Dabar_2017/18.

Zadaci i rešenja za učenike starijih razreda osnovne škole
– školsko takmičenje
(novembar, 2017.)

Prijatelj takmičenja,

BIGZ školstvo

Sadržaj

***

Izbor zadataka za tekmičenje 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)
Suzana Miljković (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

Dabarčić

Zadatak	Težina	Broj bodova
Binarna vrata	Easy	6
Navodnjavanje	Easy	6
Levo, desno	Easy	6
Graditelji brana	Easy	6
Štap i štit	Medium	9
Kuglice	Medium	9
Cveće za Milicu	Medium	9
Ribice	Medium	9
Skakači	Hard	12
Peciva	Hard	12
Narukvice	Hard	12
Super Dabar	Hard	12

Mladi dabar

Zadatak	Težina	Broj bodova
Uljez	Easy	6
Štap i štit	Easy	6
Kuglice	Easy	6
Cveće za Milicu	Easy	6
Skakači	Medium	9
Ribice	Medium	9
Narukvice	Medium	9
Roštilj	Medium	9
Tuneli brane u Humskoj	Hard	12
Peciva	Hard	12
Kuglice	Hard	12
Sakupljanje bombona	Hard	12

Binarna vrata

Dabrovi su druželjubivi i vole da posećuju jedni druge. Ponekad nisu kod kuće. Zbog toga ostavljaju poruke svojim gostima, koristeći informacione kapije u ogradi, kao na slici ispod.

Dabrovi koriste 4 sledeće poruke:

Kod kuće smo. Molim vas uđite.	Vratićemo se u podne.	Vratićemo se uveče.	Vratićemo se u ponoć.

Mali dabar Pavle misli da je moguće postaviti više od 4 različite poruke, koristeći 3 grane, prema sledećim uslovima:

- Grane mogu biti postavljene horizontalno ili potpuno sklonjenje.
- Oblik i orijentacija grana nije važna.

Međutim, on ne zna koliko tačno poruka je moguće napraviti na ovaj način.

Pitanje:

Koji je najveći broj poruka koje se mogu napraviti na način koji predlaže dabar Pavle?

Odgovor:

A. 7
B. 8
C. 9
D. 10

Tačan odgovor je pod B: 8

<u>Informatičaka pozadina</u>

Ovaj zadatak se bavi binarnim brojevima i odnosi se na binarne sisteme, a istovremeno i na osnovnu kombinatoriku.

Učenici treba da budu izloženi matematičkim konceptima kao što su binarni broj, faktorijalni i slično, jer matematika je jedna od vitalnih komponenti računarskih nauka.

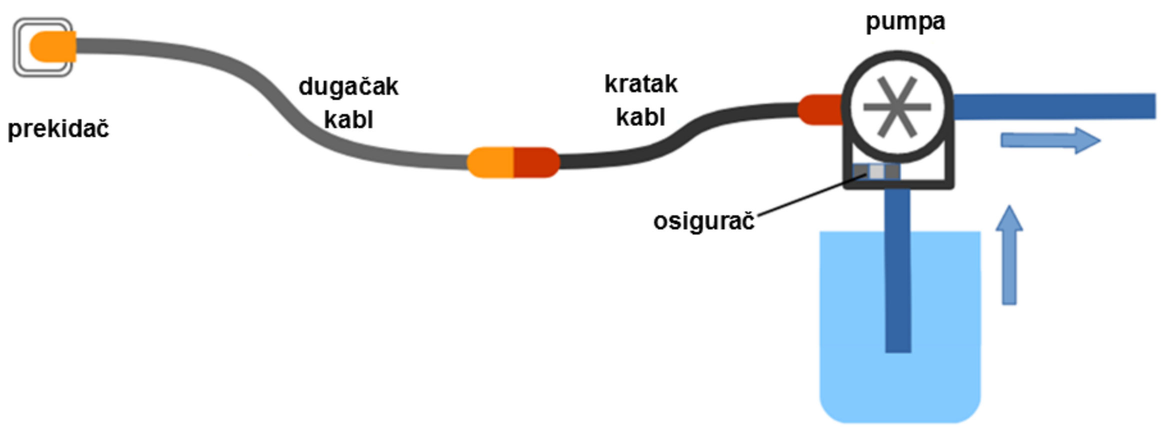
Iako ovaj zadatak može biti jednostavan za starije učenike, on može pomoći mlađim da počnu da razmišljaju o kombinatorici.

Navodnjavanje

Dabar Marko ima baštu cveća i povrtnjak. Za zalivanje je izgradio 2 ista sistema za navodnjavanje. Jedan od njih prikazan je na slici ispod.

Povezan je sa izvorom preko prekidača i sastoji se od:

- dugačkog kabla
- kratkog kabla
- pumpe
- pumpa sadrži osigurač (pumpa neće raditi ako osigurač svetli)



Jednog dana sistem za navodnjavanje bašte cveća je prestao da radi. Dabar je utvrdio da vodovodne cevi i rezervoar za vodu nisu problem. Za to vreme, svi delovi sistema za navodnjavanje u povrtnjaku rade ispravno i mogu se koristiti za testiranje.

Pitanje:

Postoji samo jedan neispravan deo i dabar Marko želi da ga pronađe. Nema očiglednog znaka, koji je deo neispravn. Dabar započinje testiranje. Koje od sledećih izjava su tačne?

1. Kada testira mora početi od pumpe, jer je to najvažniji deo.
2. Kada testira treba da počne od izvora napajanja i da proverava deo po deo dalje. Zamena jednog po jednog dela, pomaže pri pronalaženju neispravnog dela. Kada sistem proradi, poslednji zamenjeni deo je bio neispravan.
3. Kada testira treba ipak da počne od vodovodnih cevi i rezervoara za vodu, jer su najčešći kvarovi na ovim delovima.
4. Prilikom testiranja, treba menjati po dva dela istovremeno kako bi se pronašla greška, jer ovakav pristupak uvek dovodi do manjeg broja testiranja, nego zamena jednog po jednog dela.

Tačan odgovor je: 2

Odgovor 2 je tačan. Predstavlja ispravan način za pronalaženje pogrešnog dela, jer znamo da postoji samo jedan neispravan deo u sistemu.

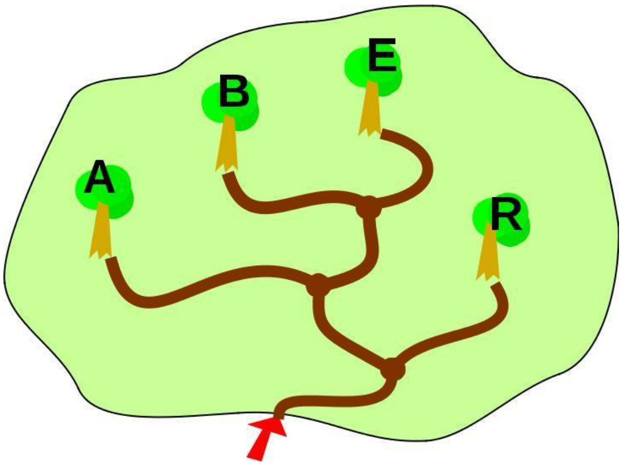
1) Značaj dela ne nagoveštava na moguće propuste.

3) Već je navedeno u zadatku da su ovi delovi provereni i ispravni.

4) Korišćenje ove strategije možda ne dovodi do manje ispitivanja nego zamena jednog dela istovremeno.

Otklanjanje grešaka je jedan od najvažnijih zadataka, jer kompleksni sistemi neizbežno imaju tendenciju da imaju greške. Pronalaženje jedne greške u programu je teško, ali pronalaženje 2 greške je još teže. Zato programeri imaju strategiju da rade na funkcionalnom kodu i pokreću testove nakon svake značajne promene. Dakle, oni se uveravaju da je do određenog dela sve u redu i da u novom delu moraju da traže jednu grešku.

Levo, desno



Dabrovi su osmislili kôd koji koriste za sledeću mapu:

- Svako drvo u parku je obeleženo slovom.
- Kôd za svako slovo dobija se tako što se beleže skretanja nalevo (L) i nadesno (D), na putu do drveta sa slovom.
- Kôd za svako slovo uvek počinje od obeleženog ulaza u park.

Primeri

Primer 1: Kôd za slovo **A** je LL, jer da biste od ulaza u park stigli do drveta sa slovom A morate dva puta da skrenete levo.

Primer 2: Kôd za reč **BAR** je LDLLLD

Pitanje

Koliko slova ima kôd za reč **BEAR**?

Odgovori:

A. 7
B. 8
C. 9
D. 10

Tačan odgovor je pod C: 9

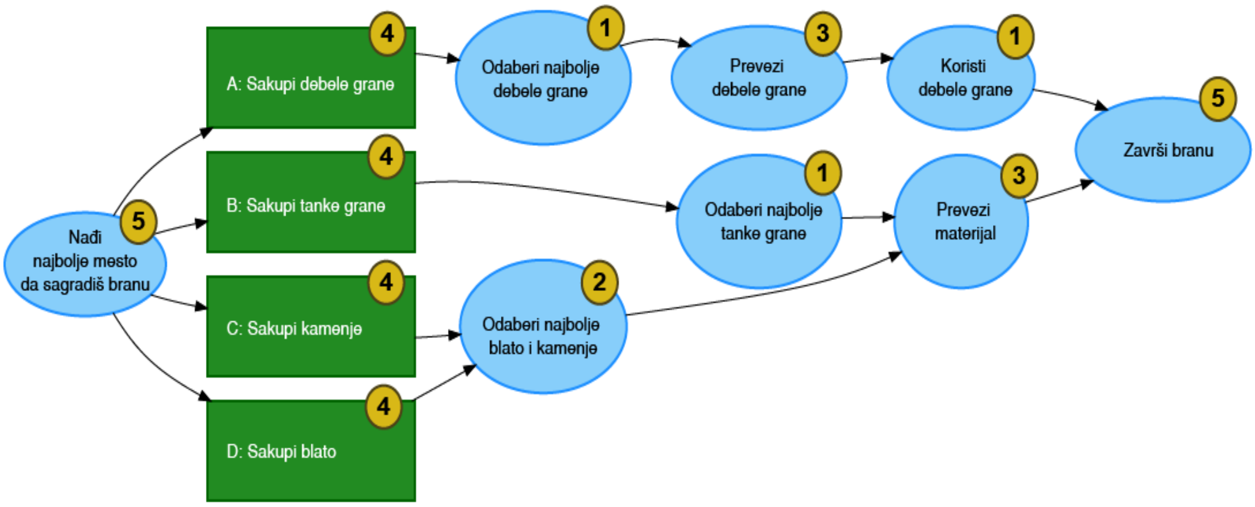
Ako računar zamenjuje slovo L u našem **levom / desnom kôdu** sa 0, a slovom R sa 1, **levi / desni kôd** postaje ono što se zove **binarni kôd**. Mapa parka postaje struktura kompjuterskih podataka zvana binarno drvo.

To znači da dugačaki i složeni podaci na lak način mogu da se skladište u kompjuteru i da pri tom zauzmu veoma malo prostora.

Zanimljiva stvar, u vezi sa ovim kôdom, je što nisu potrebne nikakve komade ili praznine. Pokušajte dekodirati odgovor kako biste videli da vam nisu potrebni razmaci da biste otkrili kada je kôd za svako slovo popunjen i počinje novo. Ova vrsta kôda se zove prefiksni kôd. To znači da je kôd još kraći.

Graditelji brana

Četiri dabra grade branu. Koriste plan sa slike ispod. Izgradnja brane je podeljena u više zadataka. Zadaci moraju biti rađeni po redosledu koji pokazuju strelice. Brojevi u žutim krugovima pokazuju koliko sati je potrebno za svaki zadatak. Jedan zadatak uvek radi samo jedan dabar.



Nakon što su dabrovi našli najbolje mesto za gradnju brane, jedan dabar je odlučio da se odmori četiri sata. Njegov zadatak će biti odložen za 4 sata.

Pitanje:

Koji od četiri zadataka bi trebalo odložiti, kako bi kašnjenje završetka brane bilo najmanje?

Odgovori:

A: Sakupi debele grane

B: Sakupi tanke grane

C: Sakupi kamenje

D: Sakupi blato

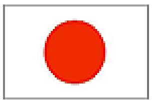
Tačan odgovor je B: Sakupi tanke grane

Tačan odgovor je B, “Sakupi tanke grane”. Odlaganje od 4 sata, kod zadataka A, C, ili D bi odložilo ceo projekat za 4 sata. Zadatak B, “Sakupi tanke grane”, ima uštedu vremena od 1 sat. Ako se ovaj zadatak odloži za 4 sata, konačno vreme da se završi projekat bi se produžilo za samo 3 sata.

Informatička pozadina

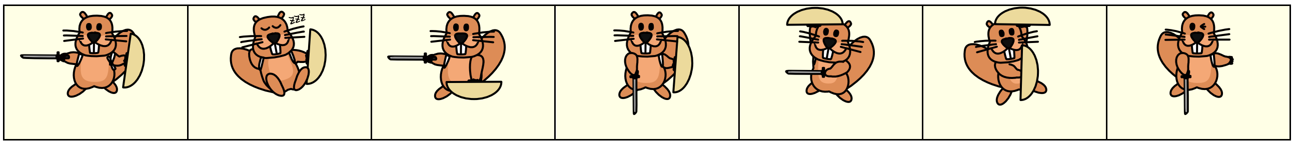
Izvršenje zadataka, uzimajući u obzir i njihove zavisnosti, tipičan je problem informatike.

Na primer, operativni sistem savremenih računara omogućava istovremeno pokretanje nekoliko programa.. Nasuprot tome, osnovni hardver često dozvoljava samo jednom programu izvršavanje jedne operacije. Zbog toga, operativni sistem sadrži složeni softverski modul, (scheduler) planer, koji odlučuje koja operacija treba izvršiti sledeće, s obzirom na ograničene raspoložive hardverske resurse i zavisnosti između zadataka.

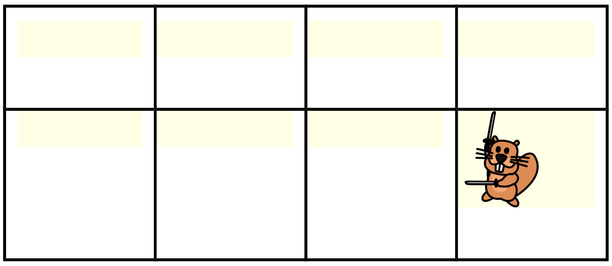


Štap i štit

Dabrica Suzana i 7 njenih prijatelja igraju igru „štap i štit“. Slika ispod prikazuje omiljenu pozu svakog Suzaninog prijatelja:



Dabrovi žele da imaju svoju sliku u školskom dvorištu. Na slici, svaki štap treba da pokazuje na drugog prijatelja, a svaki štit bi trebalo da blokira štap. Suzana je već zauzela svoje mesto, kao na slici ispod:

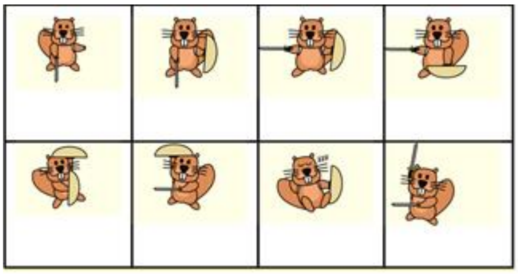


Pitanje:

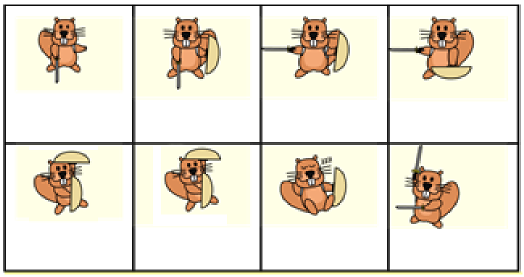
Koji je tačan raspored ostalih dabrova prema navedenim pravilima?

Odgovori:

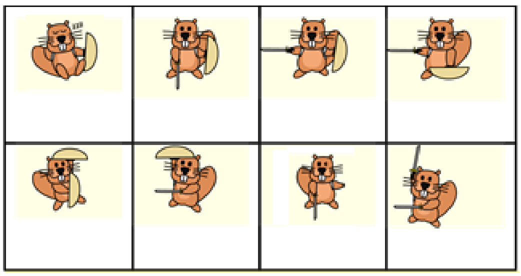
A



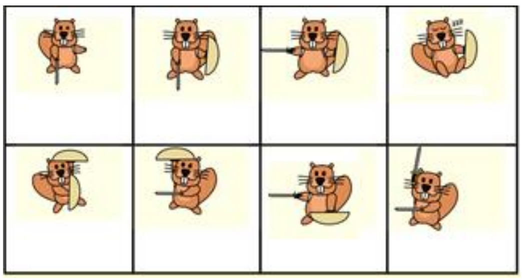
C



B



D



Tačan odgovor je pod A.

Informatička pozadina

Ovo bi zapravo mogla biti veoma složena slagalica. Samo nekoliko slika dovodi do veoma precizne pretrage među svim mogućim (ali netačnim) rešenjima. Ako dodate još jednu sliku slagalici od 6 komada, imali biste 6 puta više različitih mogućnosti postavljanja 7 kartica na prazne tačke. Za n kartice imate (n-1)! = 1 * 2 * 3 * ... * (n-2) * (n-1) različita moguća rešenja. Dakle, u ovom slučaju postoji 720 različitih mogućih rešenja (ali gotovo svi oni su pogrešni).

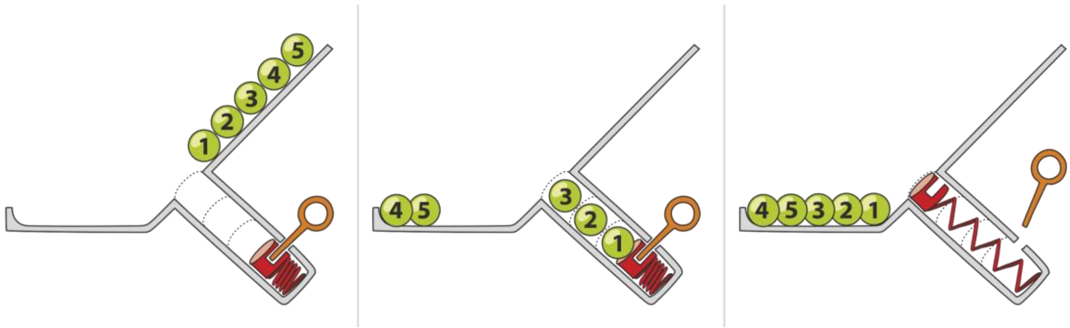
Međutim, koristeći neko logično razmišljanje, prostor za pretragu može puno biti umanjen. Na primer, svi dabrovi sa štapom koji se usmeravaju dole moraju biti postavljeni na gornji red, a postoji samo jedan dabar koji se može postaviti tačno iznad Lucije.

Potpuna iscrpna pretraga može se izvršiti koristeći algoritam koji se zove povratni udar. Prilikom korišćenja algoritma za povratni zadatak, prostor za pretragu može postati stvarno velik.

Kuglice

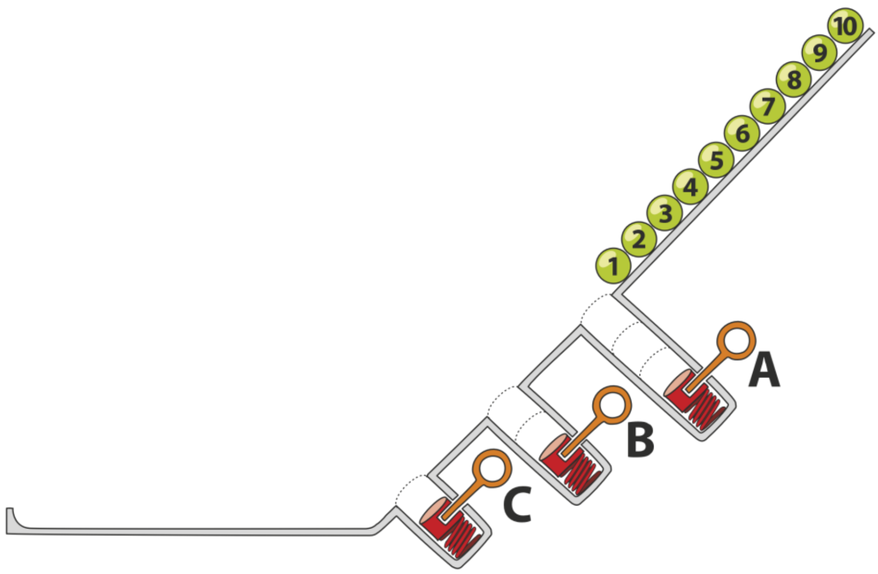
Numerisane kuglice spuštaju se niz rampu na kojoj se nalaze rupe. Kada kuglica dođe do rupe, ukoliko ima dovoljno mesta, kuglica upada u nju, u suprotnom, kuglica prelazi preko rupe.

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.

Cveće za Milicu

Marko želi da kupi buket od 3 cveta za svoju prijateljicu Milicu. Cvetovi u njegovom buketu se mogu razlikovati samo u jednoj od osobina.

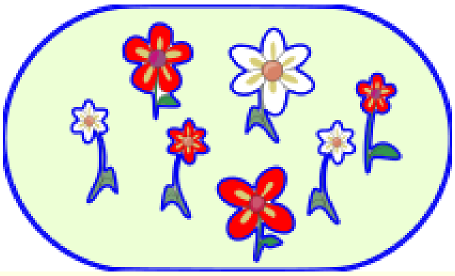
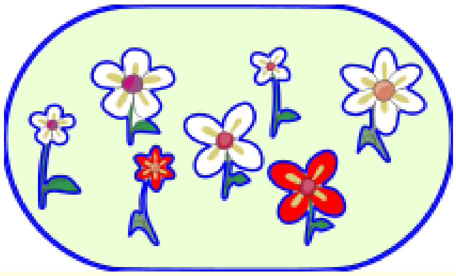
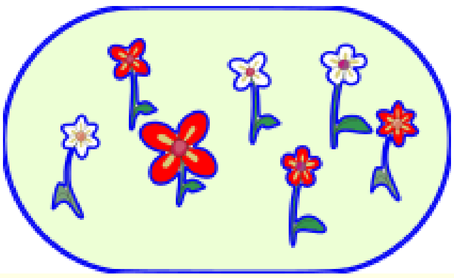
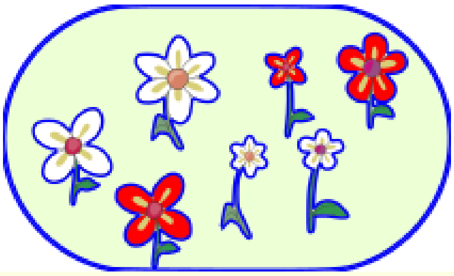
Svaki cvet ima tri osobine:

1. može imati 4, 5 ili 6 latica,
2. može biti bele ili crvene boje i
3. može biti veliki ili mali.

Pitanje:

U kojoj cvećari Marko **ne može** da kupi željeni buket?

Odgovori:

 A	 B
 C	 D

Tačan odgovor je: D

U radnji C Marko može da kupi 3 mala crvena ili 3 mala bela cveta, sa različitim brojem latica.

U radnji B Marko može kupiti 3 velika bela cveta, sa različitim brojem latica.

U radnji A Marko može kupiti 3 bela cveta sa 6 latica, različite veličine.

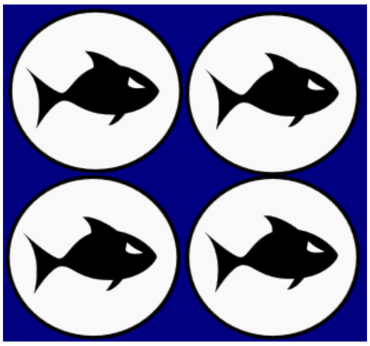
U radnji D Marko ne može kupiti 3 cveta iste boje - jer će se oni razlikovati u dve osobine, po veličini i broju latica.

Informatička pozadina

Filtriranje podataka je veoma koristan, brz i jednostavan način pronalaženja ili biranja objekata ili podataka sa željenim osobinama. U ovom primeru filtriramo cveće po boji, veličini i broju (latica).

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.

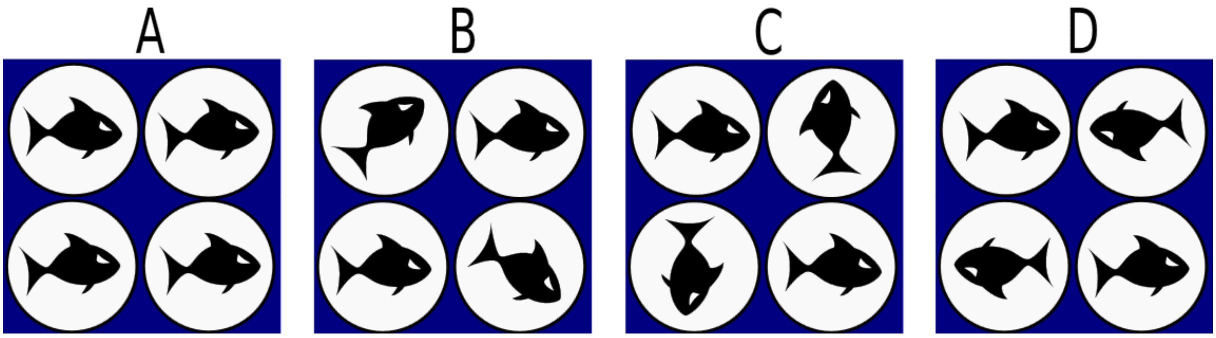
Uradi sledeće:

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:

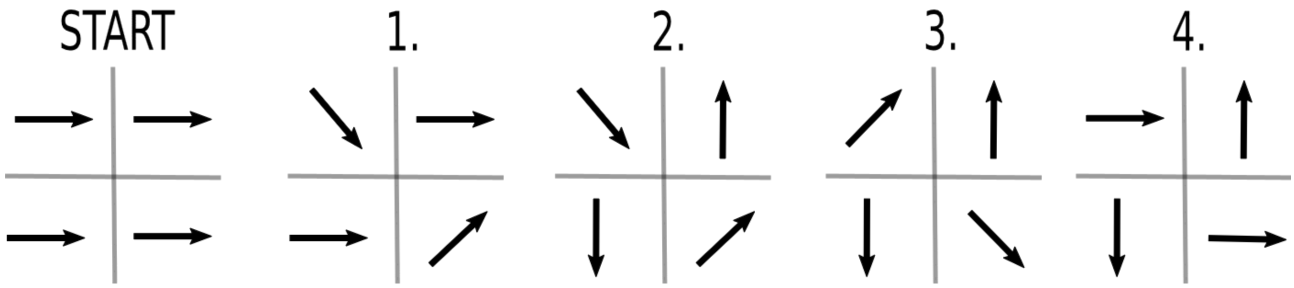
Koja slika prikazuje položaj ribica sada?

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.



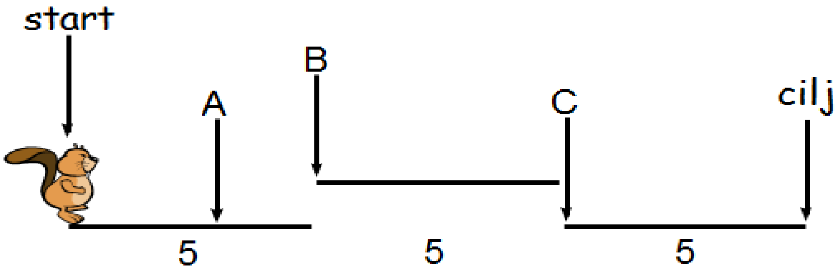
Na drugi način zadataka se može brzo rešiti čitanjem svih operacija i zapažanjem da samo korak br. 2 utiče na igračke na donjem levom i gornjem desnom uglu. Zbog toga, učenik može jednostavno pogledati ponuđene odgovore i odabrati jedan, sa donjom levom i gornjom desnom ribicom u ispravnom položaju.

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.

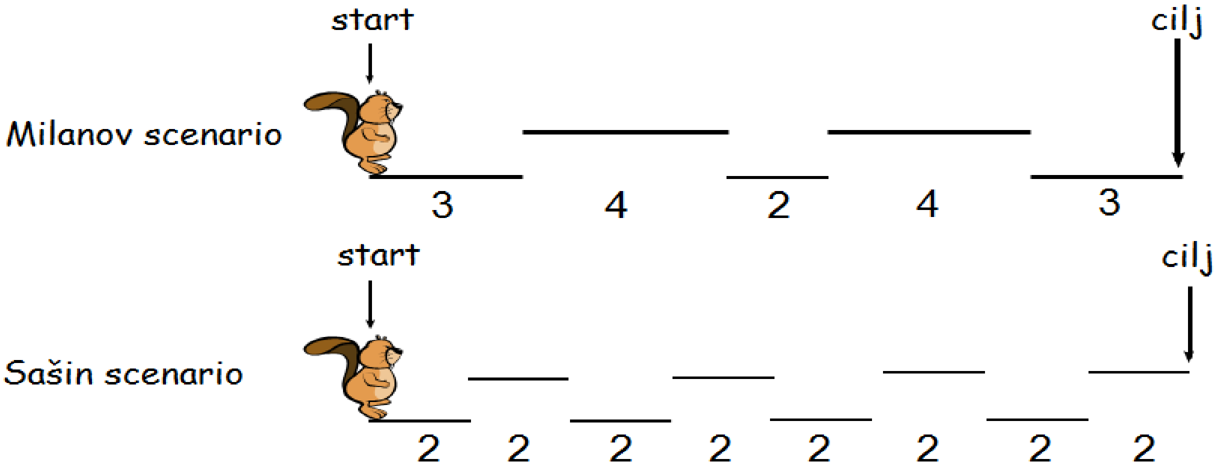
Na primer:



- 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

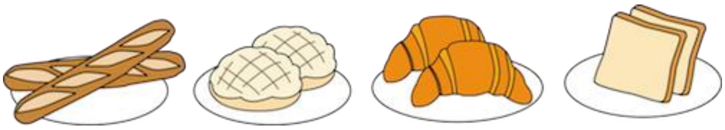
0011001100110011

0001001000110000

Informatička pozadina

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.

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)

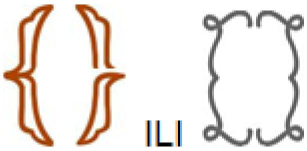
	Svetlana	Milan	Branko	Suzana
Hleb			Da (Milan)	
Krofna	Ne (Branko)			Da (Branko)
Kroasan	Ne (Svetlana/Suzana)	Da (Suzana)		
Tost				

Informatička pozadina

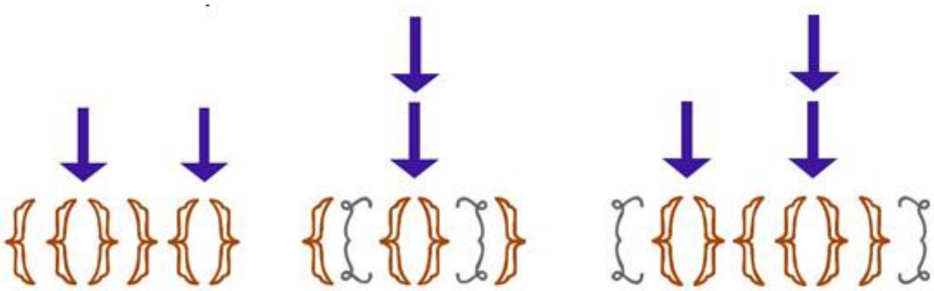
Matematička logika proučava osnovne principe matematičkih zaključaka. Bulova algebra je deo matematičke logike. Bazira se na principima deduktivnog logičkog zaključivanja, podaci o ulazu mogu imati samo dve faze: ispravno ili pogrešno. "Tabela istine" izražava odnose između dabrova i vrsta peciva.

Narukvice

Fabrika nakita proizvodi narukvice. Za izradu narukvica koriste ukrase u obliku nosača koji idu u parovima. Kada prave narukvicu počinju sa jednim od ovih parova:



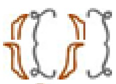
Dodatni parovi se više puta ubacuju na bilo koje mesto u narukvici, kao što se vidi u tri primera ispod:

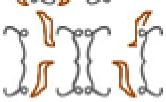


Pitanje:

Koja od sledećih narukvica je napravljena na opisani način?

Odgovori:

A. 

B. 

C. 

D. 

Tačan odgovor je pod D .

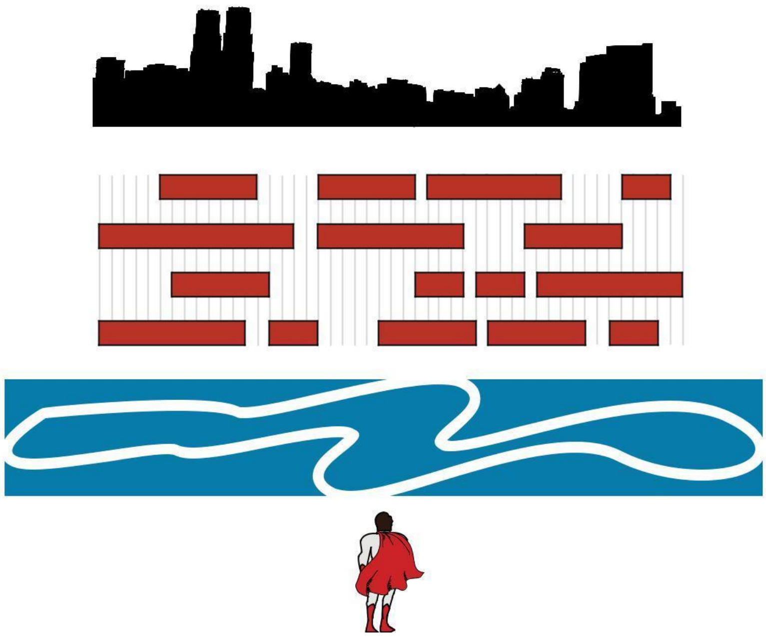
D je tačan odgovor. Počeli su sa dve zagrade, postavili par između druge dve zagrade, a zatim između još dve. Sve ostale narukvice nisu u skladu sa opisanom metodom.

<u>Informatičaka pozadina</u>

Pravila za izradu narukvica su upravo takva kao i pravila za postavljanje zagrada. Kompjuterski naučnik naziva korektne izraze "dobro formirani". Izrazi koji sadrže greške nazivaju se "malformisani". Izraz koji je dobro formiran, takođe se može nazvati "sintaksički korektan", što znači da se pokorava potrebnoj sintaksi.

Sintaksne greške mogu biti teške za pronalaženje u kompleksnim izrazima, ali je sintaksne greške obično mnogo lakše pronaći od "semantičkih grešaka", što su logičke greške napravljene od strane programera

Super dabar



Dok preko reke posmatra grad, Super dabar stoji na ravnoj stazi. Sa bilo kog mesta na stazi, dabar mora biti u stanju da vidi tačku u gradu, koja se nalazi tačno preko puta njega. Nažalost, između reke i grada stoji 16 zidova različite dužine.

Super dabar ima moć i može da vidi kroz zid. Međutim, jednim gledanjem može da vidi samo kroz jedan zid.

Super dabar je dovoljno jak da može da uništava zidove. Nažalost, uništavanje zidova Super dabra čini umornim.

Pitanje:

Koji je najmanji broj zidova koje Super dabar treba da uništi, da bi preko reke video tačku u gradu?

Odgovori:

A. 9
B. 10
C. 11
D. 12

Tačan odgovor je pod A: 9

Opšte rešenje za ovaj problem je pohlepni algoritam. To jest, to je sekvenca koja uvek rešava problem izbora, u svakom koraku vrši izbor koji se čini najboljim "lokalno" ili "trenutno". Ne uzima u obzir "širu sliku" ili puni skup informacija.

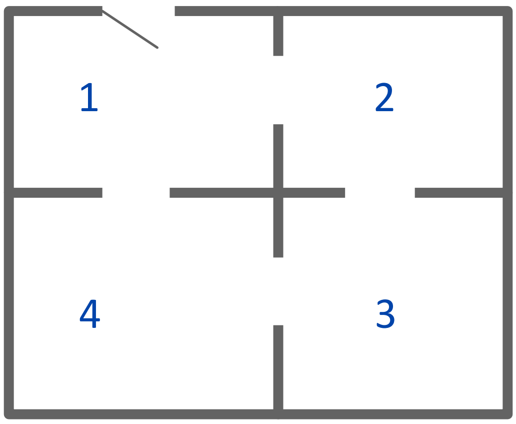
Uljez

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.

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, ...

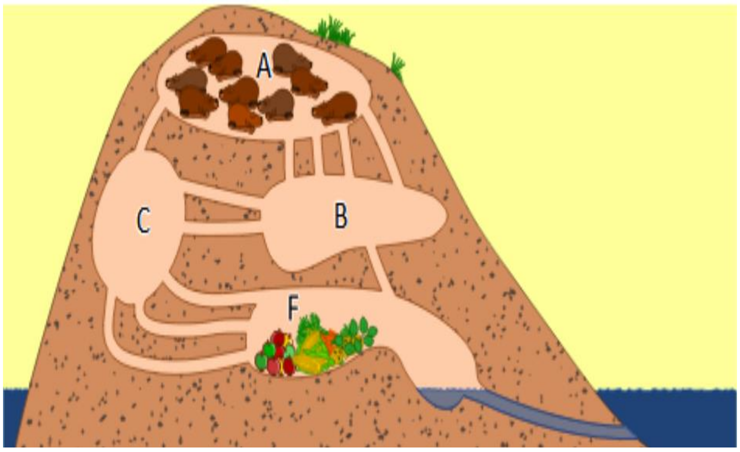
Postoji nekoliko efikasnih sortirnih algoritama, često ilustrujući standardne algoritamske strategije dizajna.

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.

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:

$$v(i,0) = 0$$

$$v(0,j) = 0$$

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

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.