca acesta sa minta?

- Iti garantez ca voi gasi hotul interogand mai putin de 20 de persoane.
- Interogand 20 de persoane nu va fi suficient (poate doar daca am noroc), insa pot gasi cu siguranta hotul interogand mai putin de 200 de persoane.
- Asta este o sarcina grea: voi fi nevoit sa interoghez cel putin 200 de persoane si cel mult 1999.
- Nu pot promite nimic. Daca sunt de-a dreptul ghinionist, este posibil sa trebuiasca sa interoghez fiecare vizitator.

Raspuns corect:

Iti garantez ca voi gasi hotul interogand mai putin de 20 de persoane.

Explicatie:

In mod surprinzator, inspectorul Bebro trebuie sa investigheze un numar foarte mic de vizitatori, inumatatind in permanenta lista folosind urmatoarea metoda:

Vizitatorii de la 1 la 2000 - dupa ordinea in care au intrat in camera. Inspectorul Bebro il intreaba pe vizitatorul 1000 ce culoare avea diamantul pe care l-a vazut.

Daca raspunsul vizitatorului 1000 este “albastru”, atunci hotul a venit dupa vizitatorul 1000 (intre 1001 si 2000). Daca raspunsul vizitatorului 1000 este “verde”, atunci hotul a fost in primii 1000 de vizitatori. (De mentionat ca hotul poate fi chiar vizitatorul cu numarul 1000!). In ambele cazuri, numarul vizitatorilor suspecti este redus de la 2000 la 1000, deci injumatatit.

Apoi inspectorul Bebro pune aceeasi intrebare persoanei “din mijlocul” listei ramase (vizitatorul numarul 1500 in primul caz, vizitatorul 500 in al doilea caz). Acest lucru ii va permite din nou sa injumatateasca numarul suspectilor.

Continuand in aceeasi maniera, el va reduce numarul de suspecti la 500, 250, 125, 63, 32, 16, 8, 4 si in final 2. Atunci cand raman 2 suspecti, el va pune intrebarea primului suspect. Daca acesta va raspunde “verde”, el este hotul; daca nu, celalalt suspect este hotul. Ca urmare, inspectorul Bebro va gasi hotul dupa intervievarea a numai 11 persoane.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Evaluare (EV)

Domeniu – Algoritmi si programare

Notiuni – Cautare binara

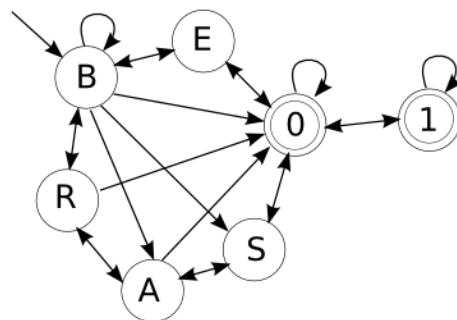
Ideea de a injumatati un set de obiecte in cautarea unui anumit obiect este o tehnica foarte des intalnita in informatica. Probabil cel mai cunoscut exemplu este cautarea binara dupa un anumit element in cadrul unei liste ordonate.

In cazul nostru, s-a folosit o cautare binara pentru a gasi primul element “verde” dintr-o lista ce continea elemente verzi si albastre, toate elementele albastre fiind plasate la inceputul listei. Faptul ca exista un hot care intotdeauna minte este o intorsatura interesanta a lucrurilor.

De mentionat ca problema nu poate fi rezolvata daca nu se cunoaste dinainte ce culoare va spune hotul ca avea piatra.

Rafting

Castorii construiesc plute. Pentru controlul circulatiei fluviale, toate plutele trebuie inmatriculate. Asta inseamna ca fiecare pluta trebuie sa aiba o placuta de inmatriculare cu un text unic. Textul consta in litere si cifre, asa cum este prezentat in desenul de mai jos. Numarul de inmatriculare trebuie sa inceapa cu litera B si sa se termine cu cifra 0 sau 1.



Intrebare:

Care sunt cele 2 placute care nu pot fi inmatriculate?

BB0001 BBB100 BBB011 BB0100 BR00A0 BSA001 BE0S01

Raspuns corect:

BBB100 si BR00A0

Explicatie:

Cea mai buna metoda de rezolvare este pur si simplu urmarirea diagramei si verificarea solutiilor una cate una.

BBB100 este incorect, intrucat partea cu cifre incepe cu 1 (nu se poate ajunge din B in 1) si BR00A0 este incorect intrucat nu se poate ajunge din 0 in A intrucat este o sageata cu un singur sens.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Evaluare (EV)

Domeniu - Date, structuri de date si reprezentari

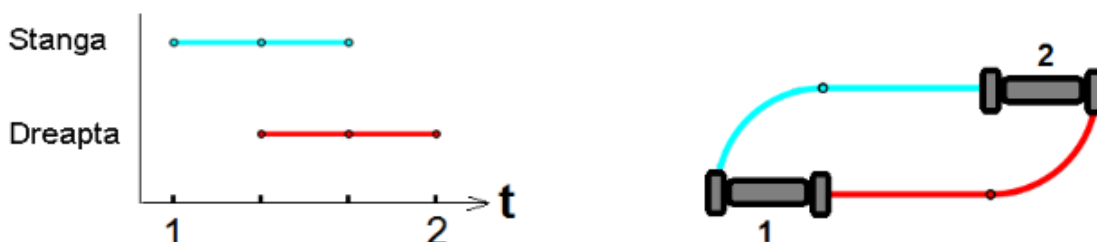
Un automat finit este o parte importanta a teoriei informaticii. Computerele adesea citesc o secventa de caractere sau cuvinte dintr-un document sau dintr-o aplicatie folosind un automat finit. Prin aceasta metoda, se pot cauta secvente de instructiuni, se pot recunoaste sabloane si accepta sau respinge datele in mod corespunzator. In cazul in care caracterele prezentate reprezinta o combinatie permisa, atunci cuvantul este acceptat. In caz contrar, cuvantul este respins.

Jan are un vehicul special de tip Segway. Il deplaseaza apasand pe 2 butoane: un buton albastru (deschis) in partea stanga si un buton rosu (inchis) in partea dreapta. Cand apasa un buton, roata de pe partea respectiva a vehiculului se invarte. Daca sunt apasate ambele butoane in acelasi timp, ambele roti se vor invarti si vehiculul se va deplasa inainte. Daca apasa pe un singur buton, doar o singura roata se va invarti si vehiculul se va roti.



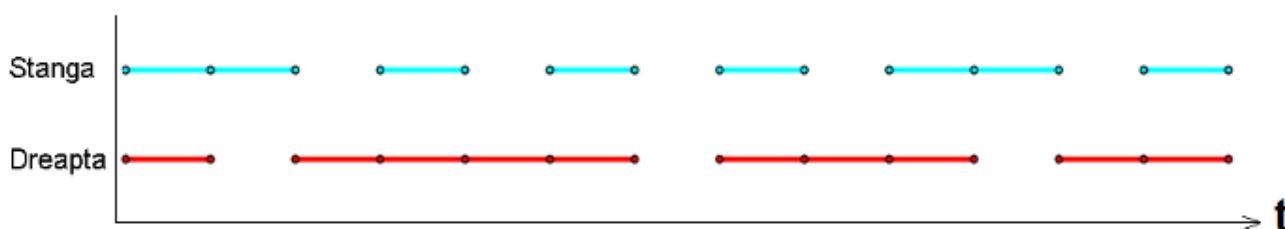
De exemplu:

Imaginile de mai jos arata ce butoane au fost apasate si in ce moment, precum si modul in care vehiculul s-a deplasat din locatia 1 in locatia 2.



Intai a fost apasat butonul albastru si vehiculul s-a deplasat catre dreapta. Apoi, ambele butoane au fost apasate, determinand vehiculul sa se deplaseze inainte. In final, a fost apasat butonul rosu, ceea ce a cauzat intoarcerea spre stanga a vehiculului. Orientarea vehiculului este acum aceeaasi ca si la inceput: cu fata spre zidul din fata.

Intrebare:



Mai sus exista un grafic cu butoanele apasate, pana cand vehiculul s-a lovit de unul din ziduri. La inceput, vehiculul era asezat cu fata spre peretele din fata lui. Spre ce perete era orientat vehiculul la final?

- A. in fata
- B. in spate
- C. la stanga
- D. la dreapta

Raspuns corect:

B. in spate

Explicatie:

Butonul stang a fost apasat de 8 ori, in timp de butonul drept a fost apasat de 10 ori. Ca urmare, vehiculul s-a intors spre stanga de 2 ori si a ajuns cu fata in directia opusa celei de pornire. Deci, el va lovi peretele din spate.

Relationarea cu gandirea computationala:

Competente - Abstractizare (AB), Gandire algoritmica (AL)

Domeniu - Date, structuri de date si reprezentari

Notiuni - Robotica, automate finite

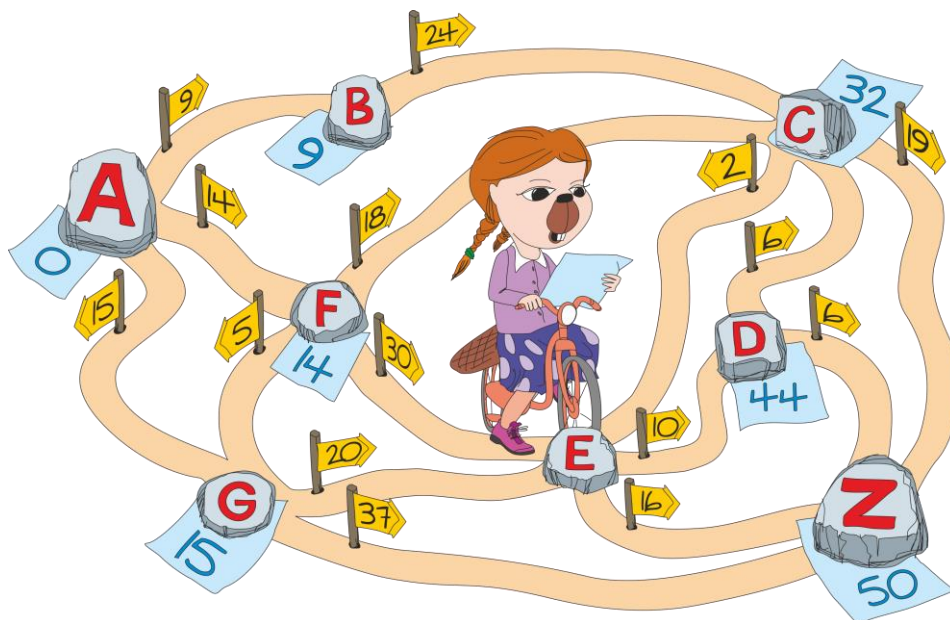
Astfel de vehicule pot fi controlate de la distanta prin folosirea a 2 semnale pentru 2 motoare. Computerul poate genera semnale pentru traversarea automata a unui teren accidentat, in functie de un program si de cunoasterea terenului.

Acest vehicul poate fi considerat un automat finit, ce poate avea un numar finit de stari. Vehicului nu poate fi decat intr-o singura stare, la un anumit moment.

O reprezentare grafica a acestei situatii ofera de obicei mai multa claritate decat descrierea scrisa, ajutand la reducerea numarului de erori.

Cel mai scurt drum cu bicicleta

Cleveria este un castor biciclist. Ea exploreaza drumurile cu sens unic printre comunele etichetate de la A la Z. Toate drumurile au o distanta si o directie, marcate prin steagurile galbene.



Pe parcursul numeroaselor sale excursii, a lasat biletele albastre continand un numar sub o piatra din fiecare comuna.

Intrebare:

Ce reprezinta aceste numere de pe biletelele albastre lasate sub pietre?

- A) distanta cea mai scurta de la punctul A pana intr-o anumita comuna, parcurgand cel mai mic numar de comune
- B) distanta cea mai scurta de la punctul A pana intr-o anumita comuna
- C) distanta cea mai scurta de la punctul A pana intr-o anumita comuna prin virarea la stanga la intersectii, atunci cand este posibil
- D) distanta cea mai scurta de la punctul A pana intr-o anumita comuna prin virarea la dreapta la intersectii, atunci cand este posibil.

Cel mai scurt drum cu bicicleta

Raspuns corect:

B) distanta cea mai scurta de la punctul A pana intr-o anumita comuna

Explicatie:

Pentru a gasi raspunsul corect, trebuie calculate distantele pentru fiecare comuna, conform specificatiilor de la punctele A, B, C si D:

- A) este gresit intrucat ar insemna ca $D = 45$ si $Z = 52$;
- C) este gresit intrucat ar insemna ca $C = 33$, $D = 45$, $Z = 52$;
- D) este gresit intrucat ar insemna ca $C = 51$, $D = 45$, $Z = 52$.

Ca urmare, numerele albastre indica distanta celei mai scurte rute din punctul A pana la o anumita comuna (astfel, raspunsul B este cel corect).

Pentru a determina cea mai scurta ruta, se poate utiliza algoritmul:

1. Se asigneaza o valoare pentru fiecare intersectie prin setarea ei la 0 pentru A (intersectia initiala) si la infinit pentru toate celelalte intersectii.
2. Se seteaza A ca si intersectie curenta.
3. Pentru intersectia curenta, se analizeaza toti vecinii nevizitati si se calculeaza distantele pana la acestia. Pentru fiecare vecin, se compara noua distanta calculata cu valoarea curent asignata si se ia in considerare valoarea mai mica.
4. Dupa ce s-au analizat toti vecinii intersectiei curente, se marcheaza intersectia ca si vizitata, ea nemaifiind apoi reverificata.
5. Se selecteaza intersectia nevizitata care a fost marcata cu cea mai mica distanta, se seteaza ca si "intersectia curenta" si se revine la pasul 3.
6. Daca destinatia finala (Z) a fost marcata ca si vizitata, atunci procesul de incheie. Algoritmul a fost finalizat.

Relationarea cu gandirea computationala:

Competente - Gandire algoritmica (AL)

Domeniu - Date, structuri de date si reprezentari

Notiuni – Drumul critic

Problema determinarii celei mai scurte rute poarta denumirea de "problema drumului critic" si reprezinta o cerinta fundamentala in aplicatiile informatice de zi cu zi. Algoritmii de determinare a celei mai scurte rute sunt folosite pentru a gasi in mod automat indicatii de a ajunge de la o locatie la alta, pe site-uri precum MapQuest sau Google Maps. Algoritmul Dijkstra este unul din cele mai populare algoritmi de determinare a celei mai scurte rute. Acest algoritm va asigna o serie de valori pentru distanta initiala si apoi va incerca sa le imbunatateasca pas cu pas. Structura utilizata pentru a reprezenta locatiile si rutele ce conecteaza aceste locatii poarta denumirea de graf, o foarte importanta structura de date din informatica.

În țara castorilor există o regiune cu mai multe peșteri. Peșterile sunt unite prin cărări; între 2 peșteri există doar o singură cărare, așa cum este afișat în imaginea de mai jos.

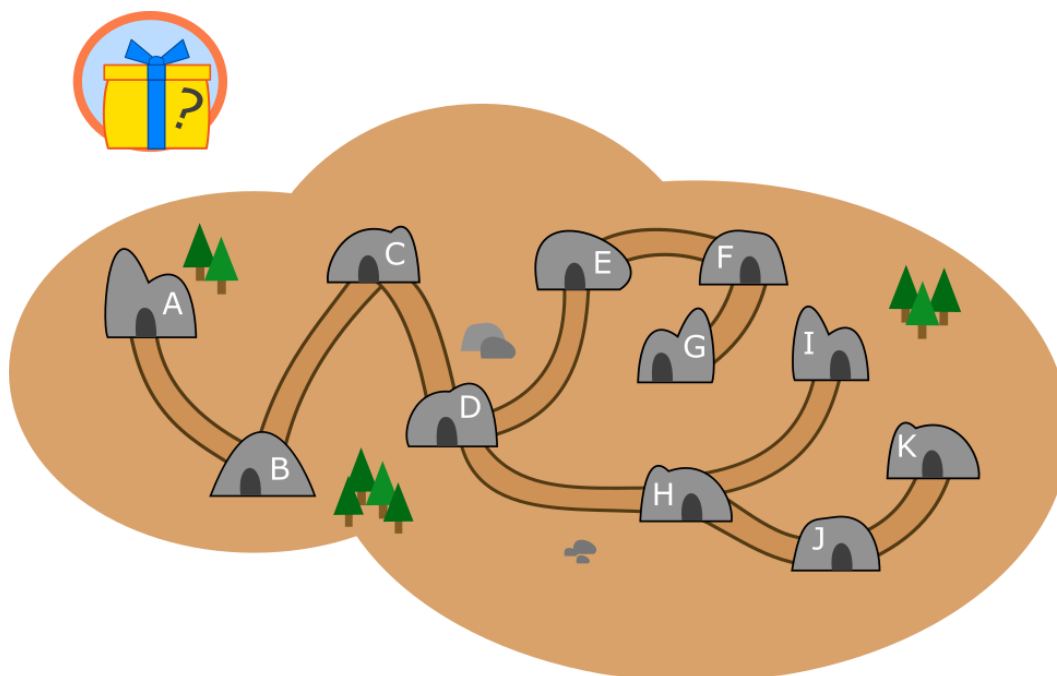
Hale și Serge locuiesc în această regiune și se joacă mereu un joc. Hale ascunde o jucărie în una din peșteri.

Serge vrea să descopere care este peștera cu jucăria.

Pentru aceasta, se folosește de harta de mai jos și poate pune doar întrebări de forma “Jucăria se află în peștera X?”. Dacă a ghicit, Hale răspunde “da”.

Dacă nu a ghicit, îi va spune lui Serge care peștera vecină cu X se află pe cărarea către jucăria ascunsă.

Atunci când Serge știe sigur unde este ascunsă jucăria, jocul se termină și el merge în peștera pentru a o recupera.



Întrebare:

Serge vrea să pună cât mai puține întrebări cu putință pentru a afla unde este ascunsă jucăria. În cel mai rău caz, câte întrebări trebuie să pună Serge pentru a fi sigur că a găsit jucăria?

Raspuns corect:

3

Explicatie:

Se va demonstra mai intai ca, indiferent unde este ascunsa jucaria, Serge poate afla raspunsul in 3 intrebari. Iata un exemplu de intrebari:

Prima intrebare: Jucaria este in pestera D?

Exista apoi 4 raspunsuri posibile si pentru fiecare dintre ele, putem trage o concluzie in cel mult 2 pasi:

- Da – Serge a aflat unde este jucaria – numar total de intrebari:1
- C – Apoi Serge intreaba de pestera B si afla raspunsul - numar total de intrebari:2
- E - Apoi Serge intreaba de pestera F si afla raspunsul - numar total de intrebari:2
- H – Apoi Serge intreaba de pestera H. Fie acesta este raspunsul corect, fie raspunsul primit este I –

numar total de intrebari:2; sau raspunsul primit este J si daca Serge intreaba de pestera J, el gaseste jucaria – numar total de intrebari: 3.

Apoi se va demonstra ca, daca Serge va adresa mai putin de 3 intrebari, vor exista cazuri in care nu se poate determina cu siguranta pestera in care este ascunsa jucaria. Bineinteles, Serge nu poate ghici raspunsul fara sa puna intrebari sau doar adresand o singura intrebare. Se presupune ca Serge adreseaza 2 intrebari. Varianta optima este sa intrebe de pestera D: daca raspunsul este a, b, c, d, e, f sau g, atunci Serge va sti raspunsul adresand doar 2 intrebari. Daca insa Hale raspunde H, atunci, mai avand doar o singura intrebare la dispozitie, Serge nu poate afla cu siguranta daca jucaria este in h, i, j, sau k. Ca urmare, alegand la prima intrebare o alta pestera decat D, nu poate afla raspunsul in mai putin de 3 intrebari.

Relationarea cu gandirea computationala:

Competente - Gandire algoritmica (AL)

Domeniu - Date, structuri de date si reprezentari

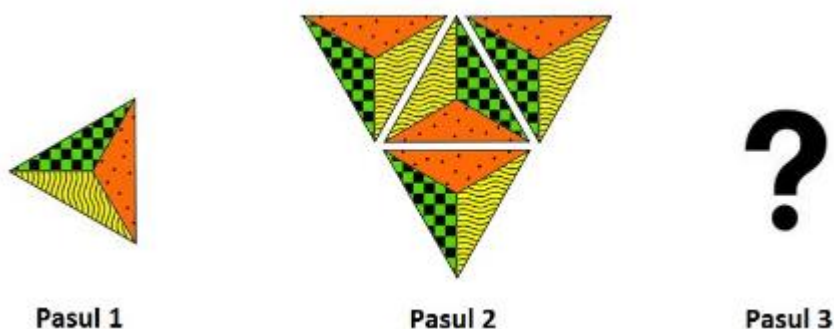
Aceasta intrebare se refera la cautarea informatiilor intr-o structura arborescenta. Construirea si utilizarea algoritmilor corecti pentru extragerea informatiilor este un subiect important in domeniul informaticii.

In aceasta intrebare, pesterile reprezinta nodurile arborelui, iar nodurile care au cel putin 3 rute care realizeaza interconectarea lor sunt de o importanta extrema. Ele descompun arborele in “ramuri” si, daca se cunoaste “ramura” care trebuie urmata pentru a gasi informatia cautata, acest lucru permite restrangerea cautarii si economisirea timpului. Acesta este motivul pentru care structura arborelui si modul de plasare a datelor in interiorul acesteia este crucial: organizarea corecta a datelor permite analiza doar a unei singure ramuri si nu a arborelui in intregime si, in plus, existenta numeroaselor noduri ofera posibilitatea analizei pariale a unei mici portiuni din arbore. Daca, intr-adevar, exista o singura ramura in arbore, atunci trebuie explorat intreg arborele pentru identificarea informatiei cautate. Pe de alta parte, daca pesterile ar fi fost dispuse in forma de stea (o pestera in centru), atunci raspunsul corect se putea determina dupa o singura intrebare si anume cea referitoare la pestera din centru.

Un castor vrea sa creeze un mozaic folosind dale triunghiulare identice.

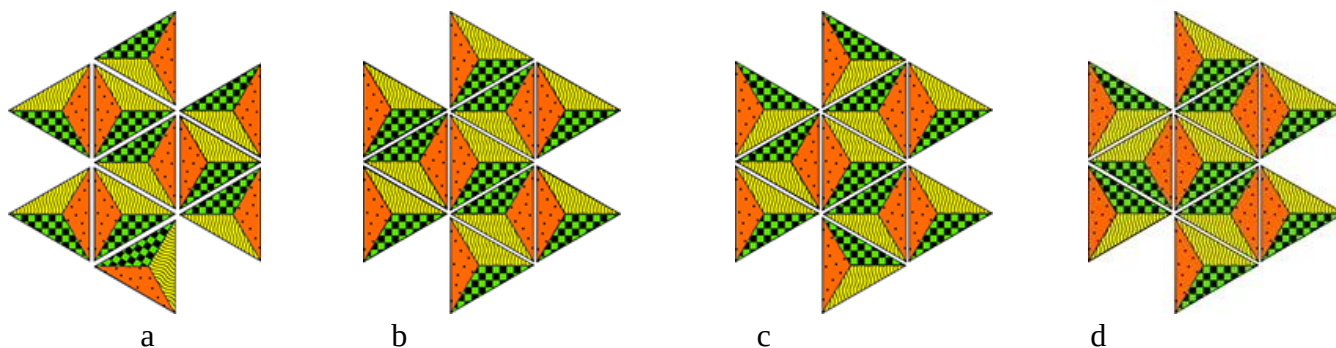
El porneste de la o dala. La pasul urmator roteste dala cu 90 de grade in sensul acelor de ceasornic si apoi adauga cate o noua dala pe fiecare latura a dalei triunghiulare, asa cum este prezentat in imaginea de mai jos.

La pasul 3 roteste intreaga forma obtinuta cu 90 de grade in sensul acelor de ceasornic si apoi adauga noi dale pe fiecare latura, exact ca la pasul anterior.

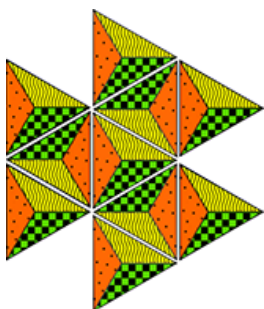


Intrebare:

Care va fi forma finala a triunghiurilor dupa pasul 3?



Rapuns corect:



Explicatie:

Raspunsul a este incorect intrucat dalele nu sunt rotite cu 90 de grade in sensul acelor de ceasornic. Raspunsurile c si d sunt incorecte intrucat dalele nu se potrivesc pe laturile adiacente.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Evaluare (EV), Generalizare (GE)

Domeniu – Algoritmi si programare

Aceasta intrebare presupune executarea unei secvente de instructiuni in mod iterativ. Executarea algoritmilor reprezinta o practica cheie in domeniul informaticii. Aceasta intrebare solicita abilitati de gandire algoritmica, extrem de importanta in executarea unui algoritm. Procesul de evaluare are loc atunci cand elevii iau in considerare diverse solutii in raport cu cele 2 elemente ale secventei de pasi. Intrebarea ofera de asemenea oportunitatea elevilor de a demonstra abilitati de generalizare prin recunoasterea sabloanelor in cadrul variantelor de raspuns propuse.

■ ■ Carti de joc si conuri

Inés are un pachet de carti de joc. Pe fiecare carte de joc este tiparit un numar de la 1 la 9. Pachetele de carti contin mai multe carti de acelasi tip.

Inés plaseaza in fata ei 3 conuri colorate:



Ea intentioneaza sa creeze o stiva de carti de joc sub aceste conuri, cartile fiind asezate cu fata in sus. De fiecare data cand va adauga o noua carte de joc in stiva, aceasta va acoperi tot restul stivei.

Prietena sa, Jules, isi ia notite in timp ce Inés plaseaza pe rand cartile de joc, sub conuri.

Inés incepe prin plasarea unei carti de joc cu numarul 5 sub conul rosu. Jules noteaza: **A <-- 5**

Apoi Inés plaseaza o alta carte de joc sub conul rosu, peste cartea existenta. Jules noteaza: **A <-- 3**

Apoi Inés se uita sub conul rosu si gaseste o carte de joc in pachetul de carti, avand aceeasi valoare cu cartea de sub con.

Ea plaseaza aceasta carte sub conul albastru. Jules noteaza: **B <-- A**

Notitele finale ale lui Jules arata astfel:

A <-- 5

A <-- 3

B <-- A

B <-- 3

A <-- B

B <-- 5

A <-- 6

C <-- B

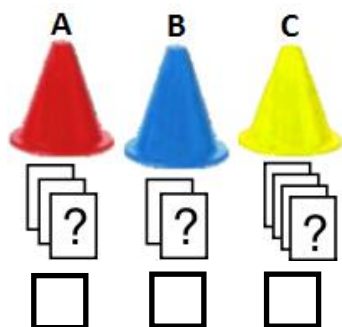
A <-- B

B <-- 1

Intrebare:

Ce carti sunt vizibile atunci cand conurile sunt ridicate?

Scriti numerele corecte in spatiile aflate sub semnele de intrebare.



Carti de joc si conuri

Raspuns corect:

5 1 5

Explicatie:

In tabelul de mai jos este prezentata prima carte din stiva de carti de joc aflata sub fiecare din cele 3 conuri, dupa fiecare instructiune consemnata de Jules.

Notite	A	B	C
A <-- 5	5		
A <-- 3	3		
B <-- A		3	
B <-- 3		3	
A <-- B	3		
B <-- 5		5	
A <-- 6	6		
C <-- B			5
A <-- B	5		
B <-- 1		1	

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL)

Domeniu - Date, structuri de date si reprezentari

Notiuni- Stiva

Aceasta intrebare se refera la modul de functionare a unei aplicatii software. In acest caz, conurile reprezinta o variabila. Variabilele stocheaza o valoare, in acest caz un numar intreg. Pe masura de programul ruleaza, valorile din conuri sunt inlocuite cu altele. Aceste conuri functioneaza de asemenea ca o stiva. Valoarea din partea de sus a stivei este modificata in mod constant. Ultima carte plasata sub con este de fapt prima valoare citita, aceasta fiind o analogie cu stivele din domeniul informaticii.

Apoi, valorile pot fi copiate dintr-un 'con' in altul. In programare, acest lucru poarta denumirea de "transfer prin valoare". De fapt, anumite limbaje de programare sustin copierea referintelor, adica, in locul copierii valorii efective, se va copia referinta. Astfel, in conul rosu se va crea o referinta catre conul albastru, ce ii va spune programului ca valoarea din conul rosu este egala cu cea din conul albastru. Prin urmare, daca se schimba valoarea din conul albastru, se modifica simultan si valoarea din conul rosu!

■ ■ Bilele rosii si albastre

Castorul Emil incearca sa rezolve un puzzle nou pe computerul sau. El trebuie sa aseze o stiva de bile colorate intr-un cilindru.

Regulile sunt urmatoarele:

Bilele trebuie sa fie ori rosii, ori albastre.

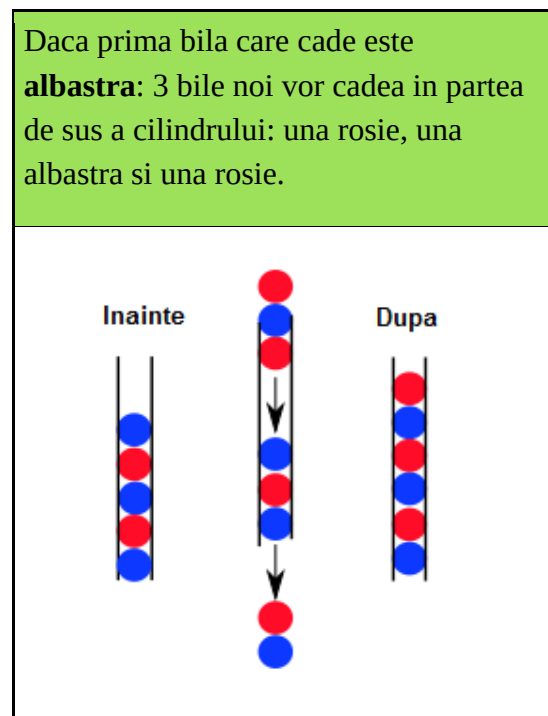
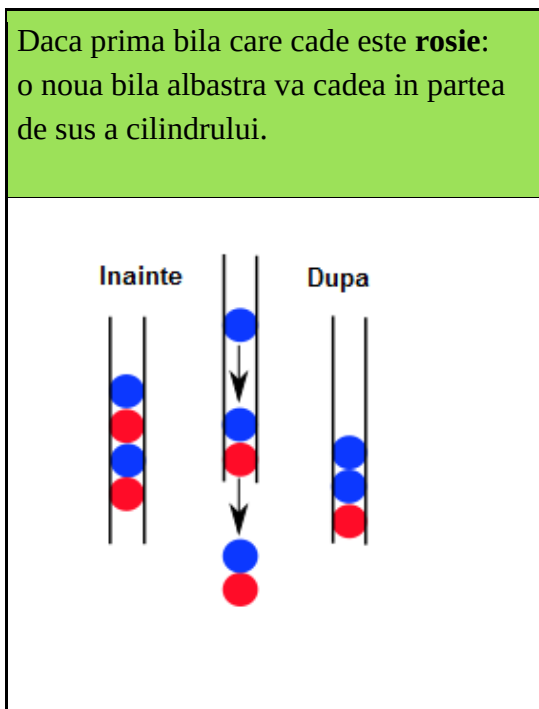
La inceput trebuie sa fie cel putin 3 bile in cilindru.

Scopul este:

Realizarea unei stive de bile care sa nu contina niciodata mai putin de 3 bile in cilindru atunci cand butonul GO este apasat in mod repetat.

La fiecare apasare a butonului GO, ultimele 2 bile din partea de jos a cilindrului cad.

Apoi, in functie de culoarea primei bile care a cazut, se pot intampla 2 lucruri:

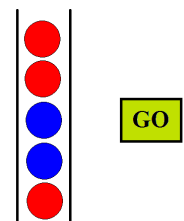


Daca dupa apasarea butonului GO raman in cilindru cel putin de 3 bile, atunci Emil va apasa inca o data pe buton.

Jocul se termina atunci cand in cilindru raman 2 bile sau mai putin.

Exemplu:

Stiva de bile reprezentata in partea dreapta creaza un joc care se incheie dupa 5 apasari ale butonului GO. La final, doar 2 bile albastre vor ramane in cilindru.



Intrebare:

Creati o stiva de bile care consta in doar 3 bile care sa produca un joc fara sfarsit.

Bilele rosii si albastre

Raspuns corect:

Orice stiva formata din 3 bile, in care bila de la baza este albastra.

Explicatie:

Jocul nu se va sfarsi daca Emil incepe de la o stiva formata din 3 bile dintre care cea de la baza este albastra.

In cazul in care stiva de bile contine 3 bile si cea de la baza este rosie, jocul se va sfarsi dupa un singur click. Daca insa stiva de bile contine 3 bile si cea de la baza este albastra, dupa cel mult 5 apasari ale butonului GO, cilindrul va contine stiva RARRAR (R=rosu, A=albastru). Astfel, se prezinta mai jos optiunile existente, afisand culorile de la partea de sus a stivei spre baza:

AAA → RARA → RARRA → RARRAR;

ARA → RARA → RARRA → RARRAR;

RAA → RARR → ARA → RARA → RARRA → RARRAR;

RRA → RARR → ARA → RARA → RARRA → RARRAR.

Astfel, jocul se transforma intr-un ciclu de 4 clickuri intrucat dupa 4 clickuri, aceeasi configuratie RARRAR se repeta:

RARRAR → ARARR → AARA → RARAA → RARRAR.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Descompunere in factori primi (DE)

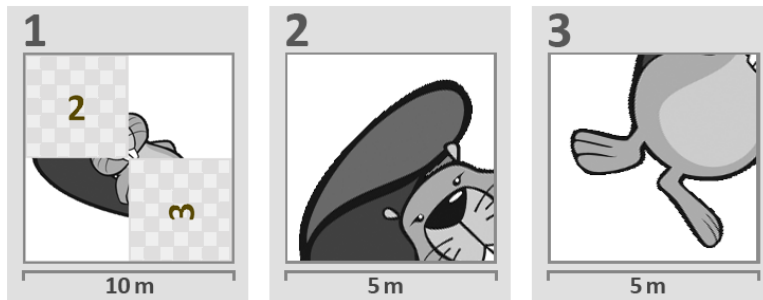
Domeniu – Algoritmi si programare

Acest puzzle reprezinta un model computational ce utilizeaza rescrierea sirurilor. Rescrierea modelelor consta in diferite sisteme de formule care permit inlocuirea unui sir cu altul. Un model poate fi vazut ca un sistem de obiecte (finit), cu un set finit de relatii, definind manipulari si transformari posibile ale acelor obiecte.

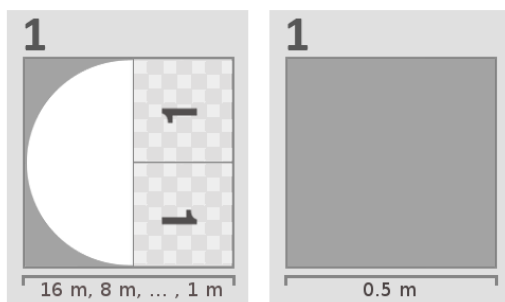
Teoria din domeniul informaticii face referire azi la aceste sisteme folosind termenul de “gramatici independente de context”. De exemplu, un sistem de multiplicare si adunare poate fi definit utilizand simple gramatici independente de context, continand doar cateva reguli. Un alt exemplu este utilizarea pentru definirea precisa si completa a unui limbaj de programare, atat pentru scop educational, cat si pentru implementarea pe diverse dispozitive.

Pictura

Beaver si prietenul sau s-au oferit voluntari sa ajute la renovarea Muzeului de Informatica. Ei trebuie sa picteze o podea de 16 x 16 metri din interiorul uneia dintre camerele cu exponate. Exista un set special de instructiuni pentru pictura. Instructiunile sunt tiparite pe foi de hartie, care fac referire la alte foi de hartie, prin numarul inscris pe ele. Fiecare foaie de hartie are o dimensiune tiparita in partea de jos. Iata mai jos un exemplu al unei astfel de podele pictate in cadrul unui proiect anterior. Pictura finala reprezinta un castor.



Beaver primeste planul pentru noul proiect.



Foaia cu planul contine o referinta catre ea insasi si ambele foi au acelasi numar!

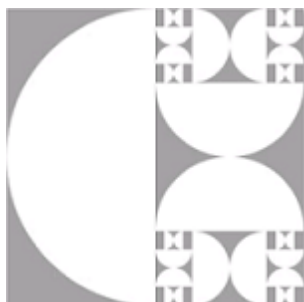
Prietenul lui Beaver il intreaba cum este posibil acest lucru si Beaver raspunde: "Putem face acest lucru. A doua foaie de hartie este importanta pentru ca ne spune cand sa ne oprim."

Intrebare:

Cum va arata pictura finala?



Raspuns corect:



Explicatie:

Se ia in considerare partea stanga a planului. Aceasta descrie modul cum trebuie pictata partea stanga a podelei, cu un semicerc a carui parte rotunda sa fie orientata spre stanga. Pentru partea dreapta a podelei, trebuie sa se foloseasca aceeasi foaie de 2 ori, insa partea pictata trebuie sa aiba cel putin 1 m lungime. Se observa ca orientarea celor 2 cifre de 1 este opusa, ceea ce indica faptul ca foaia trebuie rotita de fiecare data pentru o orientare corespunzatoare.

Se observa de asemenea ca cele 2 margini rotunde ale foii 1 se ating, ceea ce inseamna ca in pictura corecta marginile rotunde ale semicercurilor se vor atinge intotdeauna.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Descompunere in factori primi (DE), Evaluare (EV)

Domeniu – Algoritmi si programare

Notiuni - Recursivitate

Instructiunile care fac referire la ele inele se numesc recursive. Recursivitatea este un concept important in informatica. Solutiile recursive sunt de obicei mai scurte si au un impact mai mare decat alternativele lor, insa sunt uneori mai dificil de inteles. Sabloanele recursive se regasesc in mod frecvent si in natura. Conditiiile de incheiere a recursivitatii, precum cea prezentata mai sus, joaca un rol important in evitarea buclelor fara sfarsit.

Aplicatia de posta electronica

Exista 2 aplicatii de posta electronica: T-Mail si B-Mail.

Atunci cand se redirectioneaza un email catre o alta persoana, T-Mail intotdeauna adauga continutul noului mesaj in partea de sus a conversatiei, in timp ce B-Mail adauga intotdeauna continutul noului mesaj la finalul conversatiei.

4 prieteni si-au trimis mesaje intre ei. Anna si Bella folosesc doar T-Mail.

Chloe si Diane folosesc alternativ cand T-mail, cand B-Mail.

Sa presupunem ca Anna este primul expeditor al unui mesaj catre Chloe.

Chloe utilizeaza B-mail pentru a redirectiona mesajul catre Bella.

In final, Bella redirectioneaza mesajul catre Diane.

Conversatia finala este afisata in imaginea din partea dreapta.

Bella	
Anna	
Chloe	

Imaginea de mai jos arata o alta conversatie. Nu este insa clar cine a trimis primul email.

Tabelul din dreapta arata ce aplicatie de posta electronica foloseste fiecare persoana.

Bella	
Diane	
Anna	
Chloe	
Bella	
Diane	
Chloe	

Utilizator	Program posta electronica
Anna	T-mail
Bella	T-mail
Chloe	T-mail, B-mail
Diane	T-mail, B-mail

Intrebare:

Cine NU ar putea fi expeditorul primului email?

Anna

Bella

Chloe

Oricine ar fi putut trimite primul email

Aplicatia de posta electronica

Raspuns corect:

Anna

Explicatie:

Anna nu ar putea fi expeditorul primului email. Daca Anna ar fi fost primul expeditor, atunci emailul lui Bella nu ar fi aparut sub emailul Annei.

Daca Bella ar fi fost primul expeditor, atunci emailurile redirectionate de la Anna sau Bella ar fi aparut deasupra primului email. Emailurile redirectionate de Chloe sau Diane ar fi fost apoi deasupra sau sub mesajului anterior. Rationamentul este similar daca Chloe sau Diane ar fi fost primul expeditor.

Ca urmare, oricare din celelalte 3 persoane ar fi putut fi primul expeditor.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Descompunere in factori primi (DE)

Domeniu – Comunicatii si retele

Prin intelegerea starii finale a unui proces si a setului de reguli ce guverneaza acest proces, se poate utiliza rationamentul logic pentru a deduce starea initiala. In acest caz, se pot afla posibili expeditori ai primului email, precum si aplicatia de posta electronica utilizata.

Mesajele electronice si forumurile adopta stiluri de publicare: in partea de sus sau in partea de jos. O aplicatie de posta electronica are in mod normal un stil implicit de trimitere a emailurilor si de obicei poate fi modificat de utilizator. In cazul forumurilor, cu toate ca sunt folosite ambele stiluri, fiecare comunitate online adopta stilul pe care il considera corespunzator sau acceptabil.

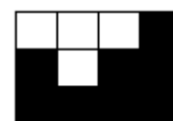
2 scanere codifica o imagine prin transformarea pixelilor ei intr-un cod special. Codul arata numarul tuturor pixelilor consecutivi de aceeaasi culoare (alb sau negru), urmat de numarul tuturor pixelilor consecutivi de alta culoare si tot asa, pornind din coltul din stanga sus si mergand de la stanga spre dreapta, rand cu rand.

Cele 2 scannere utilizeaza metode diferite de a trata capatul unui rand: Scannerul A proceseaza pixelii rand cu rand si apoi reporneste codificarea la inceputul randului urmator. Scannerul B proceseaza pixelii rand cu rand, insa nu reporneste codificarea la inceputul randului urmator.

De exemplu, imaginea din dreapta ar trebuie sa fie reprezentata prin urmatoarele coduri:

Scannerul A: 3,1,1,1,2,4 (3 alb, 1 negru, 1 negru; 1 alb, 2 negru, 4 negru)

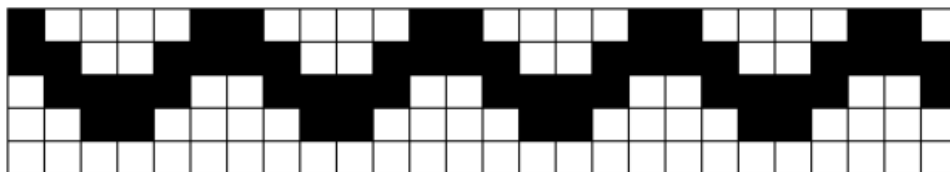
Scannerul B: 3,2,1,6. (3 alb, 2 negru, 1 alb, 6 negru)



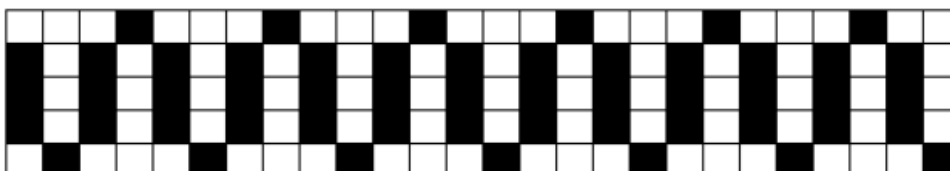
Intrebare:

Care dintre urmatoarele imagini va avea acelasi cod, indiferent ce scanner este utilizat?

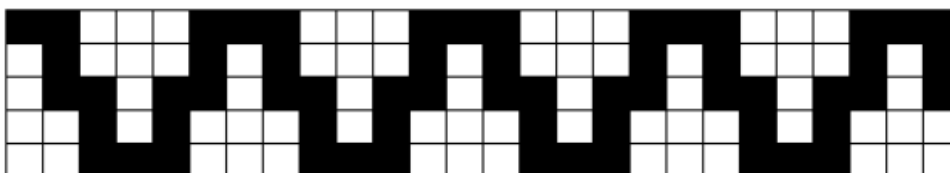
A



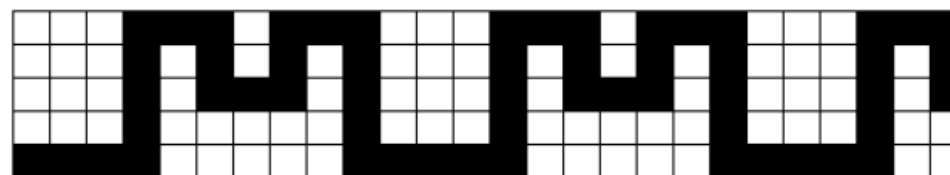
B



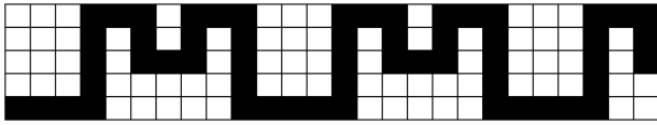
C



D

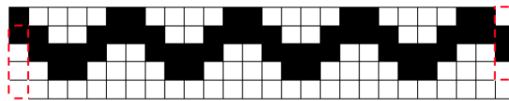


Raspuns corect:



Explicatie:

Diferenta dintre cele 2 metode consta in combinarea sau nu a ultimilor pixeli dintr-un rand cu primii pixeli din randul urmator. Scannerul A nu combina acesti pixeli. Scannerul B combina pixelii doar daca au aceeasi culoare. Daca pixelii nu au aceeasi culoare, codul rezultat de la cele 2 scannere este acelasi. Ca urmare, trebuie identificata o imagine in care ultimul pixel dintr-un rand si primul pixel din randul urmator sunt diferiti, pentru toate randurile:



Trebuie doar comparati cei 4 pixeli de la finalul fiecarui rand cu cei 4 pixeli corespondenti de la inceputul fiecarui rand in fiecare din ele 4 imagini furnizate, pentru a verifica ca ei au culori diferite:



Din cele 4 imagini, singura care intruneste aceasta conditie este imaginea D.

Relationarea cu gandirea computationala:

Competente - Abstractizare (AB), Evaluare (EV)

Domeniu - Date, structuri de date si reprezentari

Un scanner este un dispozitiv care converteste o imagine tiparita intr-una digitala. In timpul scanarii, culoarea si luminozitatea fiecarui pixel citit de senzorul scannerului sunt masurate si stocate sub forma unei valori numerice. Acest process poarta numele de digitizarea imaginii.

Un pixel reprezinta cel mai mic element al unei imagini digitale. Fiecare pixel reprezinta un esantion al imaginii initiale si, cu cat numarul de pixeli este mai mare, acuratetea imaginii creste.

Scannerul A utilizeaza intreruperi de linie pentru a restarta codificarea odata cu trecerea la randul urmator, in timp ce Scannerul B citește pixelii sub forma unei imagini continue. Fiecare metoda are propriul set de avantaje: de exemplu, in cazul unei imagini lungi, se utilizeaza mai putine numere cu scannerul B, dar trebuie codificata de asemenea si dimensiunea imaginii. Acest lucru nu este practic pentru imaginile de dimensiune mica. Aceste compromisuri reprezinta decizii foarte importante care trebuie luate in domeniul informaticii.

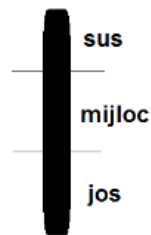
Codul Kix

Oficiul postal al castorilor are coduri postale ce constau in 36 de caractere ('A'..'Z' si '0'..'9').

Pentru ca echipamentele de la Oficiul Postal sa poata citi codurile postale, castorii convertesc aceste coduri postale in coduri Kix.

Intr-un cod Kix, fiecare caracter este reprezentat de un cod cu 2 sectiuni (sus si jos). Ambele sectiuni contin 4 bare verticale. Sectiunea de sus contine doar bare pe mijloc si sus, in timp ce sectiunea de jos contine numai bare pe mijloc si jos.

Tabelul de mai jos afiseaza codurile pentru cateva caractere:



	0	1	2	3	4	5
0						
6						
C						
I						
O						
U						



De exemplu, codul Kix pentru "G7Y0" este

Intrebare:

Un alt cod postal are codul Kix.



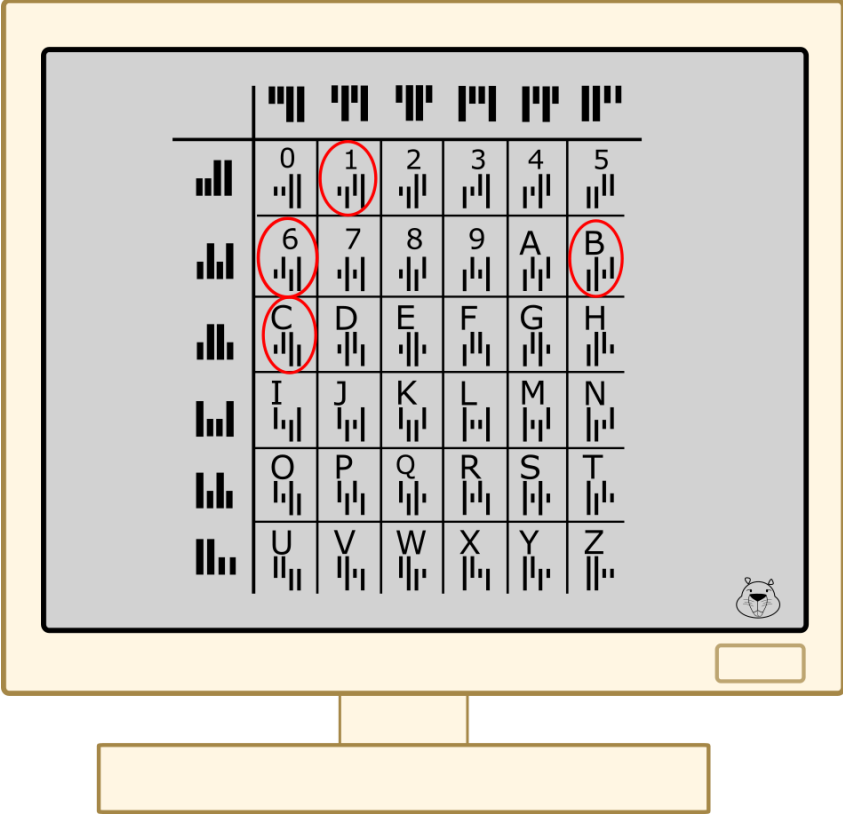
Care este codul postal?

Raspuns corect:

BC16

Explicatie:

Raspunsul poate fi obtinut prin consultarea tabelului si convertirea codului Kix in caracterele corespondente.



	0	1	2	3	4	5
0	0	1	2	3	4	5
1	6	7	8	9	A	B
2	C	D	E	F	G	H
3	I	J	K	L	M	N
4	O	P	Q	R	S	T
5	U	V	W	X	Y	Z

Relationarea cu gandirea computationala:

Competente – Gandire computerizata (AL), Generalizare (GE)

Domeniu - Date, structuri de date si reprezentari

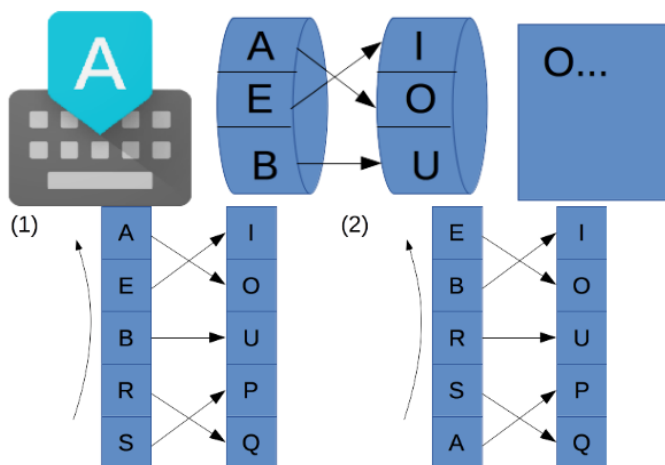
Codurile Kix (folosite de fapt de posta olandeza) reprezinta un exemplu de coduri de bare. Acestea reprezinta coduri ce pot fi citite de catre diverse dispozitive, fiind utilizate in automatizarea selectiei si sortarii plicurilor.

In acest exemplu, informatia din tabel este separata intre sectiunea de sus si cea de jos. Aceasta tehnica este folosita uzual in prezentarea informatiilor.

Castorii joaca un joc impotriva echipei Deavers. Castorii trebuie sa comunice in secret unii cu altii, insa mesajul lor va trece printr-o zona controlata de echipa Deavers. Castorii au decis sa utilizeze un mecanism denumit Enigma pentru a ascunde (cripta) mesajele atunci cand le trimit dintr-o parte in alta.



Mecanismul functioneaza astfel: de fiecare data cand se tasteaza o litera pentru criptare (de ex. “A”), rotorul stang va genera o litera corespondenta in rotorul drept (de ex. “O” pentru “A” la primul pas). Dupa tastarea literei, rotorul stang se va deplasa cu o pozitie in directia indicata de sageata, ajungand in pozitia 2. Cu toate acestea, rotorul drept nu se deplaseaza niciodata. Legaturile dintre cele 2 rotoare (indicate de sageti) vor ramane de asemenea neschimbate. In imaginea de mai jos puteti vedea toate cele 5 litere disponibile.



Intrebare:

Grupul 1 de castori doreste sa trimita mesajul “BEBRAS” grupului 2. Cum va arata mesajul criptat daca se porneste de la pozitia 1?

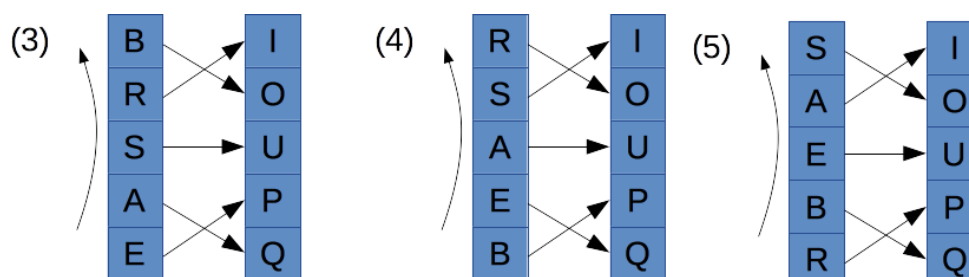
- A.** UOSAEB
- B.** UOUQOP
- C.** UOOOIP
- D.** UOOUPQ

Raspuns corect:

C. UOOOIP

Explicatie:

La pozitia 1 litera B este criptata in U si la pozitia 2 E este criptat in O asa cum este prezentat in imaginea de mai sus. Celelalte transformari pentru literele 3-5 sunt prezentate mai jos. Pentru ultima litera, se foloseste pozitia 1.



La pozitia 3, litera B este criptata in O. Ca urmare, raspunsurile A si B sunt incorecte.

La pozitia 4, litera R este criptata in O, deci raspunsul D este incorect.

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Evaluare (EV)

Domeniu – Comunicatii si retele

Notiuni – Criptologie, criptografie, criptare

Acest mecanism reprezinta o versiune simplificata a masinii Enigma utilizata in timpul celui de al doilea razboi mondial de catre armata germana pentru criptarea comunicatiilor. Codul Enigma a fost spart pentru prima data de polonezi in 1932. Serviciile secrete britanice au incercat in timpul razboiului sa descifreze mesajele codificate produse de aceasta masina. Cei mai multi licentiati in criptografie incep cu o prezentare a mecanismului Enigma. Munca depusa de Alan Turning in spargerea codului Enigma a dus la crearea primului computer.

Castorul Big se joaca impreuna cu castorul Small pe o tabla de joc speciala.

Ei incep din caseta din coloana cea mai din stanga (caseta 5). **Castorul Big incepe jocul.**

El poate decide daca se muta in sus sau in jos.

In sus se va muta in caseta 4.4; In jos se va muta in caseta 5.7.

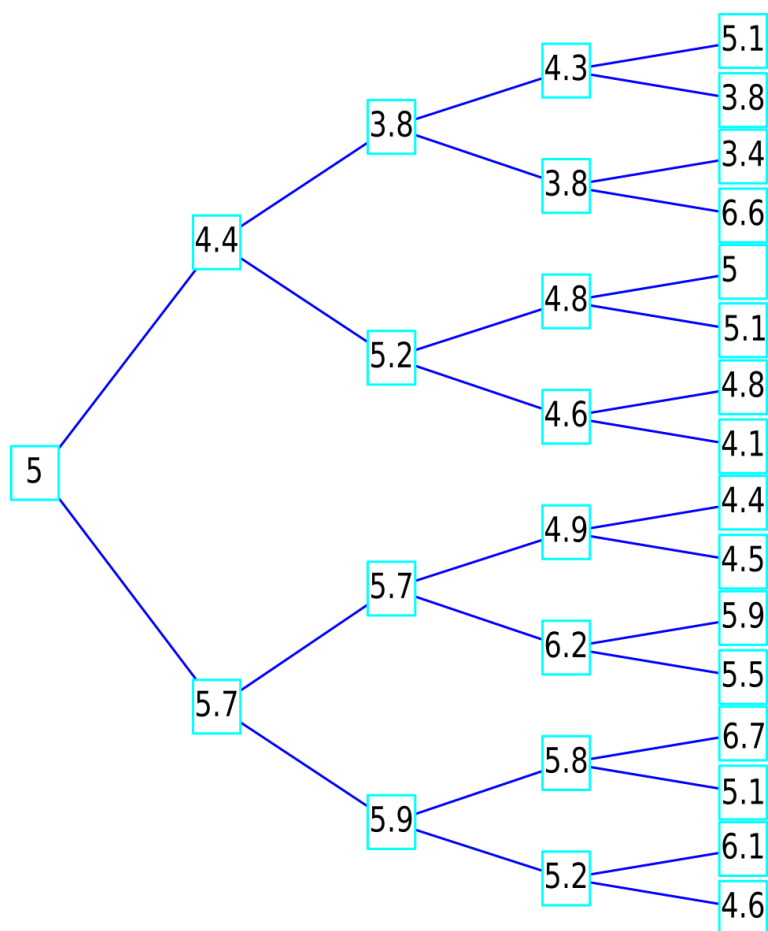
Apoi, este randul castorului Small de a se deplasa in sus sau in jos. Apoi, fiecare jucator face propria mutare pana cand, in final, castorul Small alege o caseta din coloana cea mai din dreapta.

Avand in vedere ca ambii castori pot vedea in permanenta toate numerele de pe tabla de joc, ei isi pot crea niste strategii de joc corespunzatoare.

Intrebare:

Castorul Big joaca astfel incat ultima caseta sa aiba cea mai mare valoare posibila, iar castorul Small joaca astfel incat sa obtina cea mai mica valoare posibila.

Daca amandoi castorii joaca cat pot ei de bine, care va fi numarul din caseta finala?



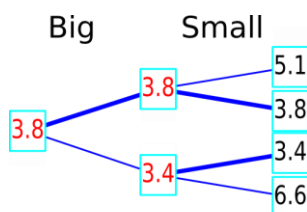
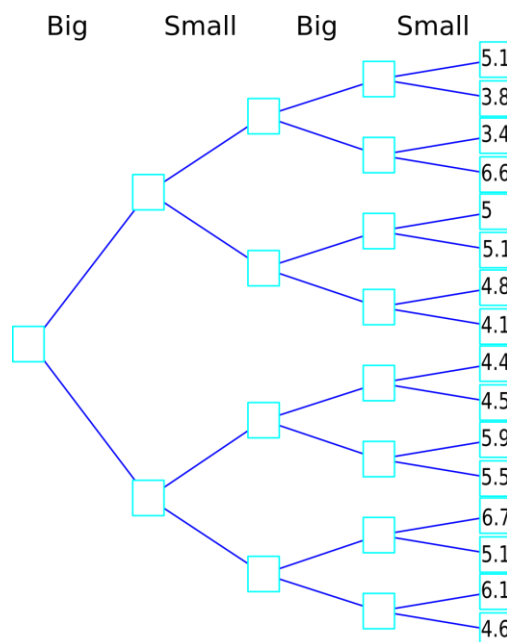
Raspuns corect:

Pozitia finala va fi caseta 5.1, ce mai de jos din partea dreapta.

Explicatie:

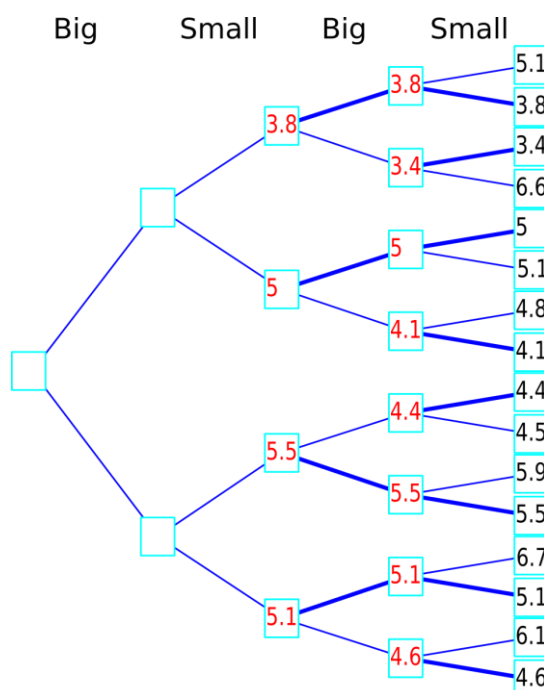
Pentru a ilustra raspunsul corect, numele jucatorilor de la fiecare pas au fost adaugate pe grafic.

Intrucat ne intereseaza doar numerele din extremitatea dreapta, celelalte numere nu sunt necesare – ele au fost plasate in intrebare doar cu scopul de a deruta. Ca urmare, toate numerele, cu exceptia celor din casetele din extremitatea dreapta, pot fi sterse.

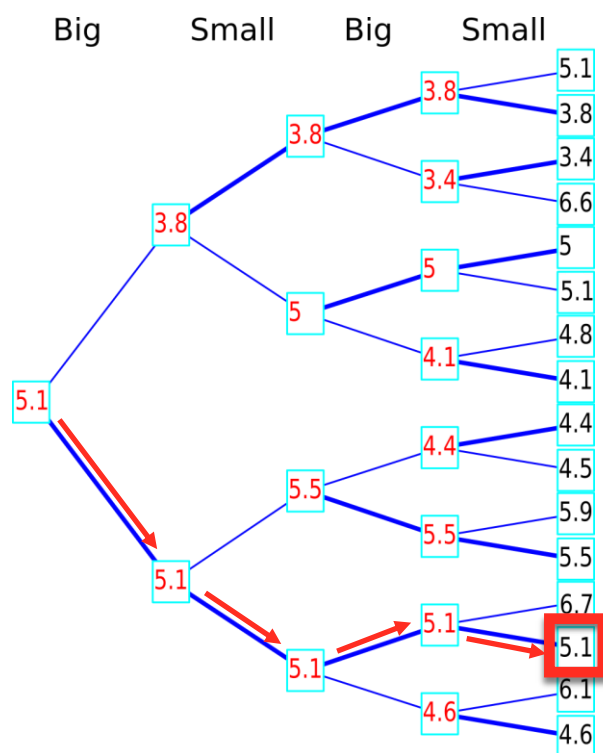


Cea mai inteligenta metoda de a planifica o strategie pentru acest jos este sa pornesti de la mutarea finala. In fiecare casuta goala este scris un numar rosu, ales de jucator, care incearca sa realizeze un joc optim. Pentru o explicatie mai amanuntita, putem folosi doar o parte din grafic. Numerele rosii sunt alese de jucatori, de fiecare data cand le vine randul. Se poate demonstra ca exista un singur numar la care jucatorii pot ajunge.

In partea dreapta puteti vizualiza procesul prin care au fost adaugate mai multe numere.



Solutia finala consta in completarea tuturor casetelor in aceeași manieră și apoi urmărirea drumului de la răspunsul 5.1 înapoi către caseta corectă:



Relationarea cu gândirea computațională:

Competențe – Gândire computerizată (AL)

Domeniu - Date, structuri de date și reprezentări

Notiuni - Graf, arbore binar

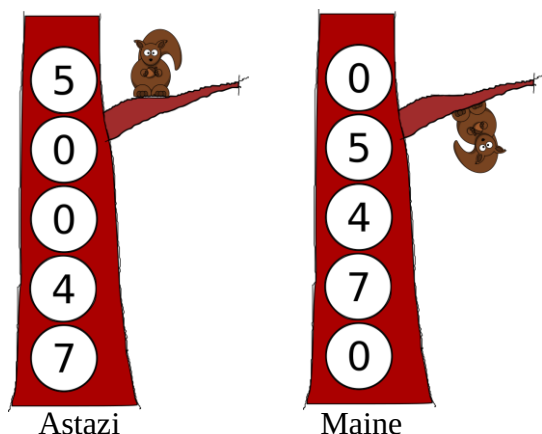
Selectarea elementelor într-un arbore binar este un proces extrem de comun în informatică.

Graful utilizat în această întrebare este o versiune rotită spre dreapta a grafurilor utilizate adesea în turnamentele sportive, ca de exemplu tenis. În astfel de grafuri, numerele din casetele poziționate în extremitatea din dreapta reprezintă 16 jucători ce concurează pentru titlul de campion. Pornind de la dreapta, în fiecare etapă, 2 jucători concurează unul împotriva celuilalt, iar jucătorul castigator avansează spre stânga.

Ca urmare, graful nostru poate descrie un turnament ciudat în care există diferite runde, fiecare cu alte reguli. De exemplu, castigatorul final este cel care are cele mai multe puncte, însă castigatorii semifinalelor sunt jucătorii cu cele mai puține puncte.

Veveritele egoiste

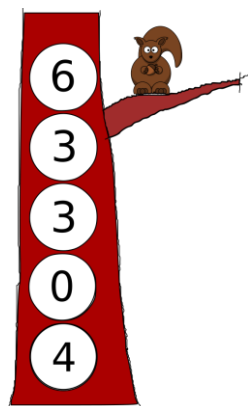
Veveritele egoiste locuiesc in scorburile copacilor. In padure exista un copac cu 5 scorburu mari asezate una peste cealalta si 16 veverite, asa ca ele sunt fortate sa locuiasca impreuna in scorburu. In fiecare zi, fiecare veverita verifica care numar este cel mai mic: numarul veveritelor din scorbura in care se afla, numarul de veverite care locuiesc in scorbura de deasupra sau numarul de veverite care locuiesc in scorbura de dedesubt. In urmatoarea noapte, fiecare veverita se muta in secret in scorbura cu cel mai mic numar. Daca valorile sunt aceleasi, veverita prefera sa ramana in scorbura actuala decat sa se mute in scorbura de mai sus si de asemenea prefera sa se mute in scorbura de deasupra decat in cea de dedesubt.



Ca urmare, de exemplu, daca astazi sunt 5, 0, 0, 4, 7 veverite in scorburu (numarate de sus in jos), atunci maine toate cele 5 veverite din scorbura cea mai de sus se vor muta in scorbura de dedesubt (0 vecini este mai bine decat 4). 7 veverite din scorbura cea mai de jos se vor muta sus (4 vecini e mai bine ca 6) si 4 veverite din penultima scorbura de jos se vor muta deasupra (0 vecini este mai bine decat 3).

Intrebare

Iata situatia initiala a numarului de veverite care locuiesc in scorburu.



Dupa cate zile toate veveritele vor ajunge sa locuiasca impreuna in aceeasi scorbura?

2 3 4 sau niciodata

Veveritele egoiste

Raspuns corect:

3

Explicatie:

Raspunsul corect este 3.

$(6,3,3,0,4) \rightarrow (0,9,0,7,0) \rightarrow (9,0,7,0,0) \rightarrow (0,16,0,0,0)$

Relationarea cu gandirea computationala:

Competente – Gandire algoritmica (AL), Descompunere in factori primi (DE)

Domeniu – Algoritmi si programare

Aceasta problema se refera la inteligenta colectiva (swarm intelligence). Ideea unor astfel de algoritmi este ca se pot rezolva probleme utilizand dispozitive foarte simple, cu conditia ca numarul dispozitivelor sa fie imens. De exemplu, comportamentul furnicilor se bazeaza pe reguli simple si functioneaza independent pentru fiecare furnica in parte. Insa daca numarul furnicilor este mare, ele sunt capabile sa faca lucruri sofisticate precum construirea musuroaielor, cautarea unui drum optim intr-un graf sau chiar taierea frunzelor.

In cazul problemei de fata, avem de a face de asemenea cu un numar mare de dispozitive (reprezentate prin veveritele, ce se comporta conform unor reguli simple). Insa se dovedeste ca, comportamentul lor colectiv este departe de a fi inteligent. Ele vor sa locuiasca intr-o scorbura cat mai spatiosa cu putinta, insa de obicei ajung sa locuiasca toate intr-o singura scorbura!

Morala acestei intrebari este ca fiecare ar trebui sa adopte reguli simple de comportament si ca uneori este mai bine sa cooperezi cu ceilalti decat sa fii egoist.

Un copac frunzos este inconjurat de 2 copaci fara frunzis si 2 palmieri



5 tipuri de banane, de exemplu P, Q, R, S, T sunt plasate in acesti copaci, fiecare tip intr-un copac diferit. O maimuta sare dintr-un copac in altul, pentru a manca cate o banana din fiecare copac. Maimutei ii trebuie

- 3 secunde pentru a sari din copacul frunzos in orice alt copac si a manca o banana,
- 2 secunde pentru a sari din copacul fara frunzis in palmier sau viceversa si a manca o banana,
- 7 secunde pentru a sari intre cei 2 copaci fara frunzis sau intre cei 2 palmieri, evitand copacul frunzos si a manca o banana.

Maimuta sare si mananca urmatoarele tipuri de banane P, Q, S, R, T, R, P.

Intrebare:

Ce tip de banane poate fi in copacul frunzos daca timpul total in care maimuta sare intre copaci si mananca bananele este cel mai mic posibil?

P sau Q sau T

P sau S sau T

Q sau S sau T

Q sau R sau S

Raspuns corect:

P sau S sau T

Explicatie:

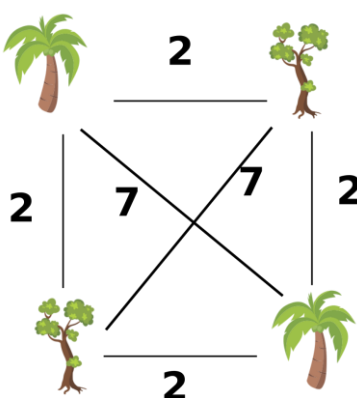
Secventa P,Q,S,R,T,R,P viziteaza toate tipurile de banane, cu 6 sarituri dintr-un copac in altul:

P-Q, Q-S, S-R, R-T, T-R, R-P.

Pentru a avea o durata minima, cele 6 sarituri trebuie sa cuprinda 4 sarituri de cate 2 secunde fiecare si alte 2 sarituri de cate 3 secunde fiecare; astfel, durata totala este de 14 secunde.

R apare in 4 sarituri: S-R, R-T, T-R, R-P; R-T si T-R trebuie sa aiba aceeasi durata, astfel exista 3 sarituri catre si dinspre R, cu timpi posibil diferiti. Prin urmare, R nu poate fi un copac frunzos intrucat cele 3 sarituri ar dura 3x3 secunde, iar secventa nu ar mai avea o durata minima. Acest rationament elimina raspunsul D si acum se poate constata ca T poate fi cu siguranta un copac frunzos. In acest caz, T-R si R-T dureaza 3 secunde fiecare, toate celelalte sarituri avand o durata de 2 secunde fiecare.

Acelasi rationament se aplica si pentru Q. Daca acesta ar fi un copac frunzos, P-Q si Q-S ar dura 2 secunde fiecare. Avand in vedere ca secventa trebuie sa aiba o durata minima, S-R, R-T, T-R si R-P trebuie sa fie toate sarituri de cate 2 secunde fiecare. Cu toate acestea, acest lucru nu este posibil, asa cum veti constata daca incercati sa marcati cu S, R si T copacii din imaginea de mai jos.



Ca urmare, Q nu poate fi un copac frunzos si ca urmare, singura posibilitate ramasa este ca P, S sau T sunt copaci frunzosi.

Relationarea cu gandirea computationala:

Competente - Descompunere in factori primi (DE), Evaluare (EV), Generalizare (GE)

Domeniu – Algoritmi si programare

Aceasta intrebare implica determinarea solutiei optime la o problema. Computerele sunt deseori utilizate pentru determinarea valorii maxime sau minime dintr-o serie de masuratori. In cazul de fata, se poate considera ca acesti copaci reprezinta aplicatii pe un touch screen, iar sariturile maimutei intre copaci reprezinta miscarilor degetelor utilizatorului de la o aplicatie la alta. Un programator al interfetei grafice ar putea fi interesat de modul in care sa aranjeze aplicatiile pe ecran pentru o secventa comuna de comenzi astfel incat acestea sa necesite cat mai putin timp posibil.

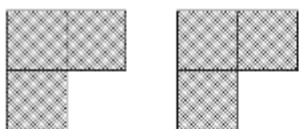
Jocul in L

Kiki si Wiwi se joaca "Jocul in L" pe o tabla de 4x4 casute. Ele plaseaza pe rand piese in forma de L pe tabla astfel incat

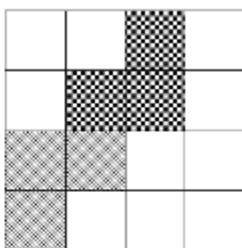
- fiecare piesa plasata de Kiki este orientata asa cum este prezentat mai jos,
- fiecare piesa plasata de Wiwi este orientata asa cum este prezentat mai jos,
- fiecare piesa este plasata in intregime pe tabla si
- piesele nu se suprapun.

Piese, odata plasate pe tabla, nu pot fi deplasate. Un jucator pierde atunci cand ii vine randul si nu este posibil sa plaseze o piesa pe tabla de joc conform regulilor de mai sus. Mai jos este prezentat un exemplu in care Kiki deschide jocul. In acest exemplu, Kiki poate castiga jocul prin plasarea unei piese in coltul din dreapta-jos.

Orientare piese Kiki



Primele 2 mutari



Orientare piese Wiwi



Intrebare:

Pentru cate variante din cele 9 posibile pentru prima mutare ale lui Kiki, este garantata castigarea jocului, indiferent de modul in care piesele sunt plasate pe tabla in urmatoarele runde?

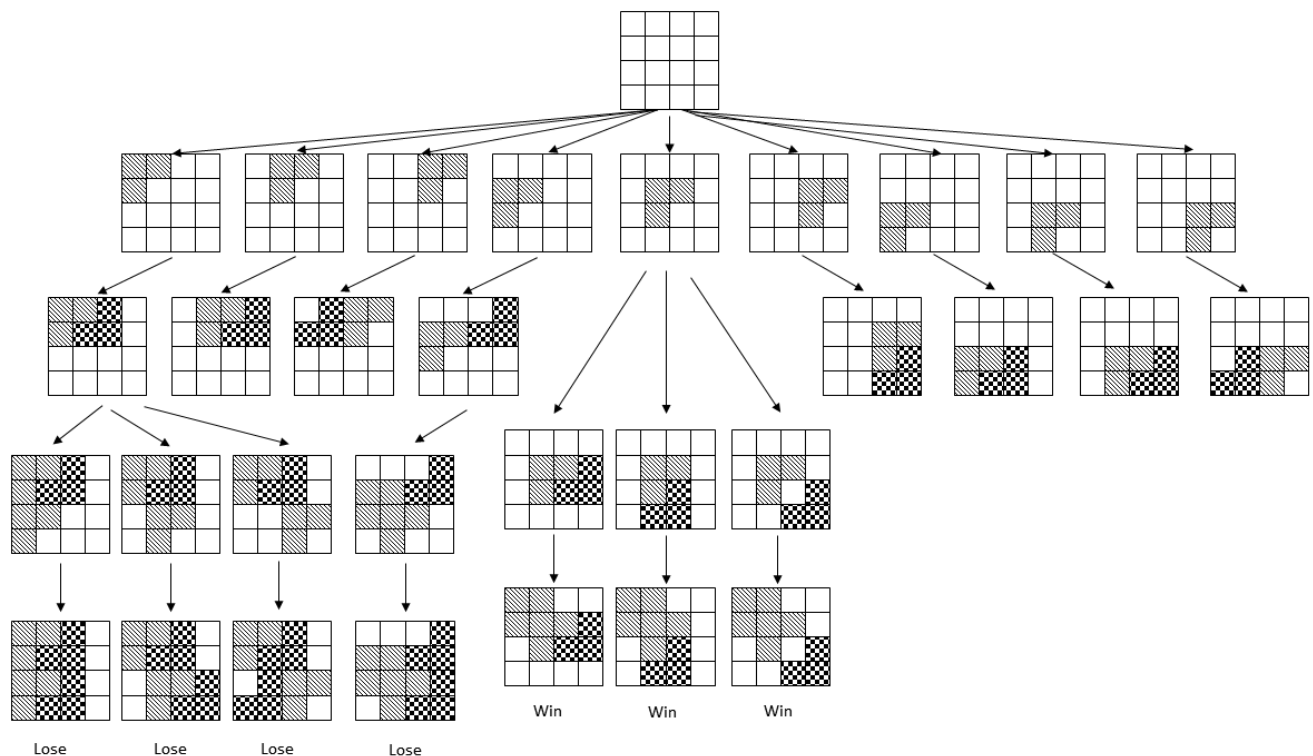
0 1 2 sau 3

Raspuns corect:

1

Explicatie:

Raspunsul corect este 1. Prin plasarea unei piese in pozitia “din mijloc”, Kiki va castiga garantat jocul. Indiferent unde isi plaseaza Wiwi piesa la prima mutare, Kiki poate sa isi plaseze o piesa in coltul din stanga sus. Conform regulilor, acest lucru inseamna ca Wiwi nu mai poate plasa alta piesa. Daca Kiki plaseaza o piesa in orice alta pozitie la prima mutare, atunci exista cel putin o modalitate sa piarda acest joc. Diagrama de mai jos detaliaza cateva posibilitati si simetrii pentru a exclude multe dintre pozitii.



Relationarea cu gandirea computationala

Competente - Abstractizare (AB), Gandire algoritmica (AL)

Domeniu – Date, structuri de date si reprezentari

Toate posibilitatile dintr-un joc pot fi explicate printr-o diagrama asemenatoare celei de mai sus. In acest joc, un nod radacina corespunde starii initiale a tablei de joc. Apoi, pentru fiecare mutare posibila, sagetile indica spre noua stare a tablei de joc. Intreaga structura este construita continuand acest model. Un joc arbore este un tip special de graf orientat.

In functie de tipul problemei, este uneori mai util sa se exploreze nodurile vecine intai inainte de a muta analiza la nodurile de pe nivelul urmator (parcurerea grafurilor in latime), in timp ce uneori este mai util ca explorarea sa se faca in adancimea fiecărei ramuri, cat de mult posibil, inainte de a se face backtracking (parcurerea grafurilor in adancime). Aceste doua strategii de cautare au proprietati si cerinte de memorie diferite.