

6 godina u Srbiji!



R

Dabar 2018/19.

Задаци и решења за ученике виших разреда основне школе

– ШКОЛСКО ТАКМИЧЕЊЕ

(новембар, 2018.)



<http://dabar.edu.rs/>

Sadržaj:

Zadaci:

Šta će Maja obući danas?	6
Ada i bojice.....	8
Redosled oblačenja	10
Roboti	12
Sobe	13
Klara je ljubitelj cveća.....	15
“Dabar babuška” lutkica.....	17
Mutacija vanzemaljca.....	19
Let	21
Gde curi voda?.....	23
Skokovi po tabli.....	25
Porodica krokodila.....	27
Cena spavanja.....	29
Mapa za dešifrovanje	31
Tajne tajne	33
Soundex algoritam.....	35
Tajne poruke.....	37
Sortiranje knjiga	41
Mape sa blagom.....	44
Telegrafaska mreža.....	46
Domino pločice	48
Optički kablovi	50
Stara računarska mašina	52
Magična mašina.....	54



O takmičeњу i priručniku

Draga deca i poštovane kolege,

Hvala Vam na želji, volji i entuzijazmu sa kojim pristupate ovom takmičenju! Ponosni smo na Vas i na činjenicu da se već 6. godinu zaredom družimo. U proteklom periodu na takmičenju je učestvovalo preko 130.000 takmičara, a na ovogodišnjem školskom nivou, preko 40.000. Ponovo smo, zajedno, pomerili granice i uključili veći broj dece nego prethodne takmičarske godine!

Posebno nas raduje činjenica da se sa nama družite od 1. razreda, pa do vašeg punoletstva i završetka srednje škole. DABAR je postalo TAKMIČENJE UZ KOJE ODRASTATE!

Takmičenje Dabar je namenjeno svim učenicima, ne samo talentovanim. Želja nam je, da kroz zabavne zadatke koje ste rešavali na školskom takmičenju 2018/19., ŠTO VIŠE DECE UVIDI DA SE SA INFORMATIČKIM PROBLEMIMA SUSREĆU U SVAKODNEVNOM ŽIVOTU I DA IH JE MOGUĆE SA LAKOĆOM REŠAVATI.

Priručnik je namenjen nastavnicima i učenicima kao pomoć pri bavljenju temama i intelektualnim problemima koji su predstavljeni kroz zadatke. Kako deca vole da se takmiče i vole da razmišljaju, naš posao je da ih i tokom godine podstičemo da razvijaju takmičarski duh i radoznalost.

Priručnik je i deo riznice Dabar intelektualnih problema, koja se iz godine u godinu uvećava. Pripručnik su pripremili organizatori takmičenja, kao nešto na šta smo ponosni ;).

U takmičenje je uloženo mnogo rada i energije, tako da nas posebno raduje što takmičenje postaje sve masovnije i popularnije, ne samo u našoj zemlji već i širom sveta.

Trenutno izdanje priručnika je "privremeno". Nakon narednog nivoa takmičenja, našoj riznica zadataka ćemo dodati nove. No, čak i oni će biti veoma brzo odrađeni od strane onih koji vole takve zadatke. Šta onda?

Pozivamo vas da pratite naše aktivnosti na sajtu dabar.edu.rs ili na sajtu Međunarodnog takmičenja Dabar bebras.org i da se zajedno sa nama radujete novim zadacima.

Uživajte u rešavanju zadataka!

Srdačno Vaš,

Programski odbor takmičenja Dabar

<http://dabar.edu.rs/>



Bodovna tabela

Dabarčić

R.Br.	Zadatak	Težina	Bodovi
1.	Šta će Maja obući danas?	Lak	6
2.	Ada i bojice	Lak	6
3.	Redosled oblačenja	Lak	6
4.	Roboti	Lak	6
5.	Sobe	Srednje težine	9
6.	Klara je ljubitelj cveća	Srednje težine	9
7.	"Dabar babuška"	Srednje težine	9
8.	Mutacija vanzemaljca	Srednje težine	9
9.	Let	Težak	12
10.	Gde curi voda?	Težak	12
11.	Skok po tabli	Težak	12
12.	Porodica krokodila	Težak	12

Mladi dabar

R.Br.	Zadatak	Težina	Bodovi
1.	Sobe	Lak	6
2.	Klara je ljubitelj cveća	Lak	6
3.	Mutacija vanzemaljca	Lak	6
4.	"Dabar babuška"	Lak	6
5.	Let	Srednje težine	9
6.	Gde curi voda?	Srednje težine	9
7.	Skok po tabli	Srednje težine	9
8.	Porodica krokodila	Srednje težine	9
9.	Cena spavanja	Težak	12
10.	Mape sa blagom	Težak	12
11.	Tajne Tajne	Težak	12
12.	Soundex algoritam	Težak	12



<http://dabar.edu.rs/>

Dabar

R.Br.	Zadatak	Težina	Bodovi
1.	Let	Lak	6
2.	Tajne poruke	Lak	6
3.	Skok po tabli	Lak	6
4.	Porodica krokodila	Lak	6
5.	Mape sa blagom	Srednje težine	9
6.	Cena spavanja	Srednje težine	9
7.	Soundex algoritam	Srednje težine	9
8.	Tajne tajne	Srednje težine	9
9.	Okretanje karti	Težak	12
10.	Stara računarska mašina	Težak	12
11.	Sortiranje knjiga	Težak	12
12.	Mapa za dešifrovanje	Težak	12

Stariji dabar

R.Br.	Zadatak	Težina	Bodovi
1.	Mape sa blagom	Lak	6
2.	Cena spavanja	Lak	6
3.	Tajne tajne	Lak	6
4.	Soundex algoritam	Lak	6
5.	Optički kablovi	Srednje težine	9
6.	Mapa za dešifrovanje	Srednje težine	9
7.	Stara računarska mašina	Srednje težine	9
8.	Sortiranje knjiga	Srednje težine	9
9.	Okretanje karti	Težak	12
10.	Telegrafaska mreža	Težak	12
11.	Magična mašina	Težak	12
12.	Domino pločice	Težak	12

Izbor zadataka za takmičenje i prevod, Programski odbor takmičenja:

Milan Rajković (predsednik programskog odbora);
Suzana Miljković (član programskog odbora);
Jelena Hadži-Purić (član programskog odbora);
Bojan Milosavljević (član programskog odbora);
Marija Andonović Radojević (član programskog odbora);
Ivica Bekrić (član programskog odbora);
Saša Jevtić (član programskog odbora);
Jasmina Dobrić (član programskog odbora);
Nemanja Đorđević (član programskog odbora);

Tehnička podrška:

Branislav Dolić



<http://dabar.edu.rs/>



Šta će Maja obući danas?



Svakog jutra Maja bira šta će obući. Za oblačenje koristi sledeća pravila:

- Ako nosi **pantalone**, onda nosi **majicu bez slike** ili sa **zvezdicama**;
- Ako nosi **suknju**, onda nosi **majicu sa slikom dabra**;
- Ako nosi **majicu bez slike** ili sa **zvezdicama**, onda nosi **jaknu sa srcem**;
- Ako nosi **jaknu sa srcem**, onda nosi **kapicu sa sličicom**.

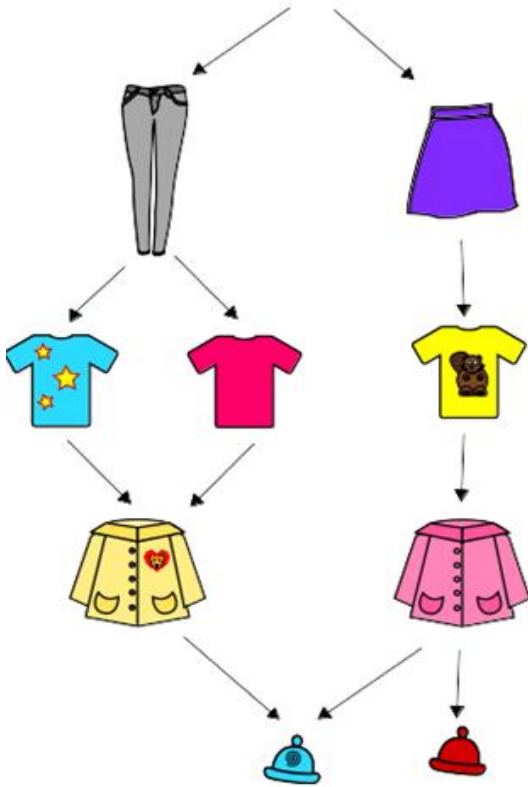
Pitanje / Izazov

Koju od sledećih kombinacija Maja može obući danas?

- A)
- B)
- C)
- D)



Tačan odgovor je: **B**



Kada Maja odluči da li da obuče **pantalone** ili **suknju**, shodno tome ograničava druge izbore. Na slici iznad možete videti koje kombinacije odeće Maja **može obući**.

Informatička pozadina

Kod kompjuterskog programiranja često odlučujemo koje akcije treba izvršiti proverom nekih uslova: "**Ako, eng. IF**" ispunjen je neki uslov (izabrana je suknja), "**Tada, eng. Than**" vrši se određena akcija (izaberite majicu sa dabrom).

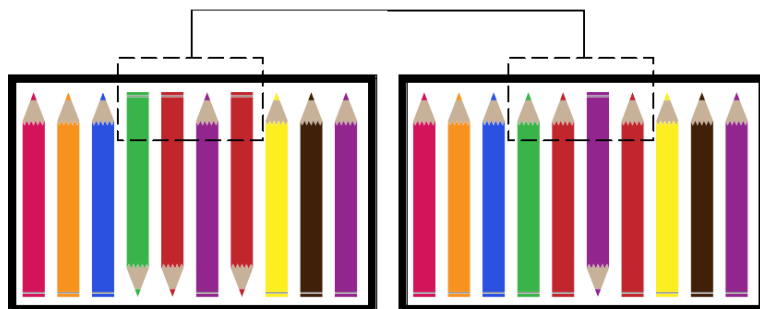
Autori: Lidija Pečar i Špela Cerar (Slovenija)



Ada ima kutiju sa 10 bojica. Neke su okrenute nadole, a neke nagore. Ona želi da ih rasporedi tako da sve budu okrenute na isti način (nagore ili nadole). Smislila je igru da to uradi prema sledećim pravilima:

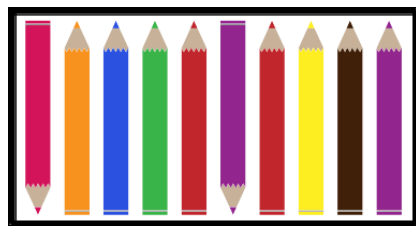
1. Izaberi dve ili više susednih bojica;
2. Bojice koje pokazuju nagore, okreni nadole i obrnuto;
3. Postupak ponavljaj dok sve bojice ne budu okrenute nagore, ili nadole.

Pogledajte sliku:



Pitanje/ Izazov

Koji je najmanji broj ponavljanja postupka potreban da bi se bojice na slici ispod rasporedile tako da sve pokazuju nagore ili nadole?



Odgovori:

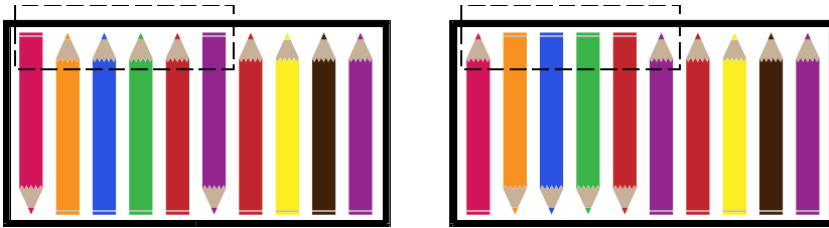
- A) 2
- B) 1
- C) 3
- D) 4



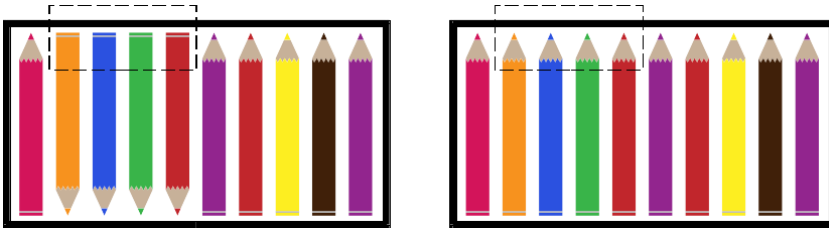
Tačan odgovor je:

Ne možemo rešiti problem bez ponavljanja postupka, jer su 2 bojice okrenute nadole razdvojene, one nisu jedna pored druge. Moguće je sa dva ponavljanja postupka:

Prvi korak: okrenuti prvu, drugu, treću, četvrtu, petu i šestu bojicu.



Drugi korak: Okrenuti drugu, treću, četvrtu i petu bojicu.



Informatička pozadina

Rešavanje problema sa minimalnim brojem koraka jedna je od najvažnijih veština u računarskoj nauci i životu. Dobar programer je uvek malo lenj i želi pronaći najbrže rešenje problema. U gornjem slučaju, najbolje rešenje se može naći samo u 2 koraka, ali u složenijem slučaju možda ga nije lako pronaći.

Autor: Zsuzsa Pluhár (Mađarska)





Redosled oblačenja

Mama dabar pažljivo na stolu reda odeću i obuću za svog sina Bruna. Na slici dole su dati delovi omiljene odeće i obuće. Ti delovi redom su: košulja, potkošulja, pantalone, donji veš, tregeri, čarape, cipele. Bruno je pametan dečak koji zna redosled oblačenja garderobe. Bruno voli svoju mamu i uvek poštuje redosled oblačenja tako što počinje oblačenje od garderobe koju je mama stavila na vrhu gomile na stolu. Dalje, Bruno zna da se čarape moraju obući pre cipela, potkošulja se mora obući pre tregera i pre košulje,...



Košulja	Potkošulja	Pantalone	Donji veš	Tregeri	Čarape	Cipele

Pitanje/ Izazov

Pogledjate slike. Koje slike predstavljaju pravilno poređenu gomilu garderobe na stolu u smislu gore opisanih pravila?

A)	B)	C)	D)



Tačan odgovor je:

D)

Da bi proverili korektnost rešenja, najpre posmatramo vrh gomile odeće na svakoj slici. Zbog toga odgovori (A) i (B) nisu tačni, jer nije korektno da se prvo obuče košulja, a posle nje potkošulja. Potom proverimo redosled oblačenja nakon što Bruno obuče komad garderobe koji je na vrhu. Stoga nije tačan odgovor (C), jer ne mogu se obući tregeri pre košulje.

Informatička pozadina

Komadi garderobe su čvorovi usmerenog acikličkog grafa (**DAG**). Grane **DAG** grafa označavaju relaciju “prethodi oblačenju”. Na primer u tom grafu, postoji grana od čarape do cipela. Algoritam topološkog sortiranja grafa omogućuje da čvorovi grafa dobiju numeričke oznake tako da ako čvor **A** ima manju numeričku oznaku od čvora **B**, to znači da komad garderobe **A** se mora obući pre komada garderobe **B**. Algoritam topološkog sortiranja ima brojne primene u računarstvu, upravljanju projektima, pisanju projektne dokumentacije kada se konkuriše za finansiranje projekta, konfigurisanju *Makefile-a*,...

https://en.wikipedia.org/wiki/Topological_sorting

Autor: Urs Hauser (Švajcarska)



<http://dabar.edu.rs/>



Evo pet izjava koje opisuju tri robota iznad:

1. **Bob** i **Moe** su nasmejani.
2. **Bob**, **Moe** i **Lea** imaju po dve noge.
3. **Moe** ima okruglu glavu i **Lea** ima dve noge.
4. Sva tri robota imaju pet prstiju.
5. **Lea** ili **Bob** su podigli ruke.

Pitanje/Izazov

Koje od ovih pet izjava su tačne?

Ponuđeni odgovori:

- A) 2 i 3
- B) 1 i 3
- C) 1 i 5
- D) Nijedna

Tačan odgovor je:

Tačan odgovor je: C) 1 i 5

Objašnjenje:

Put do rešavanja zadatka:

1. Procenite sve izjave i odlučite da li su istinite ili lažne. Uzmite u obzir značenje reči **I** i **ILI**.
2. Pronađite tačan odgovor.

Informatička pozadina

U računarskoj nauci, **Boolean** tip podataka je tip podataka, koji ima dve vrednosti (obično označene **true** (istina) i **false** (laž)), namenjene za predstavljanje vrednost istine logike i Boolove algebre. Mnoge odluke u programu ili algoritmu zasnivaju se na korišćenju operatora **AND** (**I**) i **OR** (**ILI**). Obratite pažnju da je rečenica "**Bob** i **Moe** su nasmejani" istinita u našem slučaju, jer su i **Moe** i **Bob** nasmejani. Napominjemo da ova izjava na bilo koji način ne zavisi od **Lea**. Mogućnost usredsređivanja na važne informacije o problemu je ključni koncept računarskog razmišljanja: ovaj proces se ponekad naziva apstrakcija, na kojoj se usredsređuju najvažniji detalji, a nebitni detalji se zanemaruju kako bi lakše rešili problem.

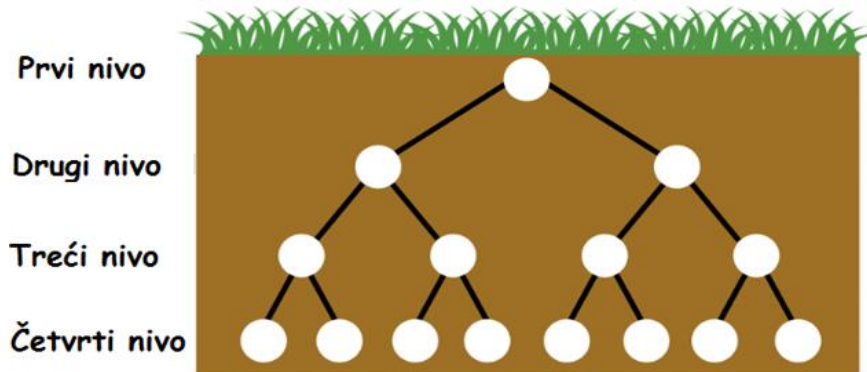
Autor: *Laura Briviba (Letonija)*





Sobe

Porodica od pet dabrova seli se u novi podzemni apartman na četiri nivoa koji je prikazan na slici. Svaki kružić na slici predstavlja jednu prostoriju, a linija predstavlja tunel dug 1 metar. Dabrovi žele odabrati prostorije koje će biti njihove spavaće sobe. Kod odabira moraju voditi računa o tome da ako neku prostoriju pretvore u spavaću sobu sve prostorije na nižim nivoima koje su sa njom povezane postaju nedostupne.



Svi dabrovi imaju svoje dnevne navike i izlaze iz spavaćih soba nekoliko puta tokom dana.

- Otac dabar izlazi 5 puta.
- Majka izlazi 6 puta.
- Branko i Luka izlaze po 2 puta.
- Sara izlazi 3 puta tokom dana.

Pitanje/Izazov

Ako želimo da dabrovi prelaze što kraći put tokom dana na kom nivou treba biti smeštena Sarina soba?

Ponuđeni odgovori

- A) Na prvom nivou
- B) Na drugom nivou
- C) Na trećem nivou
- D) Na četvrtom nivou



Tačan odgovor je:

Tačan odgovor je C

Informatička pozadina!

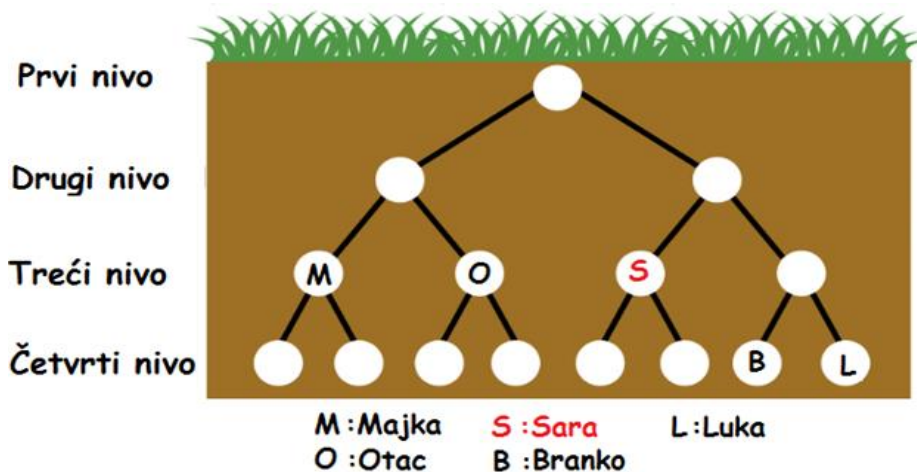
Ako želimo minimizirati ukupan put koji dabrovi prođu tokom dana očito je da dabrovi koji više puta izlaze trebaju biti bliže izlazu (to su majka i otac).

A i **B** odgovori mogu se eliminisati na sledeći način:

Odgovor **A** je netačan jer ako je spavaća soba na prvom nivou, sve ostale sobe ne bi se mogle koristiti (ostali članovi porodice ne bi imali svoje sobe).

Odgovor **B** je netačan jer ako postoji spavaća soba na drugom nivou ona bi morala pripasti majci jer ona putuje najviše. Tada bi druga soba na tom nivou morala ostati prazna jer postoji više od jednog člana porodice koji treba sobu.

Konačno, treba odlučiti između odgovora **C** i **D**. Na trećem nivou treba razmestiti što je moguće više osoba, ali nema za sve njih mesta. Dvoje mora biti smešteno na četvrtom nivou pa to onda treba da budu oni koji izlaze najmanji broj puta tokom dana. To su **Branko** i **Luka**. Prema tome, **majka, otac** i **Sara** su na trećem nivou.



Informatička pozadina

Zadatak je sličan problemu sažimanja teksta u računarskoj nauci. Cilj je umanjiti ukupnu dužinu teksta. Najčešći tekst treba kodirati najkraćim kodom. Huffmanov algoritam za komprimovanje odlično je rešenje za otkrivanje dužine najkraćeg koda (u ovom slučaju najkraće staze). Na početku sortiramo po učestalosti izlazaka naših pet dabrova u padajućem redosledu. Zatim spajamo dva dabra sa najmanjom frekvencijom i zbrajamo ih. Stvaramo novi element sa novom frekvencijom. Ponavljamo taj postupak sve dok ne spojimo sve elemente (u našem slučaju dabrove).





Autor: Yao-Ming Chang (Tajvan)





Klara je ljubitelj cveća

Klara voli raznobojne bukete cveća. Odlazi u cvećaru gde ima sledećih vrsta cveća:

			
Gladiola	Liljan	Lala	Ruža

Svaki cvet je dostupan u bojama:

Bela

Plava

Žuta





Klara želi buket od 6 cvetova tako da:

1. Svaka od boja, bela, plava, žuta, mora se ponoviti tačno dva puta;
2. Cveće iste vrste ne bi trebalo da bude iste boje;
3. Svaka vrsta cveta se sme ponoviti najviše dva puta.



Pitanje/Izazov

Koji od datih buketa zadovoljava sve tri Klarine želje?

A)	B)	C)	D)
			



Tačan odgovor je:

Tačan odgovor je D).

U buketu pod A) nalaze se tri bela cveta. U buketu B) su tri ruže, a u buketu C) dva cveta iste vrste su iste boje.

Informatička pozadina

Mnogi zadaci u računarstvu su opisani skupom ograničenja i zadatak je pronaći rešenje koje zadovoljava sva ograničenja ili što je više moguće. Postoje i složeniji zadaci gde se ograničenja kombinuju sa logičkim operatorima kao na primer konjukcijom (A i B znači da oba ograničenja moraju biti ispunjena, kao na primer tri pravila u zadatku) ili disjunkcija (A ili B znači da je dovoljno da jedno bude ispunjeno).

Autor: *Urs Hauser (Švajcarska)*

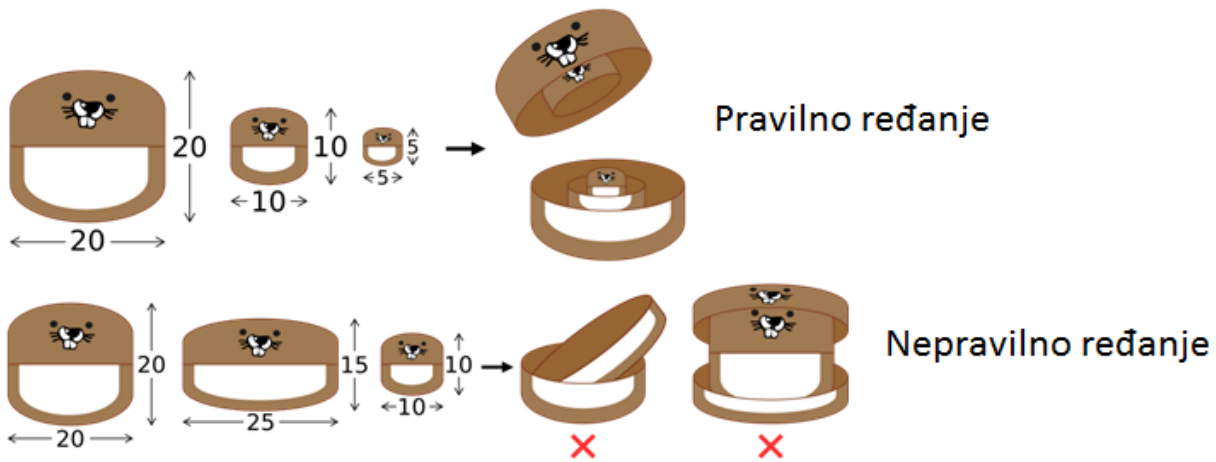


<http://dabar.edu.rs/>

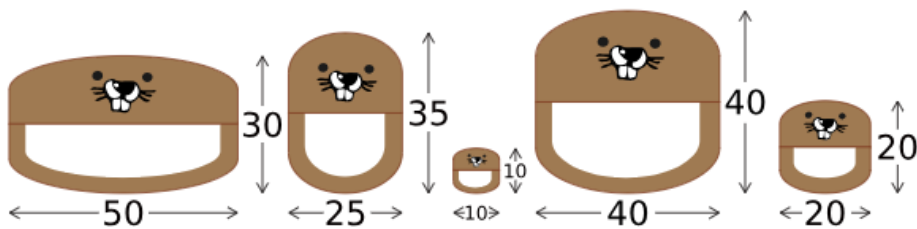


„Dabar babuška” lutkica

„Dabar babuška“ je komplet drvenih lutkica koje se nalaze jedna unutar druge. Svaka lutkica može da se otvori po sredini da bi se videla sledeća lutkica sa manjom širinom i visinom unutar nje.



Slika ispod prikazuje lutkice, kao i njihove dimenzije, koje Jelena ima. Jelena želi da ih složi što je moguće više, jednu unutar druge.



Pitanje/Izazov

Koliko najviše lutkica, jednu unutar druge, može Jelena složiti?

Ponuđeni odgovori:

- A) 4
- B) 5
- C) 3
- D) 2



Tačan odgovor:

A)

Prema pitanju svaka sledeća dabar babuška lutkica u odgovoru mora da ima manju širinu i visinu od prethodne. Ako prvo ređamo lutkice po njihovoj širini u opadajućem redosledu onda možemo biti sigurni da ih njihova širina neće zaustavljati pri sklapanju.



Sada treba uzeti što je moguće više lutkica iz niza poređanih lutkica. Zapamtite da lutkice koje se uzimaju moraju da budu po odgovarajućoj visini. Biranjem najšire lutkice dobija se dabar niz lutkicu sa 3 sloja. Međutim, ako se ne koristi najšira dabar lutkica, dobija se dabar niz lutkicu sa 4 sloja.



Naravno, započnite ređanje lutkica po visini. Onda pokušajte da uzmete što više lutkica po opadajućoj širini.

Informatička pozadina

U informatici se ovaj problem zove najduži opadajući niz. Cilj ovog problema je naći dužinu najdužeg podniza iz datog niza, takve da su svi elementi ili stvari poređani u rastućem/opadajućem nizu.

Ovakvi problemi se mogu pojaviti u matematici i fizici. Informatika pomaže da se ovakvi problemi reše.




Autor: Ya-Chun Hsu (Tajvan)












Vanzemaljca ima glavu, stomak, dve ruke i dve noge. Vanzemaljca može promeniti oblik pomoću sledećih komandi (slika ispod). Moguće je da se delu tela promeni oblik više puta.

Komande za promenu oblika:

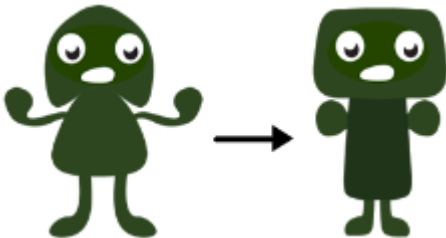
G(K): promeni oblik glave u  , **G(P):** promeni oblik glave u  , **G(T):** promeni oblik glave u 

S(K): promeni oblik stomaka u  , **S(P):** promeni oblik stomaka u  , **S(T):** promeni oblik stomaka u 

R(+): duge ruke  , **R(-):** kratke ruke 

N(+): duge noge  , **N(-):** kratke noge 

Primer promene oblika za komande **G(P)**, **S(P)**, **R(-)**, **N(-)**:

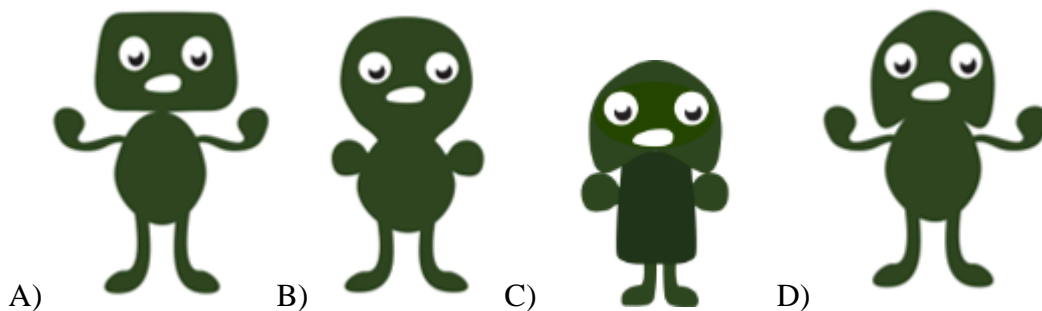


Pitanje/Izazov

Prema sledećim komandama, kako će vanzemaljca izgledati?

G(T), **N(+)**, **S(T)**, **R(+)**, **G(K)**, **R(-)**, **S(K)**

Ponuđeni odgovori:



Tačan odgovor je:

B

Za svaki deo vanzemaljaca, naredna komanda poništava rezultat prethodne.

Prema tome, konačni rezultat je glava u obliku kruga, telo u obliku kruga, kratke ruke i dugačke noge. Dakle, odgovor je **B**.

Informatička pozadina

Dok se izvršava program, instrukcije se izvršavaju u nizu.

Glava, telo, ruke i noge su kao promenljive ili funkcije koje se koriste u programu.

Podešavanje oblika: **K** za krug, **P** za pravougaonik, i **T** za trougao je kao vrednost dodeljena promenljivima ili kao parametar prenet na funkciju.

Autor: Yasemin Gülbahar (Turska)





Let

Grupa dabrova putuje avionom na odmor. Avion ima 9 redova sedišta. Dabrovi se ukrcavaju prema sledećem rasporedu:

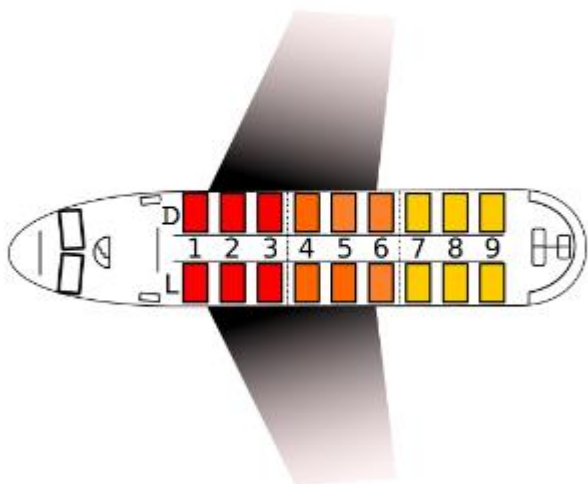
- Prvi ulaze dabrovi čija su sedišta od **7.** do **9.** reda;
- Drugi ulaze dabrovi čija su sedišta od **4.** do **6.** reda;
- Treći ulaze dabrovi čija su sedišta od **1.** do **3.** reda.

Stjuard pomaže dabrovima koji čekaju u redu da bi se ukrcali u avion. On se kreće sa leve strane reda u desnu, a zatim se vraća sa desne strane u levu, pregledajući im karte. Kada, u redu pronade one dabrove, koji ispunjavaju uslove gore navedenog rasporeda ukrcavanja, on im saopštava da uđu u avion. Postupak ponavlja sve dok svi putnici ne uđu u avion.

Pitanje

Odredite redosled ulaska putnika u avion ako oni imaju redom karte kao u tabeli ispod:

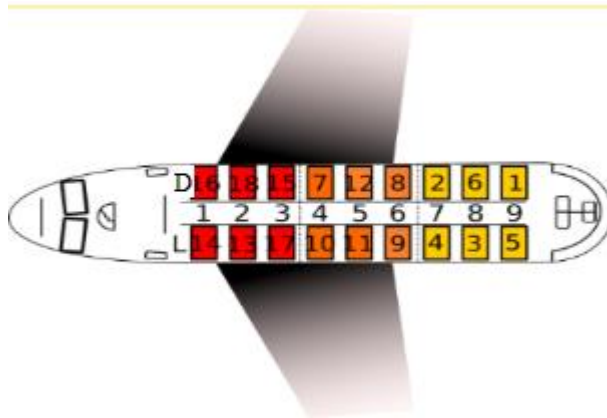
5D	2L	9D	5L	7D	1L	4L	3D	1D	8L	3L	6L	6D	7L	9L	8D	2D	4D
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Ponuđeni odgovori:

- A) 9D, 8L, 7D, 7L, 8D, 7L, 5D, 5L, 6D, 4L, 4D, 6L, 3D, 2L, 1D, 1L, 2D, 3L
B) 9D, 7D, 8L, 7L, 9L, 8D, 5D, 5L, 4L, 6L, 6D, 4D, 2L, 1L, 3D, 1D, 3L, 2D
C) 9D, 9L, 8D, 8L, 7D, 7L, 6D, 6L, 5D, 5L, 4D, 4L, 3D, 3L, 2D, 2L, 1D, 1L
D) 9D, 7D, 8L, 7L, 9L, 8D, 4D, 6D, 6L, 4L, 5L, 5D, 2L, 1L, 3D, 1D, 3L, 2D





Odgovor

Tačan odgovor je pod **D**) 9D, 7D, 8L, 7L, 9L, 8D, 4D, 6D, 6L, 4L, 5L, 5D, 2L, 1L, 3D, 1D, 3L, 2D

Objašnjenje

Treba ići po listi redom s leva udesno i odabirati one dabrove čija su sedišta od **7.** do **9.** reda. To su: **9D, 7D, 8L, 7L, 9L i 8D.** Tada se po istoj listi vraćamo ulevo (ostalo je 12 neraspoređenih dabrova) i raspoređujemo one čija sedišta su od **4.** do **6.** reda. To su: **4D, 6D, 6L, 4L, 5L i 5D.** Konačno u zadnjem prolasku s leva na desno ukrcava se poslednjih šest putnika: **2L, 1L, 3D, 1D, 3L i 2D.**

Informatiča pozadina

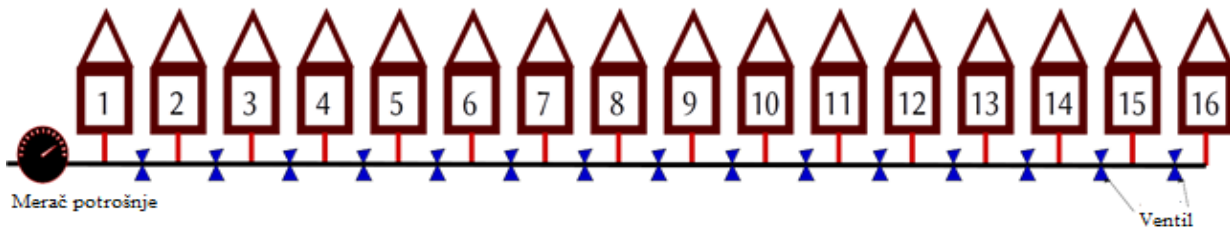
Maksimalna efikasnost postupka ukrcavanja putnika je jedan od bitnih ciljeva avionskih kompanija danas. Za uspešno poslovanje avionskih kompanija potrebno je što više sati koje će avioni provesti u letu odnosno, potrebno je smanjiti vreme tokom kojeg je avion prizemljen. Metode koje se koriste za ukrcavanje putnika veoma su važne pa aviokompanije razvijaju različite strategije s namerom da skrate to vreme.

Autor: Eugenio Bravo (Španija)





Gde curi voda?



Na vodovodnu mrežu povezano je **16 kuća** u istoj ulici. Između jedne od kuća i vodovodne mreže curi voda, ali se ne zna tačno mesto. Vodoinstalateri pokušavaju da nađu to mesto. Da bi pomogli vodoinstalaterima u potrazi, svi vlasnici kuća u ulici su isključili vodu u svojim kućama.

Da bi pronašli gde curi voda, vodoinstalateri će zatvoriti ventil na vodovodnoj mreži, između dve kuće i proveriti da li voda i dalje curi. Ako, na primer, zatvore ventil između kuće broj **8** i kuće broj **9**, a voda posle toga i dalje curi, znaće da je kvar negde između kuća **1** i **8** (tj. da voda ne curi između kuća **9** i **16**).

Pitanje / Izazov

Pretpostavimo da voda curi samo na jednom priključku (obeleženo vertikalnim crvenim linijama na slici). Najmanji broj ventila koji se moraju zatvoriti da bi se utvrdilo gde voda curi je:

Odgovori:

- A) 4
- B) 2
- C) 6
- D) 8



Tačan odgovor je:

A) 4

Ovo je vrlo praktičan primer gde se binarno pretraživanje koristi izvan informatike. Kako kvar može biti bilo gde, najbolje je da se ne oslanjate na sreću da pronađete gde curi, već da podelite područje pretraživanja na dva jednaka dela svaki put. To će pokazati u kojoj polovini područja pretraživanja postoji kvar. Ponavljanjem postupka može se izolovati kvar. Prvo uklonite polovinu područja pretraživanja. Zatim uklonite polovinu polovine levo itd. Istu binarnu tehniku pretraživanja možete koristiti za pronalaženje knjige u knjižari itd.

Informatička pozadina

Ovo je klasični algoritam pretrage: binarno pretraživanje. Ciljana vrednost se upoređuje sa srednjim elementom sortirane liste; ako su nejednake, samo zadržite polovinu dela u kojoj se nalazi ciljana vrednost i odbacite drugu polovinu. Ponovite istu operaciju na preostaloj polovini dok se proces pretrage ne završi. Ako je preostala polovina prazna, cilj ne postoji na listi.

Više detalja o binarnom pretraživanju:

https://en.wikipedia.org/wiki/Binary_search_algorithm

Autor: *Pieter Waker (Južnoafrička republika)*



<http://dabar.edu.rs/>

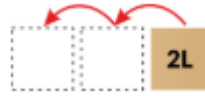


Skokovi po tabli

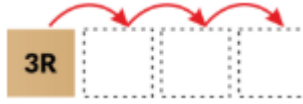
Postoji **8 polja** na tabli koja su obeležena brojevima od **1** do **8**.

Sa polja se može ići u levu stranu, u desnu stranu ili se može stajati u mestu:

Na primer: **2L** znači pomeri se **2 polja u levu stranu**:



3R znači pomeri se **3 polja u desnu stranu**:



"**0**" znači ne pomeraj se sa ovog polja.

Pogledajte sledeću tablu na kojoj su polja obeležena brojevima od **1** do **8**:

1R	3R	2L	0	3R	1R	3L	2L
1	2	3	4	5	6	7	8

Sa kog polja treba krenuti tako da, poštujući pravila, svako polje bude posećeno?

Odgovori:

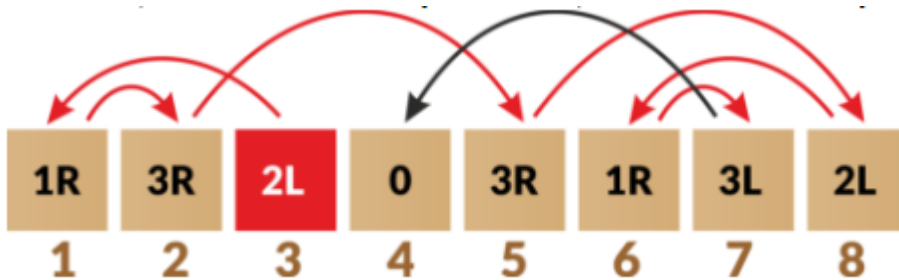
- A) 2
- B) 3
- C) 5
- D) Nije moguće posetiti svako polje.



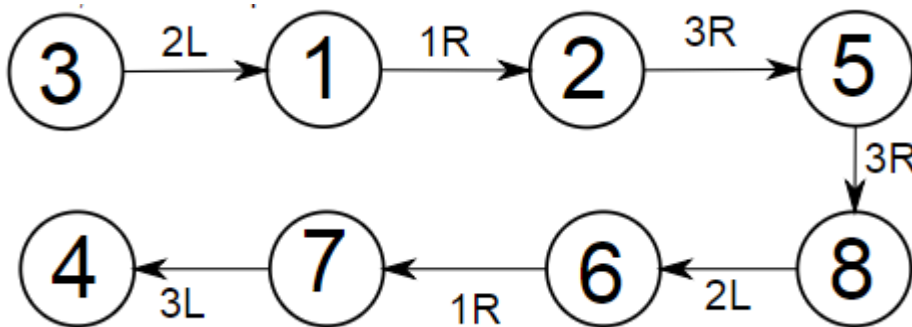
Tačan odgovor je:

B

Gledajući unazad, možemo videti da do polja 4 možemo doći preko polja 7, do kog se dolazi preko polja 6, do kog se dolazi od polja 8, do kog se dolazi od polja 5, do kog se dolazi od polja 2, do kog se dolazi od polja 1, do kog se dolazi od polja 3.



Ovo možemo nacrtati kao usmereni grafik, gde svaki čvor označava kutiju, a veza između njih označava “pomeraj između kutija”. Ovaj grafik može biti nacrtan počevši od bilo od kog čvora, i gotov je kada su sve kutije nacrtane.



Informatička pozadina

Ovaj problem pokriva sledeće pokazivače: mi možemo pomisliti na pomeraj “3D” od kutije 2 do kutije 5 prateći pokazivač. Gledajući ovu grupu pokazivača, koji su usmeren grafik, mi gledamo “roditelja” ili “glavu” čvora ove kolekcije. Pratiti niz pokazivača je važno u upravljanju memorijom od strane operativnog sistema (ili *Java garbage collection*) tako da se memorija koja se više ne koristi može reciklirati i “regenerisati” za druge programe koji će se koristiti. Mnoge greške u softveru mogu se pronaći tako što ćemo pratiti problematične kalkulacije/instrukcije nazad do njihovog “roditelja”. Naredba “go to” (ili “jump” naredba na asemblerskom jeziku) može se modelovati pomoću ove igre. Naredba “go to” pokazuje da umesto izvršavanja “sledeće” instrukcije, možete preći na drugi deo programa i nastaviti izvršavanje. U ovom zadatku, “program” je sekvenca “go to” naredba, gde je jedina “završna” naredba ona na poziciji 4.

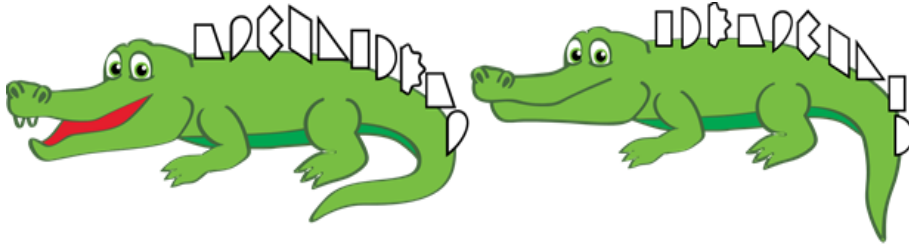
Autor: Troy Vasiga (Kanada)





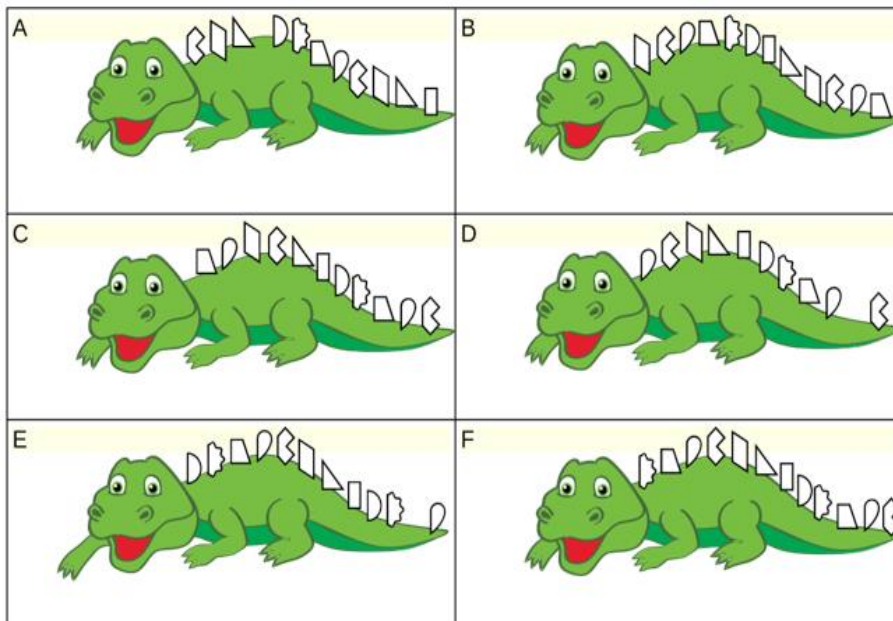
Porodica krokodila

Cheril i **Charlie** krokodili pripadaju istoj porodici krokodila. Ova porodica krokodila ima vrhove na svojim leđima koji su poređani po određenom redosledu. Ponekad, ovi krokodili izgube vrhove, ali ti vrhovi im izrastu ponovo.



Pitanje/Izazov

Koji krokodili su iz iste porodice kao **Cheril** i **Charlie**?



Ponudeni odgovor:

- A) A, E, F
- B) A, B, F
- C) A, C, F
- D) B, E, F



Tačan odgovor je:

A, E i F

Jedini pravi odgovor može biti **A**, **E** i **F**, pošto prate šablon (kao što vidite, početna tačka nije uvek ista, vrh može nedostajati, ali prati isti redosled šablona).

Informatička pozadina

Algoritmi i programiranje:

Algoritam je niz instrukcija ili skup pravila za postizanje zadatka.

Podaci, strukture podataka i predstavljanja:

Podaci mogu imati mnoge forme, na primer, slike, tekst ili brojeve. Kada pogledamo podatke u ovom pitanju, tražimo niz slika koje će pomoći u rešavanju problema. Identifikacijom ovih slika možemo napraviti predviđanja, stvoriti pravila i riješiti sve opšte probleme.

Autor: *Allira Storey (Australija)*





Cena spavanja

Ivica radi na centralnoj železničkoj stanici do koje stiže vozom koji kreće sa stanice u blizini njegove kuće. Radno vreme počinje u **8:00 časova** i za svakih **15 minuta** kašnjenja, mora da plati novčana kaznu od **10 dolara**. Na primer, ako Ivica stigne na posao pre **8:15 časova (npr. 8:11 časova)** tada ne plaća kaznu, ali ako stigne u **8:20 časova**, kaznu plaća za prvih **15 minuta** kašnjenja.

Ivica se uspavao ovog jutra i stigao na stanicu u blizini svoje kuće u **8:08 časova**.



Sledeća tabela prikazuje red vožnje različitih linija vozova koje on može da koristi od stanice u blizini svoje kuće do svog posla:

Linija voza:	Red vožnje	Vreme potrebno do centralne železničke stanice	Cena karte
Regular	Od 6,00 časova ujutru Na svakih 5 minuta	40 minuta	\$5
Shuttle	Od 6,00 časova ujutru Na svakih 10 minuta	30 minuta	\$10
Light	Od 7,00 časova ujutru Na svakih 15 minuta	20 minuta	\$15
Express	Od 7,00 časova ujutru Na svakih 20 minuta	12 minuta	\$20

Pitanje/Izazov

Koja linija voza je najisplativija za Ivicu (Uzmite u obzir cenu karte i kaznu koju mora da plati zbog kašnjenja na posao)?

Odgovori:

- A) Regular
- B) Shuttle
- C) Light
- D) Express



Tačan odgovor je:

B) Shuttle linija

Shuttle linija je najisplativija linija za Ivicu zato što:

- Ako bi koristio **Regular** liniju, on bi stigao na posao u 08:50, tako da bi ga to koštalo 35 dolara, što je 5 dolara za kartu plus 30 dolara ($3 * 10$ dolara).
- Ako bi koristio liniju **Shuttle**, on bi stigao na posao u 08:40, tako da bi ga to koštalo 30 dolara, što je 10 dolara za kartu plus 20 dolara ($2 * 10$ dolara).
- Ako bi koristio liniju **Light**, on bi stigao na posao u 08:35, tako da bi ga to koštalo 35 dolara, što je 15 dolara za kartu plus 20 dolara ($2 * 10$ dolara).
- Ako bi koristio liniju **Ekpress**, on bi stigao na posao u 08:32, tako da bi ga to koštalo 40 dolara, što je 20 dolara za kartu plus 20 dolara ($2 * 10$ dolara).

Informatička pozadina

Ovaj zadatak uvodi koncept Optimizacije. Optimizacija na svoj najjednostavniji način znači odabir najboljeg elementa (ovde je najefikasnija opcija) iz nekih dostupnih alternativa (ovde razne linije vozova) u odnosu na neki kriterijum (ovde je vreme i trošak).

Problem optimizacije obično se sastoji od maksimiziranja ili minimiziranja vrednosti funkcije sistemskim odabirom ulaznih vrednosti i izračunavanjem vrednosti funkcije.

Autor: Hamed Mohebbi (Iran)





Mapa za dešifrovanje

Dabar se prijavio da igra igru Mape. Tokom ove igre, svi putevi su postali jednosmerni, ali igračima nije data mapa. Umesto toga, data je tabela ispod.

	A	B	C	D	E	F
A		↻				
B			↻			
C	↻			↻	↻	
D						
E						↻
F		↻			↻	

Postoji 6 gradova: **A**, **B**, **C**, **D**, **E** i **F** a veza između njih prikazana je u tabeli iznad. Strelica na mapi pokazuje jednosmeran put od grada koji odgovara u redovima a završava se u gradu koji odgovara kolonama. Kada ne postoji strelica, to znači da nijedan put direktno ne povezuje ta dva grada. Na primer, postoji jednosmeran put od grada **B** do grada **C**, ali ne postoji jednosmeran put od grada **B** do grada **D**.

Pitanje/Izazov

Koje od sledećih tvrdnji su tačne?

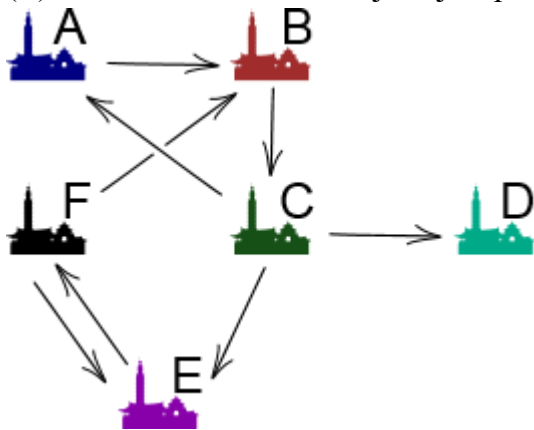
Odgovori:

- A) Dabar mora da koristi najmanje 3 puta u jednom pravcu da bi došao iz grada **C** u grad **B**.
- B) Dabar mora da koristi najmanje 3 puta u jednom pravcu da bi došao iz grada **A** u grad **D**.
- C) Dabar mora da koristi najmanje 3 puta u jednom pravcu da bi došao iz grada **A** u grad **C**.
- D) Dabar mora da koristi najmanje 3 puta u jednom pravcu da bi došao iz grada **B** u grad **E**.



Tačan odgovor je:

(B) Dabar mora da koristi najmanje 3 puta u jednom pravcu da bi došao iz grada **A** u grad **D**.



Možemo napraviti mapu koja prikazuje putanje između gradova u jednom pravcu kao što je prikazano gore. Sada razmotrite svaki od datih odgovora:

(A) Dabar može ići od grada **C** do grada **A** do grada **B** koristeći samo dva puta. Dakle ova izjava je netačna.

(B) Iz grada **A**, dabar može da ode samo u grad **B**, a potom samo u grad **C**. Ovo znači da dabar mora da koristi bar još jedan put za ukupno najmanje tri puta. Dakle ova izjava je tačna.

(C) Dabar može od grada **A** do grada **B** do grada **C** koristeći samo dva puta. Dakle ova izjava je netačna.

(D) Dabar može od grada **B** do grada **C** do grada **E** koristeći ukupno samo dva puta. Dakle ova izjava je netačna.

Informatička pozadina

Mapa prikazana u rešenju je usmeren grafik (poznat kao i digraph). Usmereni grafik sastoji se od vertikala (gradova u ovom zadatku) i usmerenih ivica (jednosmerni putevi u ovom zadatku) koji pokazuju kako su vertikale povezane.

Postoji mnogo drugih načina za predstavljanje usmerenog grafika. Na primer možemo koristiti matricu susednosti slično tabelama u ovom zadatku.

Predstavljanje podataka na više načina je vrlo često u računarstvu. Važno je razumeti prednosti i mane svakog izbora. Na primer, slike kao u rešenju su vizuele i lepe za čitanje, ali računari mogu lakše da manipulišu matematičkim tabelama.

Autor: Nina Chang (Tajvan)

<http://dabar.edu.rs/>





Xavier, Ylenia i Zoe igraju igru koja ima samo jednog pobjednika. Niko od njih ne priča drugima da li je pobjedio ili nije. Igra se igra na sledeći način:

1. **Xavier i Ylenia** tajno bacaju novčić.
 2. **Xavier i Zoe** tajno bacaju novčić.
 3. **Ylenia i Zoe** tajno bacaju novčić.
 4. Svaka osoba će reći da li su dva novčića koje su bacali pali **“isto”** ili **“različito”**.
- Oni koji **NISU** pobjedili u igri će reći istinu.
 - Ali, oni koji su pobjedili u igri će lagati u svojoj izjavi.

Na primer, ako su novčići kao na slici i **Zoe** je pobjedio u igri, svaka osoba izjavljuje **“različito”**.



Ako **Xavier** kaže **“isto”**, **Ylenia** kaže **“isto”** i **Zoe** kaže **“različito”**, koja od sledećih izjava je tačna?

Ponudeni odgovori:

- A. Možemo biti sigurno da niko nije pobjedio u igri.
- B. Sigurni smo da je neko pobjedio u igri ali ne znamo ko je pobjedio.
- C. Sigurni smo da je neko pobjedio u igri i znamo ko je pobjedio.
- D. Ne znamo da li je neko pobjedio u igri.

Tačan odgovor je:

B

Postoji nekoliko načina na koje možemo doći do ispravnog odgovora. Navešćemo neke od njih detaljnije.

Pobeda Vas navodi da lažete?

Zapažamo da, ako osoba laže to znači da je pobjedila u igri. Pobeda u igri čini nekoga lažovom. Od ovog trenutka nadalje, kada govorimo o osobi koja laže, bilo bi poželjno da mislimo "ova osoba je pobjedila u igri".

U stvarnom životu, ljudi se verovatno ne pretvaraju u lažove kada pobjede u igri. Ali u svrhu ovog problema, pretpostavimo da smo u svetu gde je to istina.

Imamo lažova!

Kao drugo, utvrdimo da je neko lagao u „istom, istom, drugom“ scenariju. Nudimo dva načina da dođemo do ovog zaključka.

Prvi način: Pronaći kontradikciju

Oni su rekli „isti, isti, drugačiji“. Pretpostavimo da su sve ove tvrdnje bile istinite. S obzirom da je Xavier rekao „isto“ tada bi XY bacanje i XZ bacanje bilo isto. S obzirom da je Ylenia rekla „isto“ tada XY bacanje i YZ bacanje trebaju da budu ista.



Ovo

<http://dabar.edu.rs/>

je u suprotnosti sa izjavom „svi su govorili istinu“. Stoga zaključujemo da nisu svi govorili istinu. Mora postojati pobjednik u igri. Oprez: Uzmite u obzir da gore navedeni argument ne znači nužno da je Zoe pobedio u igri. To samo znači da su tvrdnje „sve tvrdnje su bile istinite“ pogrešno.

Drugi način: Napraviti tabelu

Možemo da nacrtamo tabelu svih 8 mogućnosti za sva 3 bacanja. U prve 3 kolone pišemo rezultate svakog slučaja. U poslednje 3 kolone pišemo objavu svakog karaktera ako je ona istinita.

BACANJA			ISTINITE IZJAVE		
XY	XZ	YZ	X	Y	Z
Pismo	Pismo	Pismo	Isto	Isto	Isto
Pismo	Pismo	Glava	Isto	Različito	Različito
Pismo	Glava	Pismo	Različito	Isto	Različito
Pismo	Glava	Glava	Različito	Različito	Isto
Glava	Pismo	Pismo	Različito	Različito	Isto
Glava	Pismo	Glava	Različito	Isto	Različito
Glava	Glava	Pismo	Isto	Različito	Različito
Glava	Glava	Glava	Isto	Isto	Isto

Vidimo da nema reda označenog "isto, isto, različito". To znači da je neka morao lagati. Pošto smo sigurni da je neko lagao, možemo eliminisati izbore A i D.

Da li znamo ko?

Znamo da je najviše jedna osoba lagala.

- Ako je Xavier lagao, izjava bi tada trebalo biti: Različito, isto, Različito
- Ako je Ylenia lagao, tada bi izjava trebalo biti: isto, Različito, Različito
- Ako je Zoe lagao, tada bi izjava trebalo biti: isto, isto, isto.

U ovom trenutku nemamo više informacija za dalje razlike tri preostala slučaja. Tako smo sigurni da je neko lagao. Tako je neko pobedio u igri, ali ne znamo ko je.

Informatička pozadina

Ovaj protokol korišćen od strane Xavier, Ylenia, and Zoe poznatiji je kao *Dining Cryptographers protocol*. To je zanimljiv princip komunikacije u kojoj je nemoguće pratiti ni pošiljaoca niti primaoca. Zapravo, anonimne komunikacione mreže zasnovane na ovom problemu često se nazivaju DC mrežama (DC označava "*dining cryptographers*"). Anonimnost je važna, iako donekle kontraverzna, problem današnjeg interneta je što se često prikupljaju lični podaci bez pristanka.

Autor: Mattia Monga (Italija)



<http://dabar.edu.rs/>



Soundex algoritam

Bob želi da zna kako zvuče različite reči. U želji da generiše četvoromestni kod za svaku reč ponavlja sledeće korake:

1. Zadržava prvo slovo reči.
2. Izostavi sva pojavljivanja slova: 'A', 'E', 'I', 'O', 'U', 'H', 'W', 'Y'.
3. Zameni slova brojevima kao što je dato u tabeli:

Slova	Brojevi
B, F, P, V	1
C, G, J, K, Q, S, X, Z	2
D, T	3
L	4
M, N	5
R	6

4. Zameni broj koji se ponovi dva ili više puta zaredom, sa samo jednim jedinim istim tim brojem.
5. Koristi samo prve četiri cifre rezultata, dodajući nule na kraj po potrebi.

Na primer:

Reč	Kod
BOB	B100
BEAVER	B160
HEILBRONN	H416
ESSAY	E200

Pitanje/Izazov

Koji kod će generisati reč “**HILBERT**” ?

Ponuđeni odgovori:

- A) H410
- B) B540
- C) H041
- D) H416



Tačan odgovor je:

Tačan odgovor je **D**.

1. Zadrži prvo slovo reči (**H**).
2. Hilbert → **H4163** (posle primene prva tri pravila).
3. **H416** (zadrži samo prve četiri pozicije).

Informatička pozadina

Ovaj algoritam je poznat kao Soundex.

Soundex je fonetski algoritam za indeksiranje imena zvukom, po engleskom izgovoru. Takvi algoritmi se koriste za pretrage. Ova tehnika se odnosi na fonetsku korekciju: pravopisne greške koje nastaju zbog toga što korisnik postavi upit (niz reči) koji zvuči kao ciljna reč. Algoritam je tačan za zadatke "visokog odziva" (npr. Interpol), iako nije objektivan pri imenima nekih nacionalnih odrednica.

<https://en.wikipedia.org/wiki/Soundex>

Algoritam je opisao Donald Knuth u knjizi "*The Art Of Computer Programming, vol. 3: Sorting And Searching*".

Autor: Ionuț Gorgos (Pakistan)

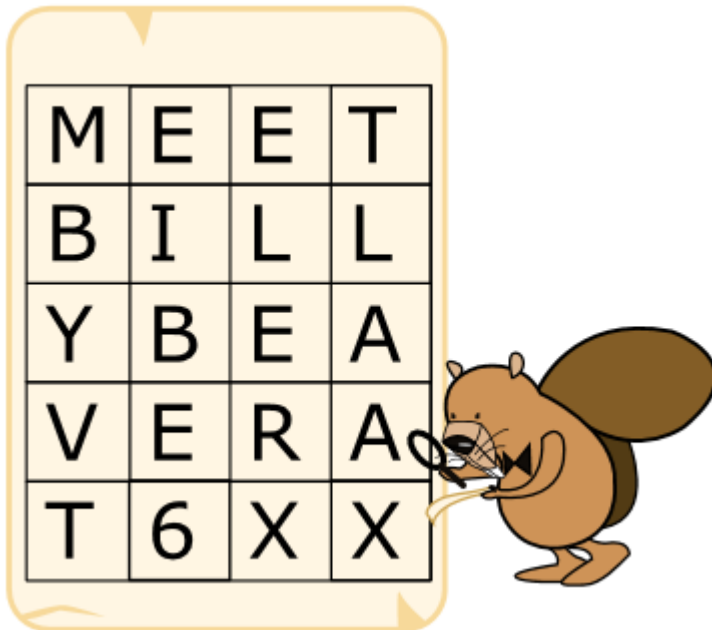




Agenti Saša i Branko sporazumevaju se tajnim porukama. Saša je Branku odlučio poslati sledeću poruku:

MEETBILLYBEAVERAT6

Svaki znak poruke upisao je u tablicu sa četiri kolone, s leva u desno, red po red, počevši od reda na vrhu tablice. U polja tablice koja su ostala prazna upisao je znak X. Rezultat je prikazan na slici.



Tajnu poruku stvorio je prepisujući znakove od vrha do dna tablice, kolonu po kolonu, počevši s leva.

MBYVTEIBE6ELERXTLAAX

Branko je odgovorio Saši koristeći istu metodu. Poslao mu je tajnu poruku sledećeg sadržaja:

OIERKLTEILH!WBEX

Pitanje/Izazov

Kako glasi izvorna poruka koju je poslao Branko?

- A) OKWHERE TOMEET!
- B) OKIWILLBETHERE!
- C) WILLYOUBETHERETOO?
- D) OKIWILLMEETHIM!



Tačan odgovor je:

B) OKIWILLBETHERE!

Objašnjenje

Izvornu poruku otkrićemo tako što u tablicu unesemo tajnu poruku, upisujući znakove od vrha do dna, kolonu po kolonu. Pri tome treba uočiti da poruka sadrži 16 znakova pa je očigledno da u svakoj koloni trebamo upisati po četiri znaka. To izgleda ovako:

O	K	I	W
I	L	L	B
E	T	H	E
R	E	!	X

Čitajući s leva na desno, red po red, otkrićemo ovu poruku:

OKIWILLBETHERE!

Informatička pozadina

Sadržaj poruka koje šaljemo putem računarskih mreža želimo zaštititi od neželjenog čitanja, posebno one koje sadrže lozinke ili lične podatke. Takve poruke šifriramo, što znači da ih pretvaramo u tajne poruke. Da bi to funkcionisalo, primalac mora biti u stanju dešifrovati tajnu poruku i otkriti izvornu poruku. U svakom slučaju, to ne bi smelo biti moguće osobi kojoj poruka nije namijenjena i koja bi mogla zloupotrebiti podatke.

Postoji mnogo vrsta šifriranja. Šifra koju smo koristili u ovom zadatku naziva se šifra premetanja. Kod nje dolazi do zamene redova u kolone i kolona u redove kada se poruka upisuje u tablicu.

Kriptografija se bavi proučavanjem raznih načina šifriranja. Moderna kriptografija predstavlja široko područje istraživanja i uključuje vrlo složene postupke šifriranja, temeljene na zahtevnim matematičkim problemima.

Autor: Sue Sentence (Ujedinjeno kraljevstvo)

<http://dabar.edu.rs/>





Okretanje karte

















Igramo se okretanja karti. Pred Vama su karte poređane u jednom redu i okrenute prema Vama licem

naličjem  .

Definišimo jedan potez u igri na sledeći način:

- Pregledamo karte zdesna nalevo.
- Ako je tekuća karta okrenuta naličjem, onda je okrenite na lice i time je tekući potez okončan.
- Ako je tekuća karta okrenuta licem, onda je okrenete na naličje (licem na dole) i nastavite do sledeće karte. Kada više ne budete imali karte, možete se zaustaviti.

Slika dole pokazuje efekat ovog koraka: najpre okrenete poslednju kartu sa desne strane koja je okrenuta licem i nastavite, potom okrenite kartu (okrenutu licem) koje je levo od te karte. Zatim okrenete kartu koja je levo od ove karte. U ovom trenutku igre, morate se zaustaviti, jer ta treća karta završava okrenuta nagore.

Pre							
Posle							

Pitanje/Izazov

Ako počinjete igru sa 7 karata okrenutih naličjem (nadole):



koliko koraka je potrebno pre nego što se prvi put dobije šema prikazana na sledećoj slici (svih 7 karata su okrenute licem)?



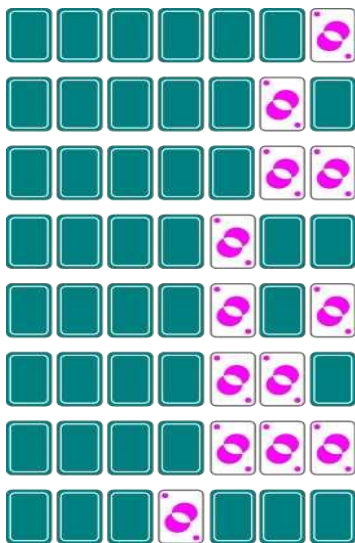
- Potrebno je 10 koraka ili manje.
- Potrebno je više od 10 koraka, ali ne više od 100.
- Potrebno je više od 100 koraka, ali ne više od 1000.
- Nije potrebno više od 1000 koraka.
- Nikada se neće desiti da se igra završi tako svih 7 karata bude okrenuto licem.



Tačan odgovor je:

C)

Postoji više načina da se dođe do tačnog odgovora. Ako posmatrate stanje špila nakon nekoliko početnih koraka igre, dobićete sledeću šemu.



Uočite da je poslednja karta sa desne strane okrenuta u prvom potezu, druga karta gledano s desnog kraja je najpre okrenuta u drugom potezu, treća karta sdesna je napre okrenuta u 4. koraku i 4. karta sdesna je najpre okrenuta u 8. koraku.

Da li postoji nešto značajno u vezi niza brojeva 1, 2, 4, 8, ...? Zapravo, svaki broj ove sekvence je stepen broja dva. Dakle, sledeći brojevi u sekvenci su 16, 32, ... Konkretno, potrebno je 16 koraka pre nego se 5. karta sdesna okrene, 32 koraka za 6. kartu i 64 koraka za 7. kartu. (Uočimo da 7. karta sdesna je prva karta sleva.) Tako da nam je potrebno barem 64 koraka.

Potrebno je uočiti da zapravo nam je potrebno 127 koraka da se sve karte okrenu licem nagore. Uočite da nakon 7. koraka (samo 1 korak pre nego što će biti okrenuta 4. karta) su okrenute prethodne tri karte licem nagore. Slično, nakon 15. koraka (samo 1 korak pre nego što će biti okrenuta 5. karta) su okrenute prethodne četiri susedne karte licem nagore. Tako da nakon 127. koraka (samo 1 korak pre nego što će biti okrenuta npr. 8. karta) je okrenuto svih 7 karti licem nagore tj. nakon koraka $2 \times 64 - 1 = 127$.

Informatička pozadina

Brojevi u računarskom sistemu se predstavljaju u binarnom zapisu koristeći samo cifre 0 i 1 (tj. bitove). U binarnom brojevnom sistemu dekadni broj 1 se predstavlja kao 0000001, dekadni broj 2 se predstavlja kao 0000010, dekadni broj 3 kao 0000011, dekadni broj 4 kao 0000100, dekadni broj 5 kao 0000101, dekadni broj 6 kao 0000110, itd. (Ovde smo koristili 7 bitova za zapis, ali današnji računari koriste 32-bitne ili 64-bitne registre za čuvanje brojeva.)

Da li lica i naličja jedne karte u našem primeru mogu da se predstave bitovima? Zaista, možemo binarnom cifrom 0 predstaviti kartu koja je okrenuta naličjem, dok binarnom cifrom 1 možemo predstaviti kartu koja je okrenuta licem nagore, Dakle sedam karti okrenutih licem nagore se mogu predstaviti binarnim brojem 1111111 čija dekadna vrednost je 127. Stoga je jasno da je zaista potrebno 127 koraka da se postigne tražena šema u kojoj je svih 7 karata okrenuto licem nagore.

Autor: Kris Coolsaet (Belgija)



<http://dabar.edu.rs/>



Sortiranje knjiga

Tri dabra imaju svoje stolove. Na svakom stolu se nalaze knjige. Kao što možete videti na slici, redosled knjiga je pomešan. Dabrovi žele da ga poprave, koristeći dva tipa razmene knjiga:

1. Svaki dabar može zameniti redosled knjiga na svom stolu (primer **(A)**)
2. Dabrovi mogu razmeniti susedne knjige između dva stola koji se nalaze jedan pored drugog

(primer: **(B)**)

Napomena: U svakom krugu razmene, mogu koristiti samo jedan tip razmene.



Trenutno stanje na stolovima je kao na slici iznad (bez primera zamena **(A)** i **(B)**).

Pitanje/Izazov

Na početku, prva raspodela je menjanje redosled knjiga na svojim stolovima. Koji je najmanji broj krugova razmena potreban da bi se knjige rasporedile u redosledu: 1,2,3,4,5,6?

Ponudeni odgovori:

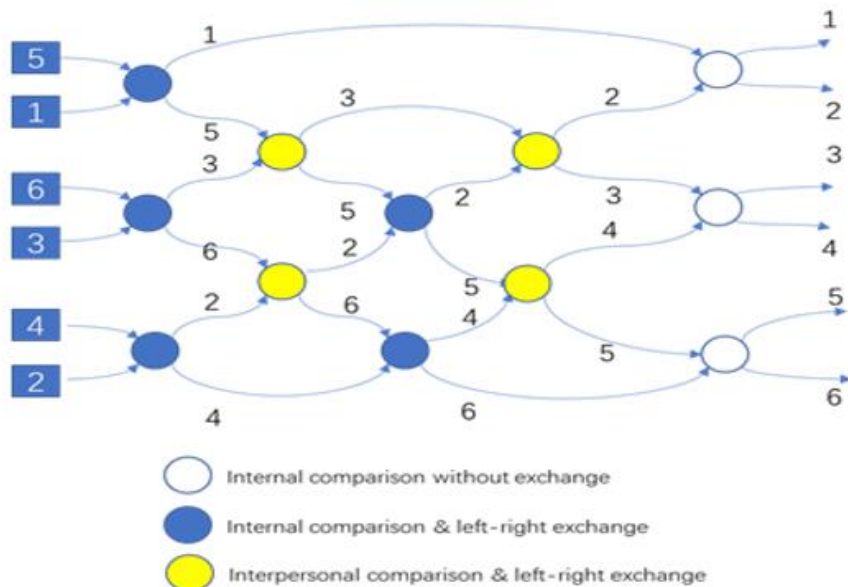
- A) Tri
- B) Četiri
- C) Pet
- D) Šest



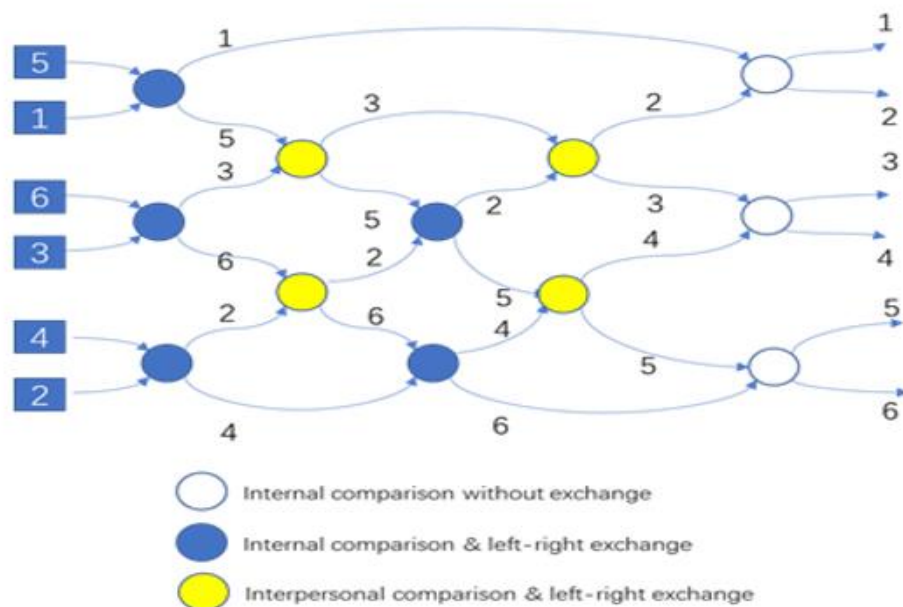
Tačan odgovor je:

B.

Ovaj problem je verzija paralelnog sortiranja mreža. Oni počinju u prvom krugu razmenom knjiga na svakom stolu kako je određeno zadatkom od tabela kako je navedeno u zadatoj izjavi. Poželjna strategija je da se odluči da li da se izvrši razmena ili ne. U drugom krugu, jedina



moguća stvar je da svi dabrovi pokušavaju da razmene knjige između stolova. A onda u sledećem krugu, jedino što je moguće je razmeniti knjige na svakom od stolova i tako dalje. Poželjna strategija je intuitivna i zapravo je optimalna. Korišćenjem ove strategije, lako se može doći do odgovora. Sledeća slika prikazuje navedenu strategiju:



Informatička pozadina

Mreža za sortiranje se sastoji od dve stavke: komparatora i žica. Žice prenose talase sa leva na desno



(jedan talas po žici), koji se kreću mrežom istom brzinom. Svaki komparator povezuje dve žice. Kada par talasa, koji putuju kroz par žica, nailaze na komparator, komparator zamenjuje vrednosti ako i samo ako je vrednost talasa gornje žice veća od vrednosti talasa donje žice. U formuli, ako gornja žica nosi x , a donja žica nosi y , onda nakon komparatora žice nose $x' = \min(x, y)$ i $y' = \max(x, y)$, tako da par vrednosti je sortiran. Mreža žica i komparatora koji će ispravno sortirati sve moguće ulaze u rastućem redosledu naziva se mreža za sortiranje.

Potpuni rad jednostavne mreže za sortiranje je prikazan u nastavku. Lako je videti zašto ova sortirana mreža ispravno sortira ulaze; imajte na umu da će prva četiri komparatora "potonuti" najveću vrednost na dno i "podići" najmanju vrednost na vrh. Konačni komparator jednostavno sortira srednje dve žice.

https://en.wikipedia.org/wiki/Sorting_network

Autor: *Kessarapan Charoensueksa (Tajland)*



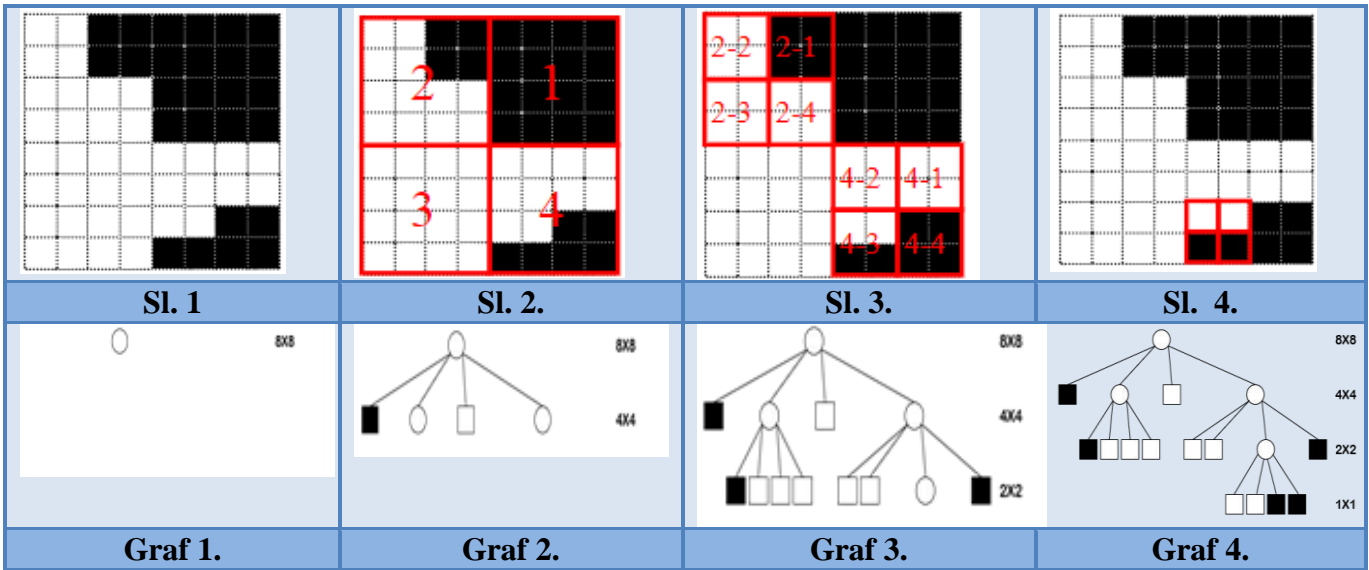


Mape sa blagom

Dabar gusar ima jako veliku mapu sa blagom koja je isečena na manje delove. Svaki delić mape prikazuje regiju širine 8 mernih jedinica i dužine 8 mernih jedinica (prikazano na slici 1). Međutim, dabar gusar ima jako mali brod i ne može da ponese sve delove mape. Pošto je pametan i snalažljiv, uspeo je da pretvori svaku regiju (jedan delić mape) u mali graf (dijagram) u svojoj svesku.

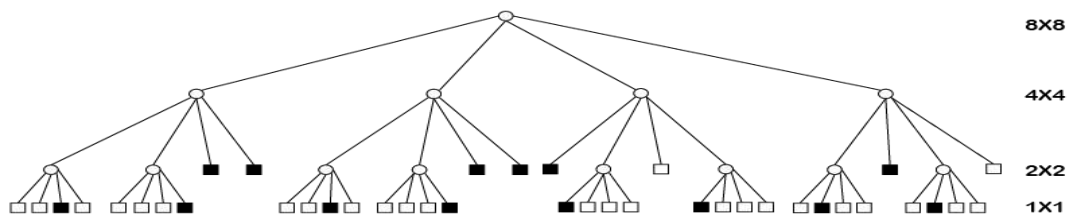
Evo kako:

1. Ako su sve merne jedinice iste boje, on nacrti kvadratić te boje u svojoj svesci.
2. Inače, nacrti "kružić" (kao na grafu1) i podeli regiju na 4 podregije (kao na slici 2) prema centralnoj tački.
3. Ponavlja korake 1 i 2 sve dok ne ucrtta sve merne jedinice (kao na grafu 4).

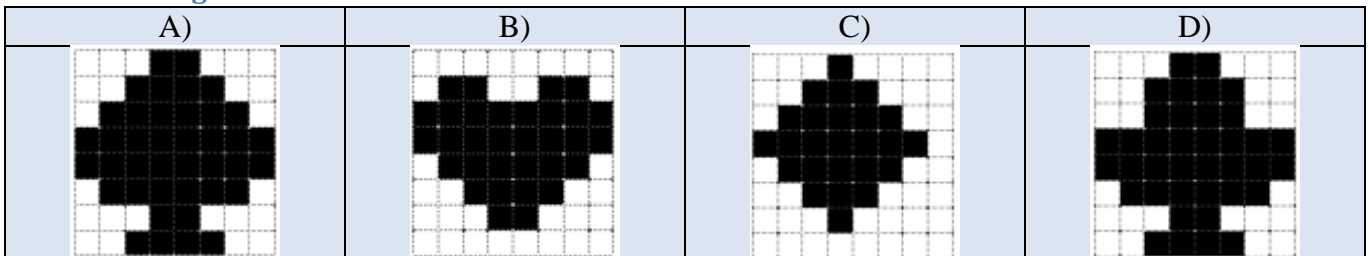


Pitanje/Izazov

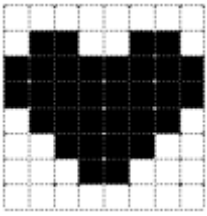
Dat je graf iz sveske dabra gusara. Koji od sledećih delića mape taj graf predstavlja?



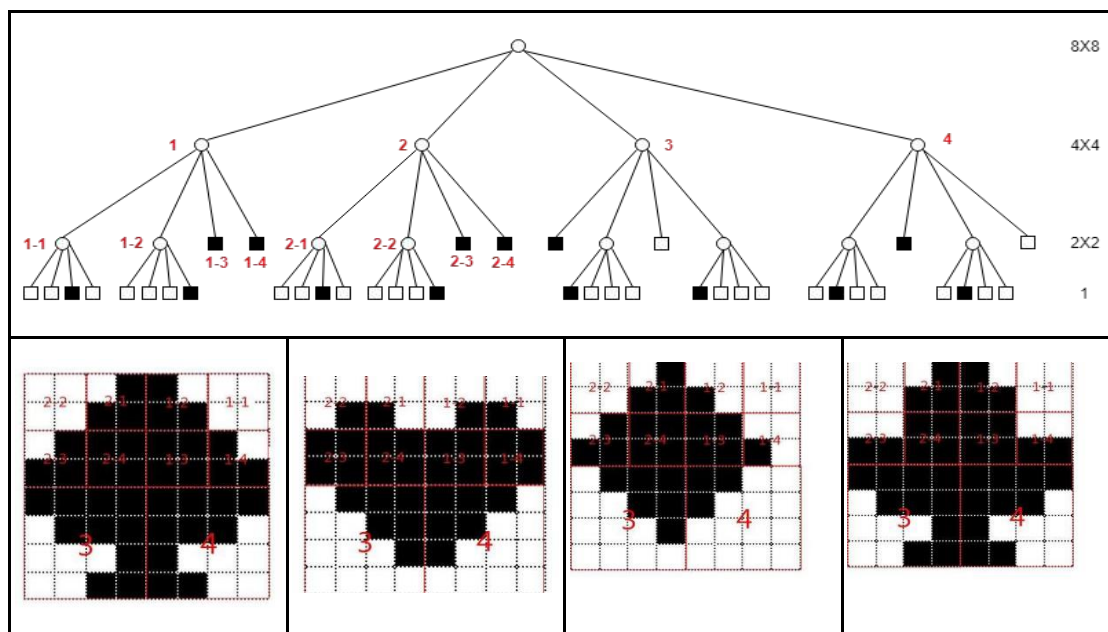
Ponuđeni odgovori:



Tačan odgovor je:



Na grafu možemo da obeležimo tačke koje predstavljaju regije veličine 4x4 sa brojevima od 1 do 4. Zatim, podregije regije 1 sa 1-1, 1-2, 1-3, 1-4, i tako dalje. Na taj način dobijamo odgovarajuće pozicije, kao na slici dole. 1-3, 1-4, 2-3 i 2-4 su obeležene crnim kvadratićem što znači da su te četiri regije veličine 2x2 na mapi crne. Samo (B) zadovoljava taj uslov.



Informatička pozadina

U računarstvu i informatici, graf sa četiri dela se naziva “Kvadrantno stablo” ili Q-stablo. Kvadrantno stablo je vrsta binarnog stabla čiji svaki unutrašnji čvor ima tačno četvoro dece. Sve forme kvadrantnog stabla dele neke zajedničke karakteristike:

- Razlažu prostor u prilagodljive ćelije
 - Svaka ćelija ima maksimalni kapacitet. Kada se dostigne maksimalni kapacitet, ćelija se deli
 - Stablo direktorijum prati prostornu dekompoziciju kvadrantnog stabla
1. Kvadrantno stablo se obično upotrebljava za predstavljanje slike (ovaj primer), obrađivanje slika, mesh generation,... itd. Više informacija možete naći na <https://en.wikipedia.org/wiki/Quadtree>.

Autor: Yin-yao Kao (Taiwan)

<http://dabar.edu.rs/>



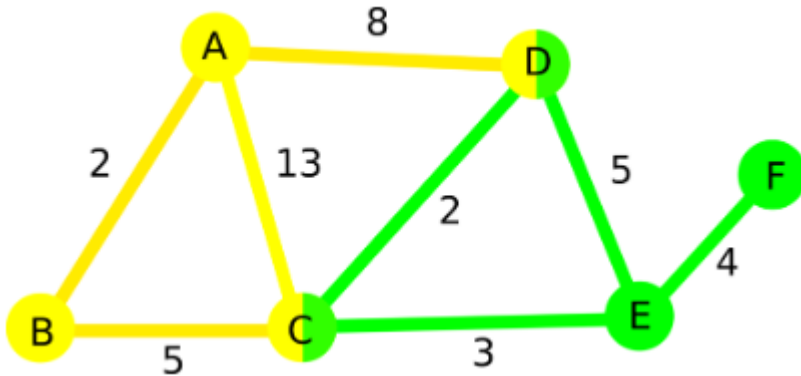


Međunarodna mreža podataka je prikazana na slici ispod. Svaka mreža ima čvorove iste boje. Čvorovi različitih boja ukazuju na veze između različitih mreža. U ovoj povezanosti postoji naplata rominga, pa ako se poruka prenosi sa čvora jedne mreže na drugu mrežu, cena prenosa je:

(Vrednost puta koju poruka mora preći u okviru prve mreže) + (Vrednost puta svih drugih mreža do primaoca) * 2.

Svaka grana mreže ima različitu cenu, prikazanu na grafiku.

Primer:



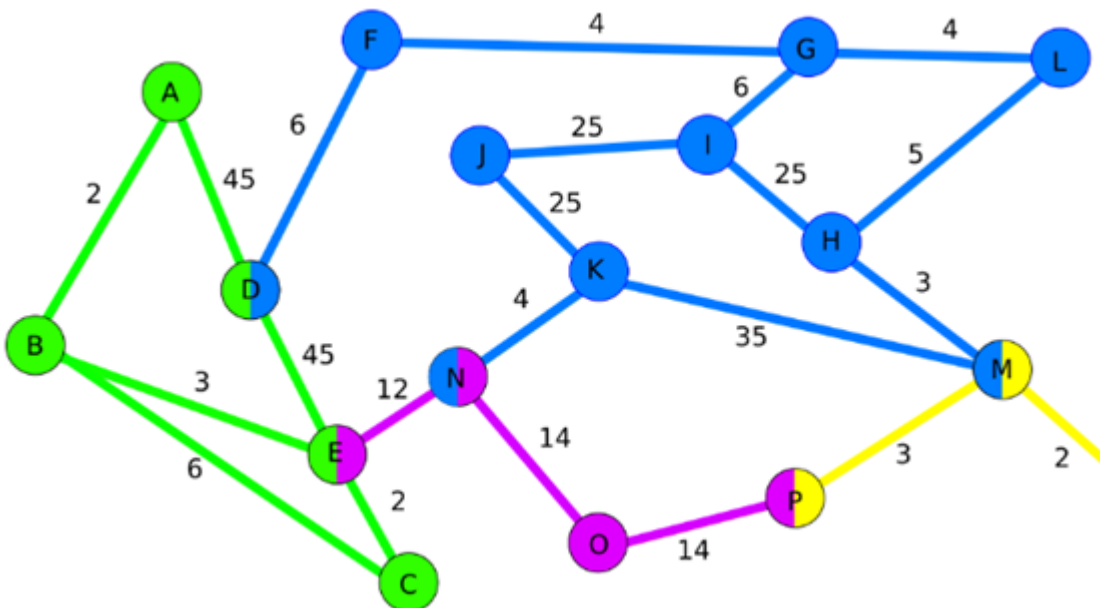
Najjeftiniji put za prenos informacija od A do F je: 21

A→B	B→C	C→E	E→F
2	5	3	4

$$2+5+(3+4)*2=21$$

Pitanje/Izazov

Koji je minimalni trošak za slanje poruke iz tačke A do tačke Q?



Ponudeni odgovori:

<http://dabar.edu.rs/>



- A) 45
- B) 93
- C) 95
- D) 111

Tačan odgovor:

B

Objašnjenje

Primetićemo da nije uvek put koji je najkraći ujedno put minimalnih troškova. Da bi prešli zelenu mrežu, i otišli u ljubičastu mrežu minimalni trošak je 5, ali od čvora E prelazeći preko N,K,M do Q, ili N,O,P,M do Q, troškovi su veći nego prelazak preko D, F, G, L, H, M do Q.

Zatim:

$$\text{Putanja: A,D,F,G,L,H,M,Q} = 45 + (6 + 4 + 4 + 5 + 3 + 2) * 2 = 93$$

$$\text{Putanja: A,B,E,N,K,M,Q} = (2 + 3) + (12 + 4 + 35 + 2) * 2 = 111$$

$$\text{Putanja: A,B,E,N,O,P,M,Q} = (2 + 3) + (12 + 14 + 14 + 3 + 2) * 2 = 95$$

Informatička pozadina

Zadatak se sastoji u prelaženju minimalnih puteva troškova od jednog čvora do drugih čvorova. Nas interesuju vezne tačke između podgrafa. Da bismo pronašli minimalne troškove možemo koristiti, na primer Dajkstrin algoritam. Dajkstrin algoritam u suštini pronalazi najkraću putanju u grafu, pa se tako koristi u mnoštvu oblasti uključujući računarsko umrežavanje. Takođe ima primeni u google mapama za nalaženje najkraćeg puta od jedne lokacije do druge. U biologiji se koristi za pronalaženje mrežnog modela u širenju zaraznih bolesti. Nažalost, zbog način na koji se najjeftiniji put izračunava u ovom zadatku, Dajkstrin algoritam se ne primenjuje.

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

Autor: *Laura Ungureanu (Rumunija)*

<http://dabar.edu.rs/>

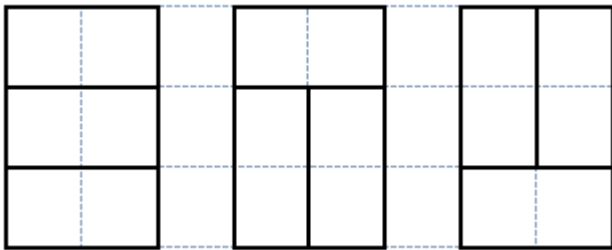




Dabar ima mnoštvo identičnih domina.



Želi da složi domine u kutiju. Domina može biti raspoređena vertikalno ili horizontalno u kutiju. Na primer, ako ima 3 domine, postoje 3 različita načina da ih složi u kutiju **dimenzija 3x2**.



Pitanje/Izazov

Na koliko različitih načina može složiti domine u kutiju **dimenzije 3x4**?



Ponudeni odgovori:

- A) 12
- B) 10
- C) 9
- D) 11



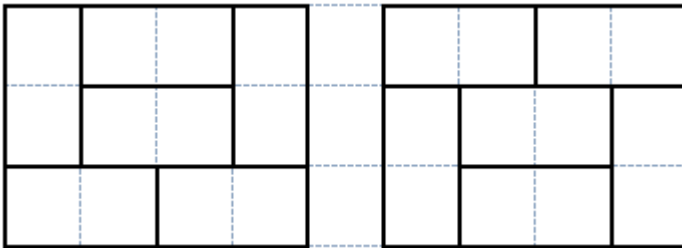
Tačan odgovor:

D) 11

Objašnjenje:

Do odgovora možemo doći tako što ćemo proveriti sve moguće načine slaganja domina. Naime, možemo podeliti veliku kutiju na dve manje kutije **dimenzija 3x2**. Postoje tri načina da to uradimo. Kako mala **dimenzije 3x2** ima **3 načina** za slaganje domina, u kutiji **dimenzija 3x4** na osnovu toga zaključujemo da imamo 9 načina slaganja domina.

Postoje još dva dodatna načina slaganja domina, prikazana na slici ispod.



Zbog toga, postoji ukupno $9+2=11$ načina ređanja domina u kutiju **dimenzija 3x4**.

Informatička pozadina

Slaganje domina zahteva geometrijske aranžmane. U geometriji, domine su oblici koji se formiraju ujedinjenjem dva kvadrata povezanih ivicama. Svaki pojedinačni raspored domina u kutiji možemo posmatrati kao mozaik sastavljen od domino pločica.

Induktivni argument je potreban da bi se otkrilo **9 mogućih mozaika** na osnovu korelacija između 3x2 kutija i 3x4 kutije, a zatim je potrebno dalje istraživanje da bi se otkrila još **2 moguća mozaika**.

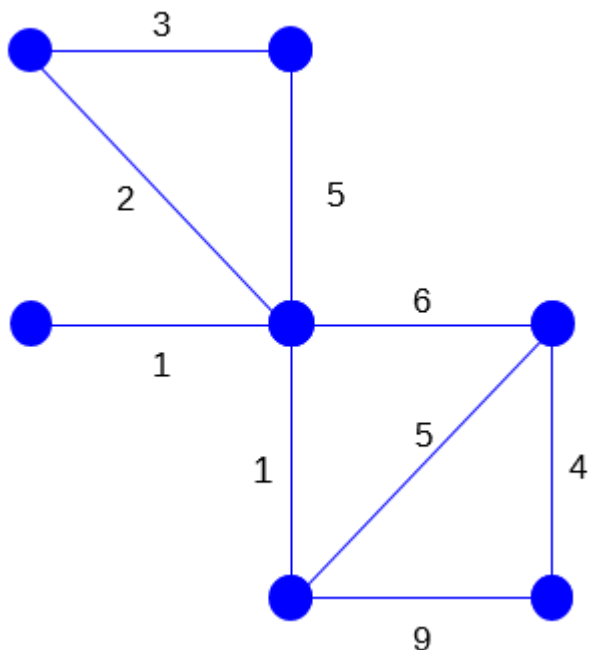
Autor: Maciej M. Sysło (Poljska)





Optički kablovi

Administrator računarske mreže želi da poveže 7 računarskih laboratorija tako da svaka laboratorija može da prima i šalje poruke do druge laboratorije preko optičkog kabla. Cene postojećih mrežnih konekcija (kablova) između laboratorija (plavi kružić) su prikazane na slici ispod.



Pitanje/Izazov

Koja je najmanja cena povezivanja svih računarskih laboratorija u postojećoj mreži?

Ponuđeni odgovori:

- A) 36
- B) 18
- C) 16
- D) 14

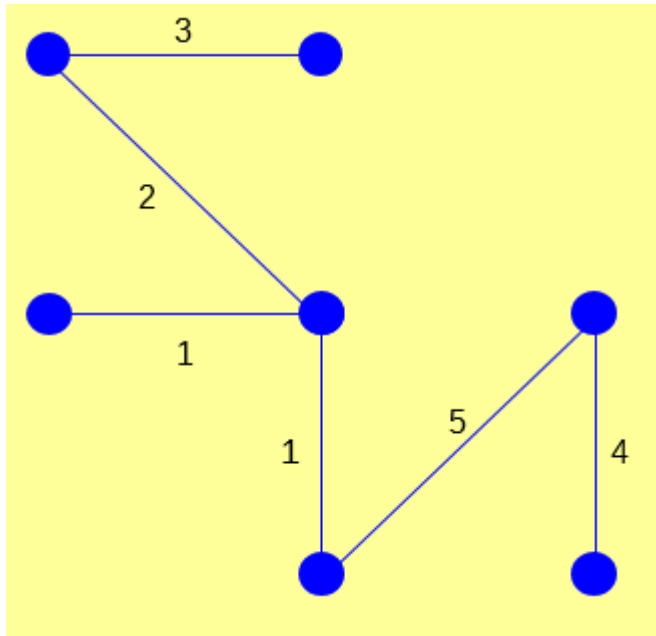


Tačan odgovor:

C) 16

Objašnjenje:

Tačan odgovor je 16. Odgovor se dobija nakon primene algoritma za nalaženje povezujećeg stabla minimalne cene. Odgovor (A) nije tačan, iako 36 jeste suma cena svih grana, ali nije najmanja cena povezivanja. Odgovor (B) nije tačan, jer cena 18 se može smanjiti (izborom grane cene 3 umesto grane cene 5). Odgovor (D) nije tačan jer ukupna cena 14 (uklanjanjem grane iz optimalnog povezivanja), jer ne obezbeđuje da rezultujući graf bude povezan (tako da svaka računarska laboratorija prima i šalje poruke sa drugim laboratorijama).



Informatička pozadina

Moramo da pronađemo poseban podgraf koji se zove povezujuće stablo uz dodatni uslov da to bude povezujuće stablo minimalne cene. Postoji više algoritama koji rešavaju ovaj problem u težinskom neusmerenom grafu. Na primer, algoritam Prim-aili algortam Kruskala.

Autor: Javier Bilbao (Španija)





Dabar Aleksa je na svom tavanu pronašao jednu staru računarsku mašinu. Želi da se poigra sa njom i da pokuša da sabere dva broja. Računarska mašina ima tri registra (memorijski prostor koji čuva broj), koji se označavaju **R1**, **R2** i **R3**. Da bi se ova mašina programirala, mora se uneti niz operacija koje mašina prepoznaje.

Evo mogućih operacija sa njihovim učinkom koje proizvode (**i** i **j** su brojevi registra, a **q** je broj operacije):

Zero (i)	Postavlja 0 u registar Ri
Inc (i)	Vrednost registra Ri se povećava za 1
Dec (i)	Vrednost registra Ri se umanjuje za 1
Store (i, j)	Kopira vrednost registra Rj u registar Ri
Jump (i, j, q)	Ako su vrednosti registra Ri i Rj iste, skoči na q-tu operaciju
JumpNeg (i, j, q)	Ako su vrednosti registra Ri i Rj različite, skoči na q-tu operaciju

Pitanje/Izazov

Dabar Aleksa želi da sabere dva cela, nenegativna broja tako što će ih upisati u registre **R1** i **R2**. Koji od sledećih programa sabira ta dva broja i postavlja konačan rezultat u **R1**?

Ponuđeni odgovori:

A) 1: Zero(3) 2: JumpNeg(1,3,5) 3: Inc(2) 4: Dec(1) 5: Jump(1,3,3) 6: Store(1,2)	B) 1: Zero(3) 2: Jump(1,3,5) 3: Inc(2) 4: Dec(1) 5: JumpNeg(1,3,3) 6: Store(1,2)	C) 1: Zero(3) 2: JumpNeg(1,3,5) 3: Inc(1) 4: Dec(2) 5: Jump(1,3,3) 6: Store(1,2)	D) 1: Zero(3) 2: Jump(1,3,5) 3: Inc(1) 4: Dec(2) 5: JumpNeg(1,3,3) 6: Store(1,2)
--	--	--	--



Tačan odgovor:

Tačan odgovor je B).

Objašnjenje:

Postoji mnogo različitih načina da se napiše ovaj program. Ali od ponuđena 4 odgovora, samo je odgovor B) ispravan. Odgovori A) i C) nisu ispravni, budući da ako je $R1 \neq 0$ (različito od 0), program će skočiti na operaciju 5, na koju neće skočiti i tako će R1 dobiti vrednost R2, što svakako nije zbir R1 i R2.

Odgovori B) i B) su slični, osim što su registri koji se koriste za operacije 3 i 4 zamenili mesta. Logika se sastoji u izvođenju petlje koja:

- 1) uvećava R2 za 1 i umanjuje R1 za 1 sve dok R1 ne stigne do 0, odnosno
- 2) uvećava R1 za 1 i umanjuje R2 za 1 sve dok R2 ne stigne do 0.

Na kraju,

- 1) zbir $R1+R2$ se nalazi u R2, odnosno
- 2) zbir $R1+R2$ se nalazi u R1

Problem odgovora D) je što se moraju uraditi dva skoka između R2 i R3 da bi petlja bila ispravna.

Informatička pozadina

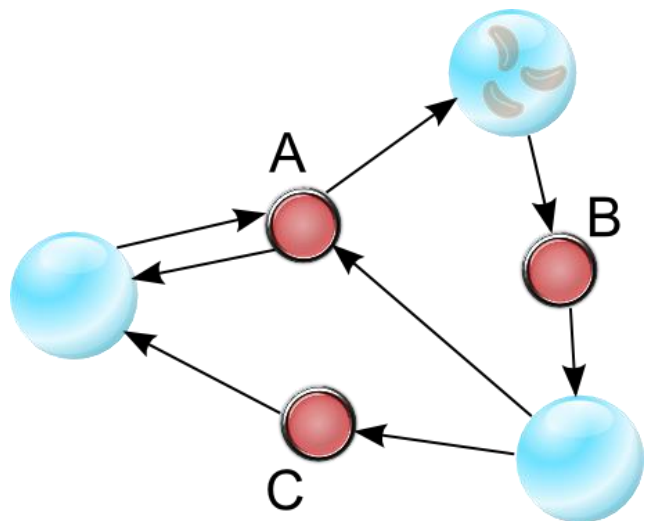
Odabir najefikasnijeg načina za postizanje je koristan u umrežavanju. To je optimizacijski problem koji ima za cilj pronalaženje staze u grafikonu koji minimizira određenu funkciju, što je ovde broj promjena antena.

Autor: (Belgija)





Dabar Milan se igra sa vrlo čudnom mašinom. Mašina je sastavljena od staklenih mehurića koji sadrže zrna pasulja. Ovi mehurići su međusobno povezani velikim tasterima. Evo slike igre:



Kada pritisnete taster, dve stvari se dešavaju jedna za drugom:

- Mašina proverava da li postoji najmanje jedno zrno pasulja u svim mehurićima koji su povezani sa tasterom (to jest, tu je strelica od mehurića do tastera)
- Ako je provera uspešna, jedan pasulj nestaje iz svih izvornih mehurića, a jedno zrno pasulja se dodaje svim ciljnim mehurićima (to jest, postoji strelica od tastera do mehurića).

Na primer, pritiskom na taster B će se ukloniti jedno zrno pasulja sa gornjeg mehurića i stvoriti jedno zrno pasulja u donjem mehuriću.

Pitanje/Izazov

Koji od sledećih nizova kombinacija tastera dovodi mašinu u situaciju u kojoj nije moguće ništa menjati na mašini, bez obzira na to koji je taster pritisnut?

Ponuđeni odgovori:

- A) B – B – C – A – B – A
- B) B – C – B – C – B – A
- C) B – B – C – B – C – C
- D) B – C – B – B – A – A



Tačan odgovor

Tačan odgovor je pod C.

Objašnjenje

Ideja da se mašina dovede u deadlock blokadu jeste da uzme sva zrna pasulja iz mehurića sa leve strane. Okida tri puta taster B, a zatim taster C (ili drugim redosledom ova dva tastera) prebacuje zrna tamo. Svi ostali predloženi scenariji ne dovode mašinu u deadlock blokadu.

Informatička pozadina

Ovaj problem predstavlja ilustraciju Petrijevih mreža, formalizma koji se koristi za opisivanje trenutnih reaktivnih sistema i koji se mogu koristiti za simuliranje njihovog ponašanja ili za vršenje analize na njima. Mogućnost simulacije Petrijeve mreže pomaže da se razume kako se ponašanje kompleksnih istovremenih reaktivnih sistema može modelirati.

Autor: Sébastien Combéfis (Belgija)

<http://dabar.edu.rs/>

