Dabar_2017/18.

Zadaci i rešenja za učenike srednje škole
– školsko takmičenje
(novembar, 2017.)

Prijatelj takmičenja,

BIGZ školstvo

Sadržaj

***

Izbor zadataka za takmičenje i prevod, Programski odbor takmičenja:

Milan Rajković (predsednik programskog odbora)
Svetlana Jakšić (član programskog odbora)
Milan Lukić (član programskog odbora)
Bojan Milosavljević (član programskog odbora)
Marija Andonović Radojević (član programskog odbora)
Ivica Bekrić (član programskog odbora)
Saša Jevtić (član programskog odbora)

Tehnična podrška: Branislav Dolić

O takmičenju i priručniku

Draga deco 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 5. godinu zaredom družimo. U proteklom periodu na takmičenju je učestvovalo preko 130.000 takmičara, a na ovogodišnjem školskom nivou, čak 42.539. 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 2017/18., 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 razijaju 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 sa zajedno sa nama radujete novim zadacima.

Uživajte u rešavanju zadataka!

Srdačno Vaš,

Programski odbor takmičenja Dabar

Bodovna tabela

Dabar

Zadatak	Težina	Broj bodova
Skakači	Easy	6
Kuglice	Easy	6
Ribice	Easy	6
Roštilj	Easy	6
Uljez	Medium	9
Peciva	Medium	9
Sakupljanje bombona	Medium	9
Najkraće rastojanje	Medium	9
Tuneli brane u Humskoj	Hard	12
Robot čistač	Hard	12
Robot	Hard	12
Tandem (ponavljanje nizova)	Hard	12

Stariji dabar

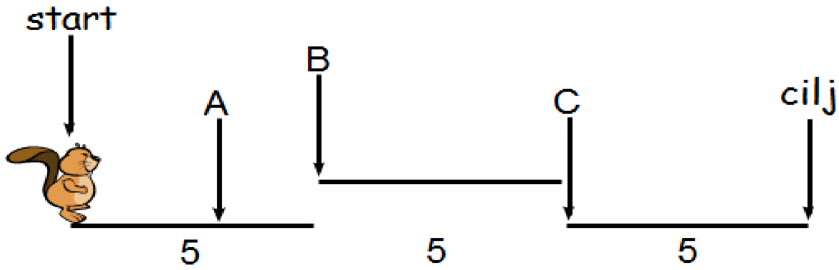
Zadatak	Težina	Broj bodova
Uljez	Easy	6
Peciva	Easy	6
Sakupljanje bombona	Easy	6
Roštilj	Easy	6
Tuneli brane u Humskoj	Medium	9
Najkraće rastojanje	Medium	9
Tandem (Ponavljanje nizova)	Medium	9
Soda Shop	Medium	9
Robot čistač	Hard	12
Robot	Hard	12
Kompresija ikonica	Hard	12
Savršeni partneri	Hard	12

Skakači

Milan i Saša igraju jednostavnu video igricu. Cilj video igre je da pomere Dabra od starta do cilja, skakanjem sa platforme na platformu.

Platforme su na dva nivoa: „donja“, sa koje Dabar uvek kreće i „gornja“. Vreme potrebno za prelazak svake platforme je prikazano ispod platforme.

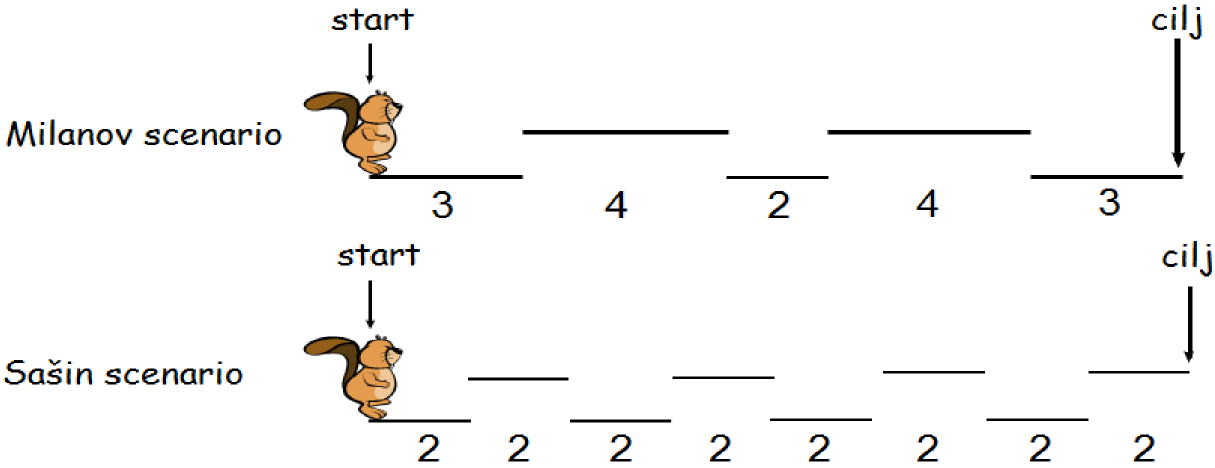
Na primer:



- Dabar je na tački A, 3 sekunde nakon starta;
- Dabar je na tački B, 5 sekundi nakon starta;
- Dabar je na tački C, 10 sekundi nakon starta;
- Dabar je na cilju 15 sekundi nakon starta.

Obratite pažnju na to da Dabar trenutno može skočiti na gronju ili donju platformu.

Milan i Saša počinju da igraju igru u istom trenutku. Oni igraju različite scenarije:



Pitanje:

Koliko sekundi se istovremeno oba dabra (Milanov i Sašin) kreću po gornjoj platformi?

Odgovori:

A. 2
B. 4
C. 6
D. 8

Tačan odgovor je B: 4

Možemo doći do odgovora tako što ćemo zapisati „kretanje po donjoj platformi u trajanju od 1 sekunde“ kao 0, a „kretanje po gornjoj platformi u trajanju od 1 sekunde“ kao 1.

Milanova igra može se predstaviti kao:

0001111001111000

Sašina igra može se predstaviti kao:

0011001100110011

Da bi našli vreme za koje su oba dabra na gornjim nivoima istovremeno, moramo pronaći vremena kada obe igre imaju vrednost 1, istovremeno. Ovo možemo uraditi primenom Boolean logičke funkcije AND na ove dve sekvence:

0001111001111000

0011001100110011

0001001000110000

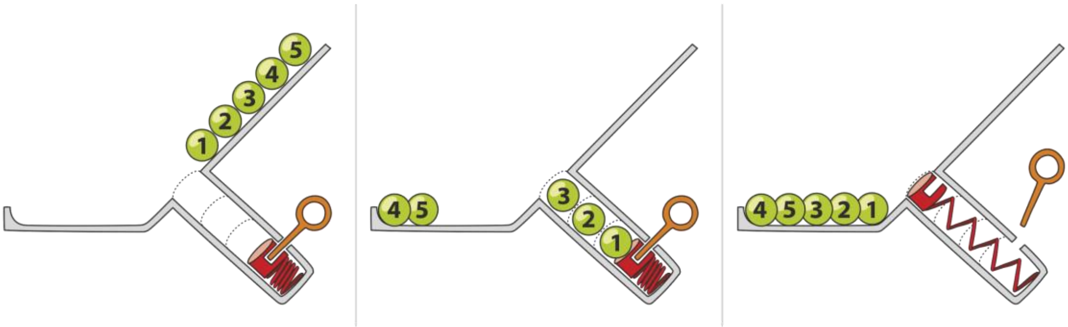
<u>Informatička pozadina</u>

Jedna od najvažnijih filozofija u računarskoj nauci je kako predstaviti informacije. U ovom zadatku možemo da „sakrijemo“ neke detalje igre da bi se fokusirali na ono što je važno, što je u ovom slučaju situacija kada je igrač na gornjem nivou. Zatim, kada su informacije predstavljene precizno, možemo transformisati ili kombinovati informacije na nove i važnije načine. Konkretno, za ovaj problem, mi tretiramo informacije kao sekvence binarnih brojeva i izvršavamo važnu AND operaciju da bi pronašli gde obe sekvence imaju istovremeno vrednost 1.

Kuglice

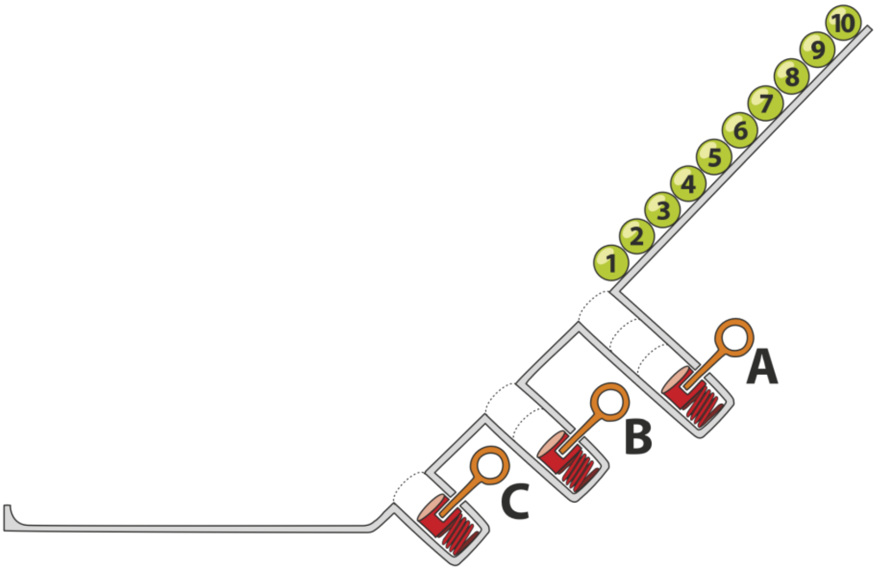
Numerisane kuglice spuštaju se niz rampu na kojoj se nalaze rupe. Kada kuglica dođe do rupe, ukoliko ima dovoljno mesta, kuglica upada u nju, u suprotnom, kuglica prelazi preko rupe.

Na dnu svake rupe se nalazi opruga koja je zaključana. Ključ se može izvući i tada opruga izbacuje kuglice.



Pre nego što 5 kuglica počnu da se kotrljaju.	Nakon što kuglice prestanu da se kotrljaju.	Konačan rezultat, nakon što se izvuče ključ.

Deset kuglica kotrlja se niz rampu, kao na slici ispod. Tri rupe: A, B i C imaju mesta za 3, 2 i 1 kuglicu. Ključevi otključavaju opruge iz rupa, po redosledu A, B, C, ali tek nakon što kuglice popune rupe ili pređu preko rupa.



Pitanje:

Koji od sledećih odgovora je konačan rezultat?

Odgovori:

 A	 B	 C	 D

Tačan odgovor je: D

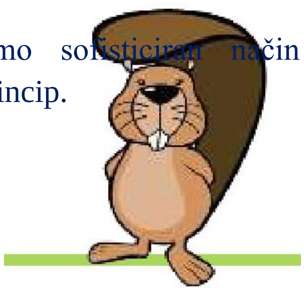
Rupa A ima mesta za tri kuglice, tako da kuglice od 4 do 10 prelaze preko nje u svom početnom redosledu. Rupa B ima mesta za dve kuglice, tako da kuglice od 6 do 10 prolaze preko nje u početnom redosledu. Rupa C ima mesta za jednu kuglicu, tako da kuglice od 7 do 10 prolaze u svom početnom redosledu. Zatim ključ A u rupi jedan povlačimo i kuglice su izbačene u redosledu 3, 2, 1 i kotrljaju se prema dnu. U tom trenutku redosled kuglica na dnu je 7, 8, 9, 10, 3, 2, 1. Zatim ključ u rupi B se povlači, i kuglice su izbačene u redosledu 5, 4. U ovom trenutku, kuglice na dnu su u redosledu 7, 8, 9, 10, 3, 2, 1, 5, 4. Konačno, ključ u rupi C se povlači i kuglica 6 se kotrlja prema dnu pri čemu dobijamo odgovor i tačan redosled kuglica na dnu.

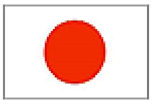
Informatička pozadina

Rupe se ponašaju kao STOK, što je struktura podataka. To je način organizovanja podataka. Ovo je primer last-in-first-out principa (LIFO). Na primer, prva kuglica koja upadne u rupu, je poslednja koja izlazi iz rupe.

Uprkos tome što je veoma jednostavna ideja, korisna je u mnogo drugih situacija. Na primer, možda želite da istražite kako se STOK može iskoristiti da bi se utvrdilo da su zagrade u izrazu izbalansirane ((1+2)*3), a zagrade u izrazu ((4+5)*(6-7) nisu. Ideja je da se sve otvorene zagrade postave u STOK (Ova operacija se naziva „Pritisak“) i kada se pronađe odgovarajući zatvarač, otvorena zagrada se uklanja sa vrha STOKA (Ova operacija se zove POP).

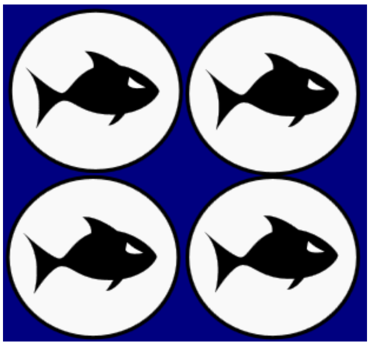
Umesto da izmišljamo komplikovan algoritam ili da upotrebljavamo sofisticiran način organizovanja podataka, ispostavlja se da nam je potreban samo poslednji princip.





Ribice

Četiri igračke ribica su postavljene na poseban poslužavnik, kao što je prikazano na slici ispod:



Ako okreneš bilo koju za 45^{o} u smeru kazaljke na satu, ribica dijagonalno od nje će se okrenuti za 45^{o}, ali u smeru suprotnom od kazaljke na satu.

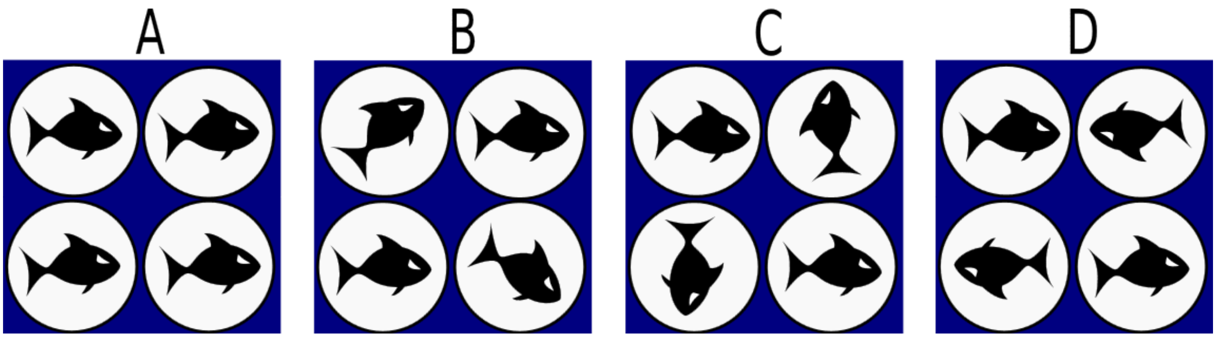
Uradi sledeće:

1. Okreni ribicu u gornjem levom uglu za 45^{o} u smeru kazaljke na satu.
2. Okreni ribicu u donjem levom uglu za 90^{o} u smeru kazaljke na satu.
3. Okreni ribicu u donjem desnom uglu 90^{o} u smeru kazaljke na satu.
4. Okreni ribicu u gornjem levom uglu za 45^{o} u smeru kazaljke na satu.

Pitanje:

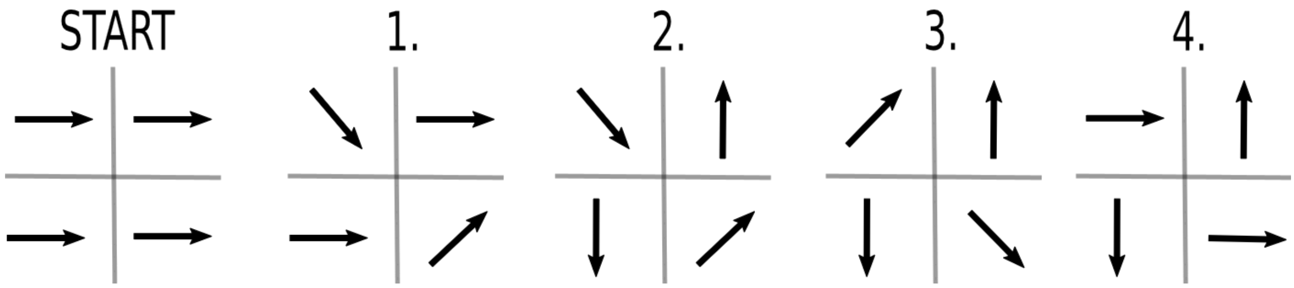
Koja slika prikazuje položaj ribica sada?

Odgovor:



Tačan odgovor (C)

Učenik koji pokuša da prati ove korake, usmenim putem, verovatno će biti zbunjen. Jednostavniji način za rešavanje ovog zadatka daje pouzdanost. Jedan od načina jeste napraviti brzu i jednostavnu belešku za praćenje promena. Sistem koji je gore prikazan je jasan i lako se izvodi ako se koriste strelice umesto ribica.



Na drugi način zadataka se može brzo rešiti čitanjem svih operacija i zapažanjem da samo korak br. 2 utiče na igračke na donjem levom i gornjem desnom uglu. Zbog toga, učenik može jednostavno pogledati ponuđene odgovore i odabrati jedan, sa donjom levom i gornjom desnom ribicom u ispravnom položaju.

Roštilj

Dabrovi na Dabar Akademiji organizuju proslavu završene školske godine. U svakom trenutku, između 10,00 i 20,00 potreban im je jedan dabar za proveru na ulazu. Neki od dabrova su se dobrovoljno javili da pomognu. Dali su vreme kada mogu biti na ulazu.

Međutim, u dole navedenoj listi, još uvek postoji vremenski period u kome nijedan dabar ne proverava ulaz.

11,00-12,00	15,30-16,30	19,00-20,00
10,00-10,30	10,15-11,15	19,15-19,30
17,15-17,45	14,00-15,00	16,15-17,30
18,15-19,00	17,30-19,00	12,00-13,30
13,45-14,30	14,45-16,00	

Pitanje:

Pronađi vremenski period u kom nema nijednog dabra na ulazu?

Odgovori:

A. 13,30 - 13,45
B. 16,30 - 17,15
C. 15,00 - 15,30
D. 10,30 - 11,15

Tačan odgovor je pod A. 13,30 - 13,45

Korak 1: sortiramo zadate vremenske intervale provećavajući vreme početka.

10,00	**10,15**	**11,00**	**12,00**	**13,45**	**14,00**	**14,45**	**15,30**	**16,15**	**17,15**	**17,30**	**18,15**	**19,00**	**19,15**
10,30	**11,15**	**12,00**	**13,30**	**14,30**	**15,00**	**16,00**	**16,30**	**17,30**	**17,45**	**19,00**	**19,00**	**20,00**	**19,45**

Korak 2:

Zatim skeniramo intervale u ovom redosledu, spajamo sve susedne segmente koji se preklapaju.

Na kraju, na takav način, dobijamo dva intervala: 10,00 - 13,30 i 13,45 - 20,00

Odgovor je pod A. 13,30 - 13,45

Informatička pozadina:

Poželjni algoritmi i sortiranje su važne i često korišćene metode za rešavanje zadataka u računarstvu. Često se brže rešavaju problemi za sortirane nego za nesortirane podatke. Na primer, traženje elementa podataka, pronalazak duplikata, ...

Postoji nekoliko efikasnih sortirnih algoritama, često ilustrujući standardne algoritamske strategije dizajna.

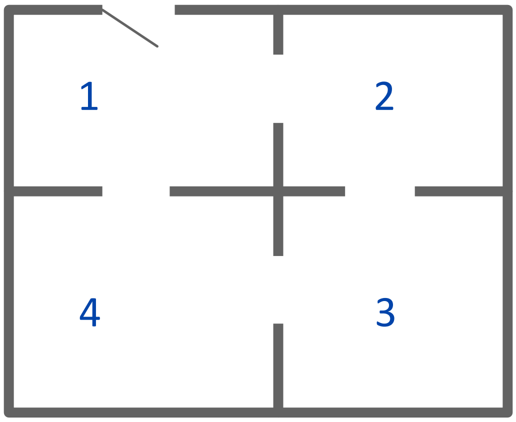
Uljez

U muzeju Moderne umetnosti Dabargrada postoji pametni siguronosni sistem koji otkriva uljeze. Uljez je dabar koji je ušao u muzej, a pri tom nije koristio ulaz.

Kada dabar uđe ili izađe iz sobe, sistem prati promenu i u tabeli beleži brojno stanje dabrova u sobama. Može se dogoditi da nekoliko dabrova uđe ili izađe iz sobe istovremeno.

Tabela prikazuje beleške pametnog siguronosnog sistema, a slika pored prikazuje raspored soba u muzeju.

Vreme	Soba 1	Soba 2	Soba 3	Soba 4
10,00	2	0	0	0
10,07	3	0	0	0
10,08	2	1	0	0
10,12	4	1	1	0
10,13	2	2	3	0
10,17	5	2	2	1
10,20	4	1	2	2



Pitanje:

U kom vremenu je sistem otkrio uljeza?

Odgovori:

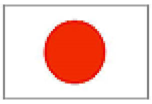
- 10,13
- 10,17
- 10,12
- 10,20

Tačan odgovor je 10,13.

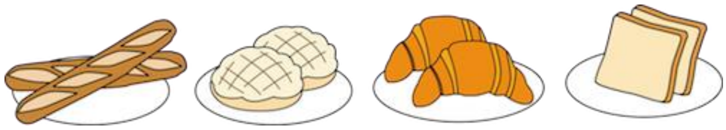
Policija je pozvana u 10,13. Tada su u sobu 3 ušla dva dabra, a prema priloženim beleškama ranije je bio samo jedan dabar (soba 2). Dakle, neko je ušao u sobu 3 izvan muzeja ko nije koristio ulaz.

Informatička pozadina

Siguronosni sistemi koji prate broj **kritičnih** osoba koje se mogu naći na mestima poput aerodroma. Računarski programi vrednuju slike kamera, otkrivaju osobe i računaju ih. Ovi programi koriste veštačku inteligenciju (na primer, prepoznaju ljude), ali i jednostavna logična pravila poput ovog zadatka za otkrivanje kršenja sigurnosti.



Peciva



Na stolu je nekoliko različitih vrsta peciva:

- Dva francuska hleba
- Dve krofne
- Dva kroasana
- Dva tosta

Četiri dabra: Svetlana, Milan, Branko i Suzana, dele peciva tako da svaki dabar dobije dve različite vrste peciva.

Nakon podele peciva poznato nam je sledeće:

1. Svetlana i Milan nemaju iste vrste peciva;
2. Branko ima francuski hleb;
3. Suzana ima krofnu, a Svetlana nema krofnu;
4. Milan ima kroasan.

Pitanje:

Koje vrste peciva ima Svetlana?

Odgovori:

A. Hleb i kroasan
B. Krofnu i tost
C. Hleb i tost
D. Krofnu i kroasan

Tačan odgovor je pod C. (Hleb i tost)

	Svetlana	Milan	Branko	Suzana
Hleb			Da (Milan)	
Krofna	Ne (Branko)			Da (Branko)
Kroasan	Ne (Svetlana/Suzana)	Da (Suzana)		
Tost				

<u>Informatička pozadina</u>

Matematička logika proučava osnovne principe matematičkih zaključaka. Bulova algebra je deo matematičke logike. Bazira se na principima deduktivnog logičkog zaključivanja, podaci o ulazu mogu imati samo dve faze: ispravno ili pogrešno. "Tabela istine" izražava odnose između dabrova i vrsta peciva.

Sakupljanje bombonâ

Slatko Robotić je programiran da sakupi što je moguće više bombonâ. On to radi dok se šeta po ćelijama tabele. Svaka ćelija u tabeli ispod sadrži: 0, 1, 2 ili 3 bombone.

Slatko počinje od ćelije ***S* (start)**, u donjem levom uglu i završava u ćeliji ***K* (kraj)**, u gornjem desnom uglu. Dok sakuplja bombone, Slatko Robotić može da se kreće samo nagore i nadesno (iz pozicije posmatrača tabele).

2	0	1	1	***K***
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
S	0	1	3	0

↑

→

Pitanje:

Koji je najveći broj bombonâ koji Slatko Robotić može da sakupi?

Odgovori:

A. 10
B. 12
C. 14
D. 16

Tačan odgovor je pod C: 14

Jedan pristup rešenju je da se popuni tabela „najboljih” mogućih količina bombonâ koje se mogu sakupiti kretanjem „po dijagonali”. Na početku, ukupno je sakupljeno 0 bombonâ, tako da se tabela može popuniti na sledeći način:

2	0	1	1	K
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	0	1	3	0

gde je podebljani element maksimalan broj bombonâ koji se može postići u svakoj ćeliji. Ako se krene nagore dobiće se 3 bombone, a nadesno dobiće se 0 bombona, tako da se to može uneti u ukupne rezultate kao što sledi:

2	0	1	1	K
1	2	0	2	3
2	2	0	2	1
3	1	0	2	0
0	**0**	1	3	0

Treba zapaziti ćeliju koja nudi 1 bombonu, koja je nadesno od podebljane 3 i iznad podebljane 0. Koja je maksimalna količina bombonâ koja bi se mogla sakupiti do dolaska u ovu ćeliju? Trebalo bi doći u ovu ćeliju posle sakupljene 3 bombone (bolje nego 0 po drugom mogućem pravcu). Tako se može biti u ovoj ćeliji sa ukupno sakupljene 4 bombone.

2	0	1	1	K
1	2	0	2	3
2	2	0	2	1
3	**4**	0	2	0
0	**0**	1	3	0

Ako se nastavi na ovaj način, može se primetiti da je maksimalan broj bombonâ koji možemo sakupiti u ćeliji, broj bombonâ koji se dobija u toj ćeliji plus veći broj od maksimalnih brojeva bombonâ koji se dobijaju u ćeliji levo i u ćeliji ispod one u kojoj se trenutno razmatra maksimalni broj bombonâ. Ovo se može zapisati matematički na sledeći način:

$$v(i,0) = 0$$
$$v(0,j) = 0$$
$$v(i,j) = c(i,j) + \max\{v(i\text{-}1,j),\ v(i,j\text{-}1)\}$$

gde je $v(i,j)$ maksimalan broj bombonâ koji se može sakupiti u ćeliji (i,j), a $c(i,j)$ je početni broj bombonâ u ćeliji (i,j).

Zato što se uvek želi proveriti leva i ćelija ispod, treba dodati kolonu nula levo i red nula na dnu tabele.

Primenjujući uspostavljenu matematičku vezu, može se popuniti ostatak tabele na sledeći način:

0	**8**	**9**	**10**	**12**	**14**
0	**6**	**9**	**9**	**11**	**14**
0	**5**	**7**	**7**	**9**	**10**
0	**3**	**4**	**4**	**6**	**6**
0	**0**	**0**	**1**	**4**	**4**
0	*0*	*0*	*0*	*0*	*0*

Stoga, može se sakupiti najviše 14 bombonâ u ćeliji K.

Informatička pozadina

Određivanje „najboljeg“ rešenja u skupu mogućih rešenja je težak i koristan problem. Konkretno, za ovaj problem sakupljanja bombonâ mogle bi se ispitati sve moguće putanje, što predstavlja rešenja primenom postupka „grube sile“. Nažalost, postoji veliki broj putanja: tačno 70 različitih putanja za ovaj problem (ovo je dobra vežba sa *Paskalovim trouglom*).

Međutim, u ovom posebnom slučaju mogu se pokušati pronaći neki „vredni“ delovi zadatka i pokušati (i uspeti) da se odatle pronađe najbolje rešenje. Budući da je tabela relativno mala, može se zaključiti da sve druge mogućnosti moraju biti lošije.

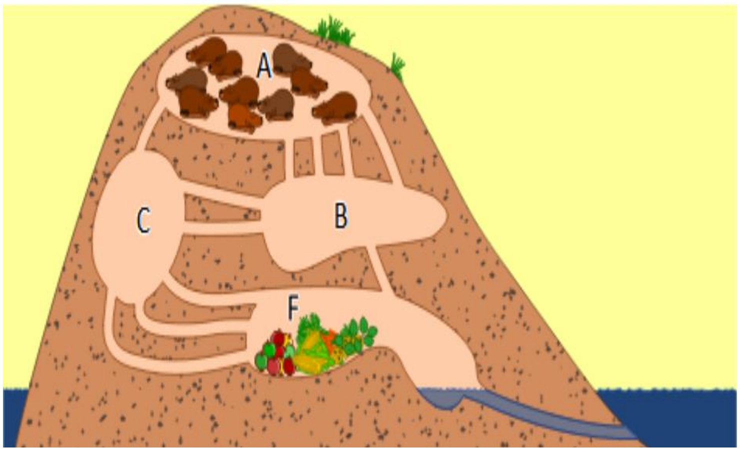
Efikasnije rešenje podrazumeva popunu tabele, kao u našem rešenju, korišćenjem postupka koji se zove *memorizacija* rekurentne veze *dinamičkog programiranja*. To znači da kada se dobije formula/veza „najboljeg“ rešenja za trenutnu ćeliju na osnovu ćelije levo ili ispod, mogu se u našem slučaju primeniti 25 izračunavanja da se izračuna maksimalan broj dostupnih bombonâ, idući tako od početnih uslova do rešenja.

Tuneli brane u Humskoj

Brana u Humskoj ulici ima tunele koji povezuju 4 sobe (A, B, C, F). Prve tri sobe (A, B, C) su dnevne sobe, a četvrta (F) je soba u kojoj se čuva hrana (pogledaj sliku).

Deset dabrova se nalazi u sobi A i veoma su gladni, tako da žele da idu u sobu F. Oni žele što pre da stignu do sobe F.

Potreban je 1 minut da se prođe jedan tunel. Pravilo je: dok se jedan dabar nalazi u tunelu, drugi ne sme ući u taj tunel (ne mogu ići jedan za drugim, već kada jedan dabar izađe iz tunela, tek tada drugi dabar ulazi u taj tunel).



Sobe su povezane određenim brojem tunela:

- između sobe A i sobe B postoje 4 tunela
- između sobe A i sobe C postoji 1 tunel
- između sobe B i sobe C postoje 2 tunela
- između sobe C i sobe F postoje 3 tunela

Sobe nemaju ograničenja tako da svaka od soba može primiti sve dabrove.

Pitanje:

Koliko je najmanje minuta potrebno da svi dabrovi stignu u sobu sa hranom (sobu F)?

Odgovori:

A. 4

B. 6

C. 7

D. 8

Tačan odgovor je poda A: 4

Objašnjenje:

	Broj dabrova u sobi			
	A	**B**	**C**	**F**
Situacija na startu	**10**	**0**	**0**	**0**
Tri dabra idu iz A u B				
Jedan dabar ide iz A u C				
Situacija posle 1 minuta	**6**	**3**	**1**	**0**
Tri dabra idu iz A u B				
Jedan dabar ide iz B u F				
Dva dabara idu iz B u C				
Jedan dabar ide iz C u F				
Jedan dabar ide iz A u C				
Situacija posle 2 minuta	**2**	**3**	**3**	**2**
Jedan dabar ide iz A u B				
Jedan dabar ide iz B u F				
Dva dabara idu iz B u C				
Jedan dabar ide iz A u C				
Tri dabara idu iz C u F				
Situacija posle 3 minuta	**0**	**1**	**3**	**6**
Jedan dabar ide iz B u F				
Tri dabara idu iz C u F				
Situacija posle 4 minuta	**0**	**0**	**0**	**10**

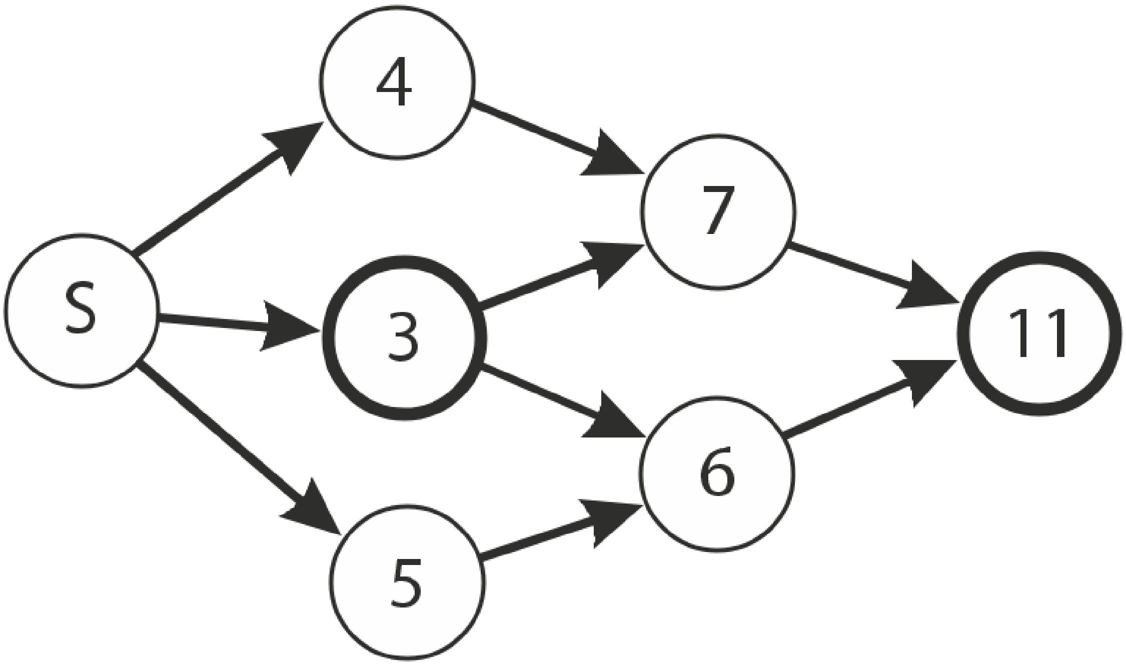
Informatička pozadina

Mrežu tunela možemo posmatrati kao mrežu protoka u teoriji grafova. Može se reći da su grafovi sastavljeni od tačaka, odnosno čvorova (vrhova), i linija između njih, odnosno grana. U ovom slučaju, u pitanju je tzv. usmereni graf, gde svaka grana ima svoj pravac.

Cilj je da optimizujemo protok (u ovom slučaju dabrove), tako da što više njih u najkraćem vremenu stigne do hrane. Proučavanje algoritama koji rešavaju probleme upotrebom grafova predstavlja veoma značajan deo informatičke nauke. Mreže imaju mnogo primena u proučavanju praktičnih aspekata teorije grafova i to se zove analiza mreža. Analiza mreža je posebno značajna za probleme modeliranja i analiziranje mrežnog saobraćaja, recimo interneta. Postoji više algoritama za rešavanje ovog problema, a jedan od njih je Ford-Fulkersonov algoritam.

Najkraće rastojanje

Na slici je prikazana jednosmerna mreža puteva. Broj u svakom krugu označava najkraće rastojanje od kruga S do kruga u kome je upisan broj.



Pitanje:

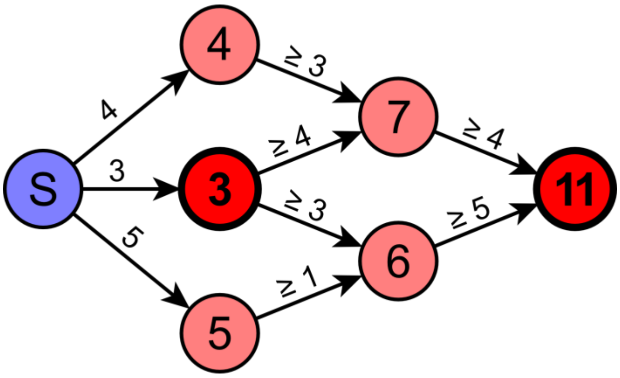
Koja od sledećih izjava je tačna, za dva kruga sa podebljanim ivicama?

Odgovori:

A. Najkraće rastojanje između ova dva kruga je 14
B. Najkraće rastojanje između ova dva kruga je 19
C. Najkraće rastojanje između ova dva kruga je 8
D. Najkraće rastojanje između ova dva kruga je 16

Tačan odgovor je pod C

Ako bi najkraće rastojanje bilo manje od 8, onda bi najkraće rastojanje između S i čvora 11 trebalo da bude manje od 11. Najkraće rastojanje može biti veće od 8, zato što najkraća putanja od S do čvora 11 može pratiti spoljašnje čvorove mreže.



Informatička pozadina

Ovaj zadatak se bavi mrežama i poznatim problemom najkraćeg puta. Više o tome na Vikipediji: https://en.wikipedia.org/wiki/Shortest_path_problem.

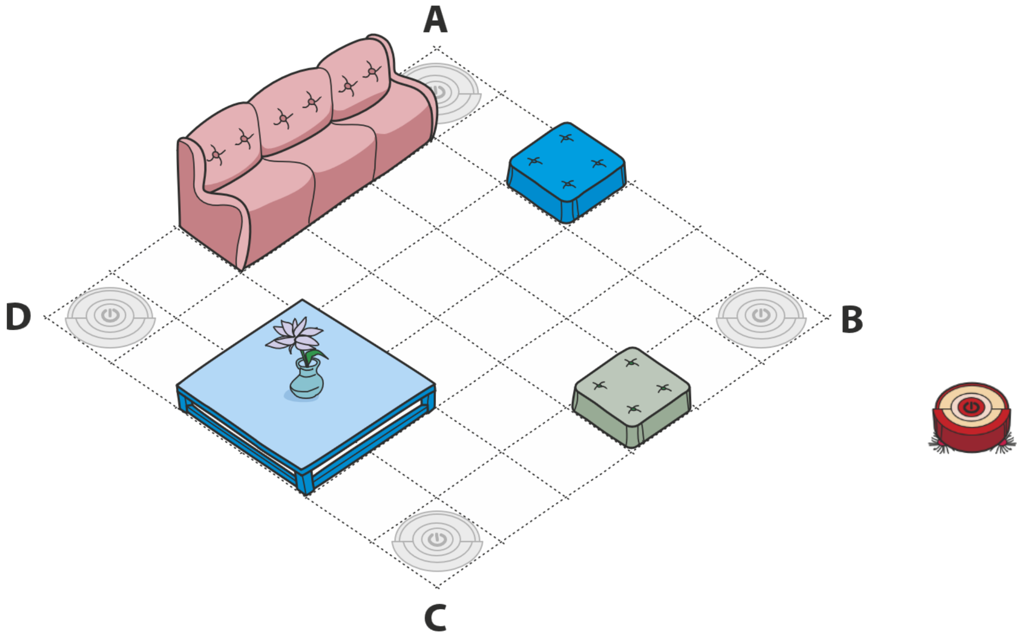
Ovaj problem može se rešiti primenom Dijkstrin algoritma. U zadatku se od učenika traži da razmišljaju o najkraćem rastojanju između dva čvora unutar mreže, umesto da samo razmatraju početne i krajnje tačke mreže.

Robot čistač

Robot čisti kvadratnu površinu poda koristeći sledeće komande:

- F - pomeri se napred za 1 pločicu (potreban je 1 minut)
- R - okreni se za 90° udesno (odmah se izvršava)
- W - očisti pločicu (potreban je 1 minut)

Robot počinje i završava u jednom od četiri ugla (A, B, C, D) - ali nije neophodno da završi na istoj pločici.



Pitanje:

Koliko je najmanje minuta potrebno da robot počisti slobodnu površinu poda?

Odgovori:

A. 28

B. 38

C. 55

D. 75

Tačan odgovor je: C

Tačan odgovor je pod C. 55.

Postoji 36 - 9 = 27 pločica poda, tako da čišćenje zahteva 27 minuta.

Ako bi stao na svaku pločicu jednom, bilo bi potrebno 26 minuta samo za pomeranje (računajući da ne mora da dođe do početne pločice).

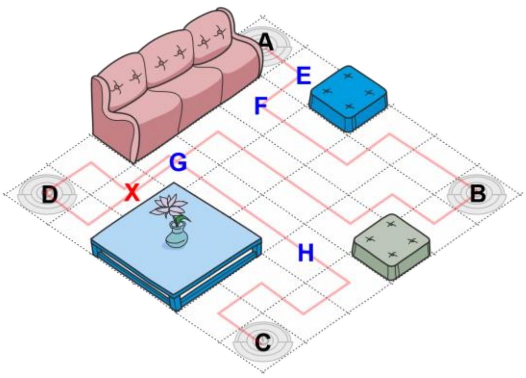
Pogledajte sliku sa leve strane. Postoji deo **X** preko koga mora da se pređe dva puta, bez obriza na odabranu putanju.

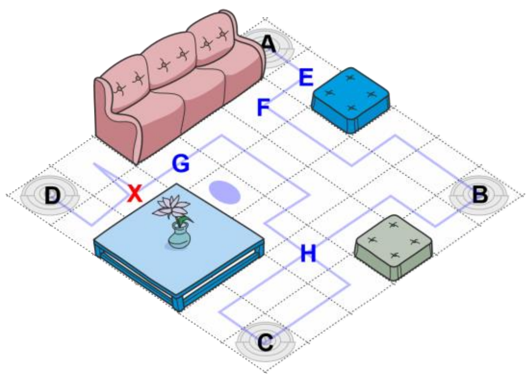
Pločice koje predstavalju "usko grlo" ka uglovima A, B, C i D opisane su slovima **E**, **F**, **G** i **H.** Pločice **E** i **F** su na prolazu do ugla A, pločica **G** na prolazu ka uglu D, a pločica **H** je na prolazu ka uglu C. Ugao B nema ovakvih pločica.

Ako pločica u uglu nije ni početna ni završna tačka, robot mora da pređe preko ovakve pločice dva puta. Ako robot krene ili završi na istoj pločici, pločice na prolazu ka tom uglu preći će samo jednom. Tako možemo pokušati da uredimo putanju da robot pređe preko nekoliko pločica na prolazima onoliko puta koliko je potrebno.

Ako robot krene i završi u istom uglu, on mora da prođe kroz prolaz bar jednom.

Pošto prolaz A sadrži **E** i **F**, dobra strategija je da se tačka A odabere za početnu (ili završnu). Onda treba da odlučimo između završnih tačaka C i D (u oba slučaja mi štedimo još jedan prelaz preko **G** i **H**).





Trag na slici sa leve strane pokazuje da je za pomeranje iz tačke A u tačku C potrebno 28 koraka napred (koji sadrži tačke **X** i **G)**. Za trag od A do D je potreban neparan broj koraka, tako da, posle 28 koraka napred, najmanje jedna pločica nije pređena (desna slika - pločica sa plavom elipsom nije još uvek pređena) tako da je putanja duža, nego od A do C.

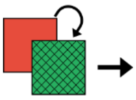
Najmanje potrebno vreme za čišćenje je 27 (za čišćenje) + 28 (za pomeranje) minuta = 55 minuta.

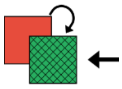
Informatička pozadina

Ovo je specijalan slučaj problema putujućeg trgovca - traženje najkraće putanje kada želimo da posetimo svaku tačku grafika ili proširen Hamiltonov problem putanje - bez obzira da li je u pitanju prelaženje preko svake tačke u grafiku. Nema tačnog algoritma da se ovo reši, ni za jedan od ova dva problema, tako da pronalaženje optimalnog rešenja za veću sobu bi zahtevalo mnogo više vremena za rešavanje, čak i sa moćnim računarima.

Robot

Milan je napravio robota koji čita obojene kvadrate, menja njihove boje i pomera kvadrat nalevo ili nadesno. Robot se ponaša poštujući pravila, kao što su ova ispod:

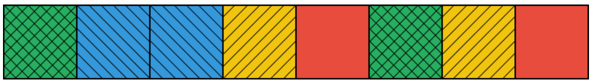
 Ako vidiš crveni kvadrat, promeni njegovu boju u zelenu i pomeri se jedan kvadrat nadesno.

 Ako vidiš crveni kvadrat, promeni njegovu boju u zelenu i pomeri se jedan kvadrat nalevo.

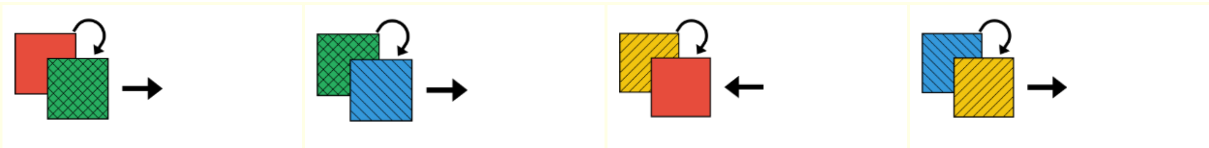
Na početku, robot stoji na prvom kvadratu sa leve strane. Prepoznaje boju kvadrata, pronalazi pravilo koje važi za ovu boju, menja boju kvadrata prema pravilu i pomera se prema pravilu. Zatim, robot ponavlja isti postupak za sledeći kvadrat na kome stoji, i tako redom.

Ako ne pronađe odgovarajuće pravilo ili izađe van obeleženih kvadrata, zaustavlja se.

Robotu je dat ovakav niz kvadrata



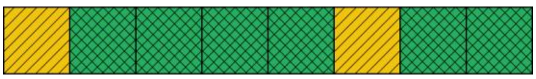
i ova pravila:

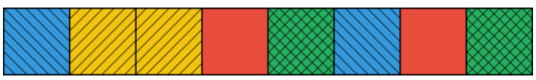


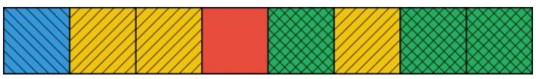
Pitanje:

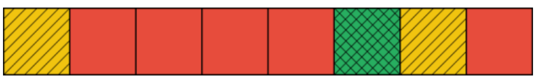
Kako će izgledati raspored kvadrata kada se robot zaustavi?

Odgovori:

a) 

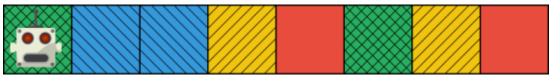
b) 

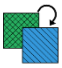
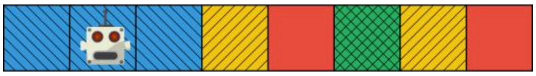
c) 

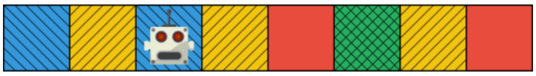
d) 

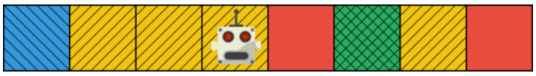
Tačan je odgovor je pod a).

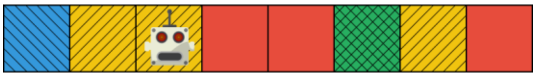
Evo kako je robot postupio:

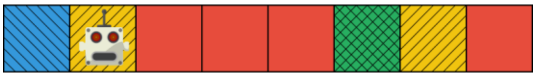
Početak: 

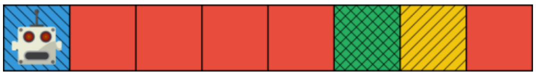
Prateći pravilo  → dobili smo 

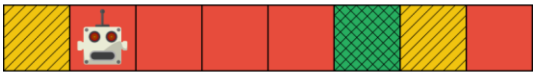
Prateći pravilo  → dobili smo 

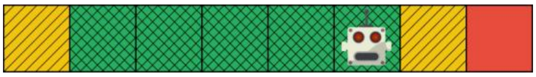
Prateći pravilo  → dobili smo 

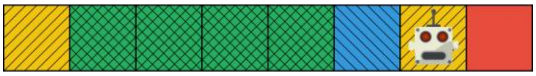
Prateći pravilo  ← dobili smo 

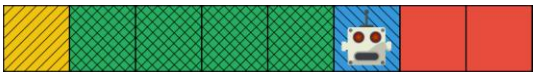
Prateći pravilo  ← dobili smo 

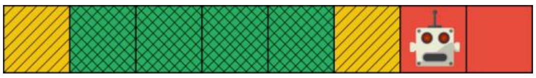
Prateći pravilo  ← dobili smo 

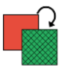
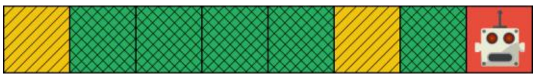
Prateći pravilo  → dobili smo 

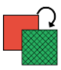
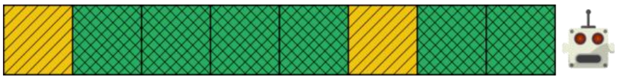
Prateći pravilo  → četiri puta dobili smo 

Prateći pravilo  → dobili smo 

Prateći pravilo  ← dobili smo 

Prateći pravilo  → dobili smo 

Prateći pravilo  → dobili smo 

Prateći pravilo  → dobili smo 

Sada je robot izvan kvadrata, zaustavlja se.

Informatička pozadina

U računarskim naukama je važno definisati model računanja za koji treba raditi. Model računanja je jednostavan skup pravila i struktura za usklađivanje. Tako, na primer, kada programiramo softver, naš model računanja je programski jezik koji koristimo.

Ovaj problem definiše model vrlo sličan, dobro poznatom modelu koji se zove Turingova mašina. Turingova mašina je veoma koristan model računarstva za računare, jer iako je veoma jednostavan, on je ekvivalentan većini programskih jezika, što znači da možemo pretvoriti bilo koji softverski program u Turingovu mašinu i obratno pretvoriti bilo koju Turingovu mašinu u program.

Tandem (ponavljanje nizova)

Svaki duži niz znakova, sa jednakim levim i desnim polovinama, naziva se tandem ili ponavljanje. Dabar Aleksa posmatra genetski niz svog prijatelja Vladimira. Za analizu ovog genetskog niza mora naći najduži tandemski deo. Pomozi mu da ga pronađe.

Pitanje:

Vladimirov genetski niz je **AACGTACGTACCTTACTAC.** Koje dužine je najduži tandemski deo?

Odgovor:

A. 2
B. 4
C. 6
D. 8

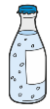
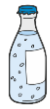
Tačan odgovor D. 8

Informatička pozadina

Traženje tandema je deo teorije žica. Koriste se u bioinformatici i analiziranju stanja različitih automatskih sistema.

Soda Shop

Četiri prijatelja su na putovanju, odlučuju da stanu i kupe piće u obližnjoj prodavnici. Tabela ispod prikazuje pića koja svako od njih više ili manje voli.

				
Anna				
Bernard				
Christine				
Daniel				

Prodavnica pića nudi četiri osvežavajuća bezalkoholna pića, međutim, ponestaje im zaliha. Imaju samo po jedno piće od svake vrste. Piće koje svaka osoba najviše voli prikazano je u tabeli najvećim brojem srca.

Primer: Anna najviše voli  i to je predstavljeno najvećim brojem srca , nešto manje voli , što je predstavljeno sa  i tako dalje, do pića koje najmanje voli.

Pitanje:

Koji je najveći ukupan broj srca koja grupa može da sakupi?

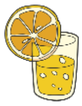
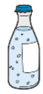
Odgovori:

A. 6
B. 8
C. 14
D. 16

Tačan odgovor je 14.

Obzirom na to da troje ljudi želi  , najveći ukupni broj srca koji grupa može dobiti je

4+4+3+3 = 14. Ovo se postiže sledećim uklapanjem narudžbi:

Anna 	Bernard 	Christine 	Daniel 

Drugi način da se ovo sagleda je sledeći: obzirom da Daniel bira  kao svoje omiljeno piće, a svi ostali su ovo piće rangirali kao poslednje, Daniel mora izabrati  da bi se postiglo najveće zadovoljstvo grupe.

Sada, prve tri osobe biraju isto piće koje najviše vole, ipak prva osoba ima  kao sledeće omiljeno piće, a ostalo dvoje imaju ovo piće kao treći izbor. Tako da, prva osoba treba da izabere  , a ostale dve mogu da izaberu  i  , kojim god redosledom.

Informatička pozadina

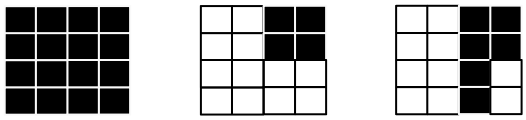
Optimizacija je veoma važan deo informatike. U ovoj situaciji pokušavamo da postignemo najveće zadovoljstvo grupe.

Problem optimizacije je sveprisutniji u informatici. Ovaj problem je tip problema sa uklapanjem. Ovde prijatelji pokušavaju da uklope najbolji mogući izbor pića, kako bi grupa bila zadovoljna.

Ovakvi problemi su veoma važni u stvarnom svetu. Npr. pacijenti koji čekaju na transplantaciju organa stavljaju se na dugačku listu. Međutim, ne mogu svi organi da se koriste za svakog pacijenta – krvna grupa, donor organa, moraju da se poklaope sa pacijentom, kako bi operacija bila uspešna.
Ova ograničenja otežavaju pronalaženje uspešnog poklapanja kojim bi se svi učinili zadovoljnim.

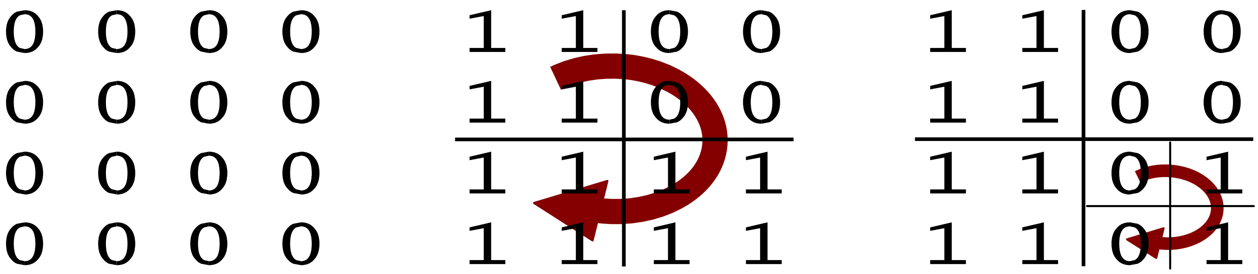
Kompresija ikonica

Pogledajte sledeće 4x4 crno-bele piksel slike:



One mogu biti sačuvane korišćenjem binarnih cifara: "1" za bele piksele i "0" za crne piksele.

Za čuvanje slike 4x4 trebalo bi 16 cifara. Sledeća metoda kompresije slike omogućava čuvanje slika korišćenjem manje cifara:



(0) (1011) (10(0110)1)

Binarne cifre su raspoređene u mrežu, poput piksela na slikama iznad.

Metod kompresije se primenjuje na mrežu tako da dobijamo sledeće rezultate:

1. Ako su sve cifre unutar mreže 0, rezultat je "0" (slika levo). Ako su sve cifre mreže 1, rezultat je "1".

2. U suprotnom, mreža se deli na četvrtine. Metod kompresije se primenjuje na svaku pod-mrežu, počevši od gornjeg levog ugla, u smeru kazaljke na satu. Rezultati su kombinovani i stavljaju se u posebne zagrade. Dva različita primera kompresije sa pod-mrežama prikazana su na srednjoj i desnoj slici.

Treba imati na umu da se pod-mreža može sastojati i samo od jedne cifre (vidi desnu sliku, donji desni ugao).

U nastavku pogledajte binarne cifre za sliku 8 × 8. Metoda kompresije se primenjuje na ovu mrežu.

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Pitanje:

Koji je rezultat kompresije?

Odgovori:

A) (1110)
B) (11(1011)1)
C) (111(1(1101)11))
D) (111(1(1011)11))

Tačan odgovor je D.

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

(111(1(<u>1011</u>)11))

<u>Informatička pozadina</u>

Kvadratna kompresija

Savršeni partneri

Dabrovi Andy, Bert, Chris, David i Eric su profesionalni balski plesači i učestvuju u TV emisiji. Amy, Brenda, Carol, Dianna i Emma su takmičarke koje će naučiti da plešu tokom ove emisije. Svakom profesionalnom plesaču biće dodeljena jedna takmičarka.

Pre početka emisije, producent je organizovao zabavu na kojoj su se svi upoznali.

Nakon zabave profesionalni plesači i takmičarke su popunili upitnik:

- svaki profesionalni plesač je rangirao takmičarku prema redosledu očekivane zajedničke uspešnosti
- svaka takmičarka je rangirala profesionalnog plesača prema očekivanoj brzini kojom bi mogla od njega da nauči ples

Evo rezultata ovih upitnika (najviši prioritet je 1, a najniži 5):

Prioriteti profesionalnih plesača

	Amy	Brenda	Carol	Dianna	Emma
Andy	1	3	2	5	4
Bert	1	2	3	4	5
Chris	2	1	4	5	3
David	5	4	3	2	1
Eric	4	5	2	3	1

Prioriteti takmičarki

	Amy	Brenda	Carol	Dianna	Emma
Andy	4	3	5	2	1
Bert	3	4	1	2	5
Chris	2	4	1	5	3
David	5	2	3	4	1
Eric	5	2	3	1	4

Producent želi da poveže profesionalne plesače i takmičarke, tako da nema ljubomore.

Pitanje:

Poveži profesionalne plesače sa takmičarkama, tako da svaki dabar ima svoju partnerku. Povedi računa da svako dvoje, koje ne odgovaraju jedno drugom, ne bi bili srećni ako bi bili povezani.

Odgovori:

Postoji samo jedno odgovarajuće rešenje, prema postavljenim parametrima.

a)

	Amy	Brenda	Carol	Dianna	Emma
Andy			x		
Bert	x				
Chris		x			
David					x
Eric				x	

b)

	Amy	Brenda	Carol	Dianna	Emma
Andy			x		
Bert				x	
Chris		x			
David					x
Eric	x				

c)

	Amy	Brenda	Carol	Dianna	Emma
Andy		x			
Bert	x				
Chris			x		
David					x
Eric				x	

d)

	Amy	Brenda	Carol	Dianna	Emma
Andy		x			
Bert	x				
Chris			x		
David				x	
Eric					x

Tačan je odgovor pod a).

	Amy	Brenda	Carol	Dianna	Emma
Andy			x		
Bert	x				
Chris		x			
David					x
Eric				x	

Gornje rešenje je nastalo na osnovu Gejl-Šejplovog algoritma:

<u>Prvi krug:</u>

- Andy traži Amy
- Bert traži Amy
- Chris traži Brendu --> privremeno Y
- David traži Emmu --> privremeno Y
- Eric traži Emmu

	Amy	Brenda	Carol	Dianna	Emma
Andy					
Bert					
Chris		x			
David					x
Eric					

<u>Drugi krug:</u>

- Andy traži Carol--> privremeno Y
- Bert traži Brendu
- Eric traži Carol

	Amy	Brenda	Carol	Dianna	Emma
Andy			x		
Bert					
Chris		x			
David					x
Eric					

Treći krug:

- Bert traži Carol
- Eric traži Diannu --> privremeno Y

	Amy	Brenda	Carol	Dianna	Emma
Andy			x		
Bert					
Chris		x			
David					x
Eric				x	

Četvrti krug:

- Bert traži Diannu

Peti krug:

- Bert traži Emmu

Šesti krug:

- Bert traži Amy --> privremeno Y

	Amy	Brenda	Carol	Dianna	Emma
Andy			x		
Bert	x				
Chris		x			
David					x
Eric				x	

Pošto ovaj konkretni zadatak treba rešiti samo za pet uparivanja, postoje i druge jednostavnije metode za dobijanje istog ovog odgovora.

Informatička pozadina

Ovo je važan problem u računarstvu. Iako nema mnogo problema koji zahtevaju ovo rešenje, mali broj njih je važan. Jedan primer gde se ovo rešenje primenjuje uključuje dodeljivanje lekara pripravnicima na njihovu omiljenu bolničku praksu. Veoma važan primer računarstvu je u pripisivanju korisnika serverima na Internetu.

Kao takva, postoje brojna istraživanja u ovoj oblasti, a najčešće se koristi algoritam Gale-Shaplei. Ovde možete pročitati više:

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