

Državni nivo takmičenja



Niš – Ekonomska škola – Niš
Novi Sad – Elektrotehnička škola “Mihajlo Pupin”
Kruševac – Gimnazija Kruševac
Beograd – Univerzitet Singidunum

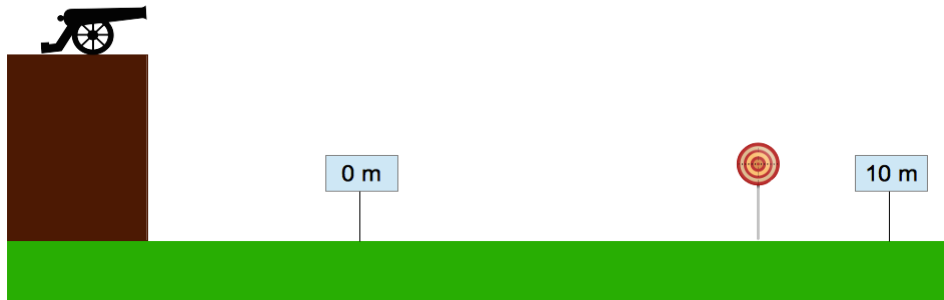
20.02.2016.god.

Državni nivo takmičenja	1
Topovsko đule	3
Put nafte	4
Biciklistička trka	5
Pečurke	6
Sistem za navodnjavanje	7
KUPOVINA U „SVETU CIPELA“	8
Drveće u šumi	9
Bezbedna lozinka	10
Mašina za računanje	11
Turistička agencija	12
Obavezno skretanje	13
Sakupljanje žirova	14
Klavir	15
Ko će ići na izlet?	16
Igra dugmićima	17
Špijuni	18
Zdrav ručak	19
Dabar Alhemičar	20
BD kod	21
Pravljenje čipa	22
Sastanak	23
Donošenje vode	24

Topovsko đule

Dabar Krca pokušava topovskom paljbom da pogodi cilj. Top se može podesiti tako da ispaljuje đule u rasponu od 0 do 10 metara.

Položaj mete je nepoznat, ali posle svakog ispaljivanja, njegov prijatelj Saša mu kaže da li je topovsko đule palo ispred ili iza cilja.



Pitanje

S obzirom na to da cilj ima širinu od 50 cm, koji je minimalan broj topovskih ispaljivanja koje će Krca ispaliti da bi bili sigurni da će pogodi metu, bez obzira na to gde se ona nalazi u rasponu od 0 do 10 metara?

- A. 3
- B. 4
- C. 5
- D. 6

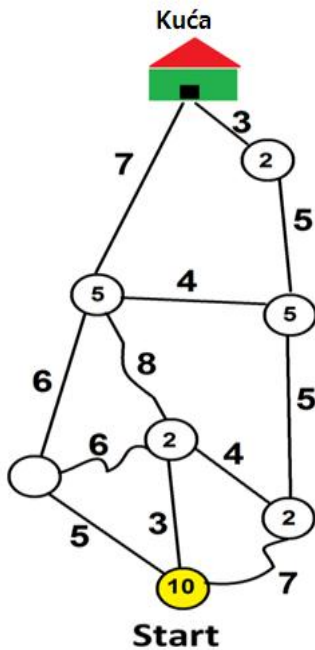
Odgovor: c.5

Informatička pozadina: U računarstvu binarna pretraga je algoritam za pronalaženje položaja neke stavke u poređanom nizu. Binarna pretraga je dihotomni, „podeli pa vladaj“ pretraživački algoritam.

Put nafte

Dabar Bane želi da stigne kući iz fabrike u kojoj radi (početna tačka). Nažalost, ima dovoljno goriva za 10 kilometara.

Na putu do kuće Bane će prolaziti kroz neke gradove. U tim gradovima postoje benzinske pumpe koje mogu da toče gorivo sa kojim se može proći određen broj kilometara (prikazano u krugu).



Pitanje:

Koju putanju će izabrati dabar Bane ?

7-5-5-3

3-4-5-4-7

3-8-4-5-3

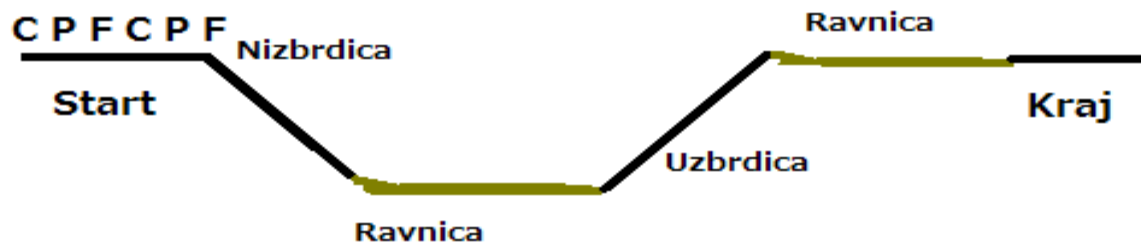
3-4-5-5-3

Tačan odgovor: c

Informatička pozadina: Ovo je klasičan problem grafova.

Biciklistička trka

U ovogodišnjoj biciklističkoj trci vozači različitih timova imaju različite prednosti. Kada se vozi nizbrdo, vozači **plavog tima** će preteći jednog vozača ispred njega. Kada se vozi uzbrdo, vozači **crvenog tima** su brži i svaki vozač crvenog tima će preteći vozača ispred njega. Na ravnom delu puta članovi **fluorescentnog tima** će proći vozača ispred njega/nje.



Pitanje: Ako su vozači poređani kao na slici C, P, F, C, P, F i prvo preticanje će biti na nizbrdici, kakav će redosled biti na cilju?

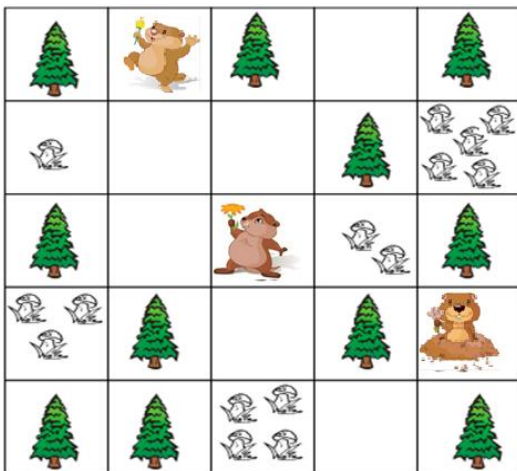
1. P, F, C, P, F, C
2. P, C, P, C, F, F
3. P, C, P, F, C, F
4. P, C, F, P, F, C

Tačan odgovor: 3

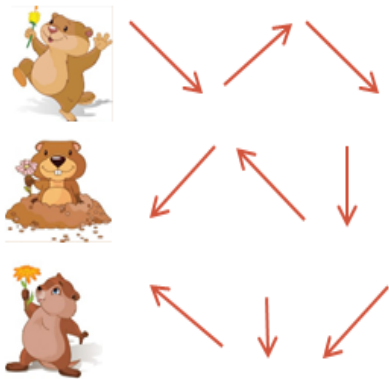
Informatička pozadina: Organizovanje podataka prema određenim i unapred utvrđenim pravilima je vrlo važan segment u informatici. Ovaj zadatak je primer kako se različitim filterima (nizbrdica, ravnica, uzbrdica) dobija određena organizacija podataka.

Pečurke

Dabrovi su pozicionirani kao na slici ispod:



Oni sakupljaju pečurke i kreću se na sledeći način:



Pitanje: Koliko je pečuraka sakupio svaki od dabrova ako su poređani kao na slici ispod?

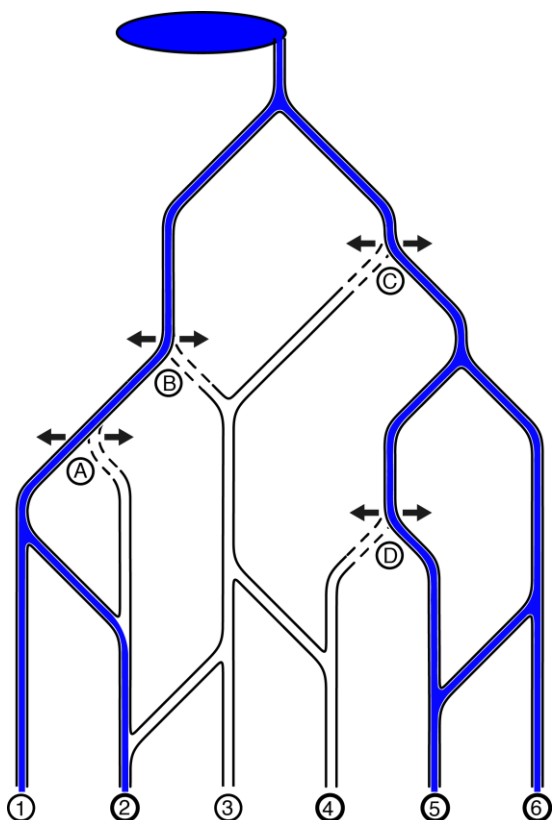


- a. 2, 4, 1
- b. 2, 1, 4
- c. 5, 4, 1
- d. 3, 4, 5

Tačan odgovor **D**: 3, 4, 5

Informatička pozadina: Algoritam je opis za rešavanje nekog problema. Jednostavni grafički algoritmi su važna osnova za početnike.

Sistem za navodnjavanje



Dabrovi su napravili sistem za navodnjavanje svojih polja useva obeleženih brojevima 1,2,3,4,5,6. Voda teče iz jezera na vrhu brda sve do polja u podnožju.

Duž kanala za navodnjavanje dabrovi su napravili 4 vodene kapije A,B,C,D koje usmeravaju vodu levo ili desno.

Pitanje: Kako treba podesiti vodene kapije da bi se navodnjavala samo polja 2,4,5 i 6?

A)	A: ←	B: ←	C: →	D: ←
B)	A: →	B: ←	C: ←	D: →
C)	A: →	B: ←	C: →	D: ←
D)	A: ←	B: →	C: →	D: →

Tačan odgovor C

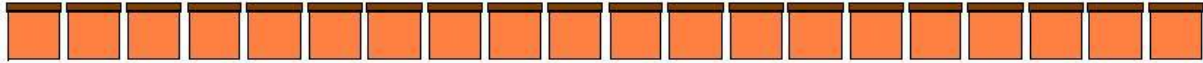
Informatička pozadina

U osnovi sistem za navodnjavanje se ponaša kao direktni grafikon u teoriji grafova. Oblik grafika je sličan drvetu sa korenom, granama i listovima.

KUPOVINA U „SVETU CPELA“

Dabar Sebastijan se nalazi u „Svetu cipela“. On želi da kupi novi par cipela. Posle dužeg traženja konačno je pronašao cipele koje mu se sviđaju. Svaki model cipela se nalazi u 20 kutija. Nažalost, veličina cipela nije napisana na kutijama, već samo na cipelama.

Kutije su poredane na osnovu rastućeg broja cipela. U jednom trenutku može da otvori samo jednu kutiju.



Pitanje:

Koliko najmanje mora da otvori kutija da bi sigurno pronašao željeni broj cipela?

- a) 4
- b) 5
- c) 10
- d) 20

Odgovor:

Tačan odgovor b).

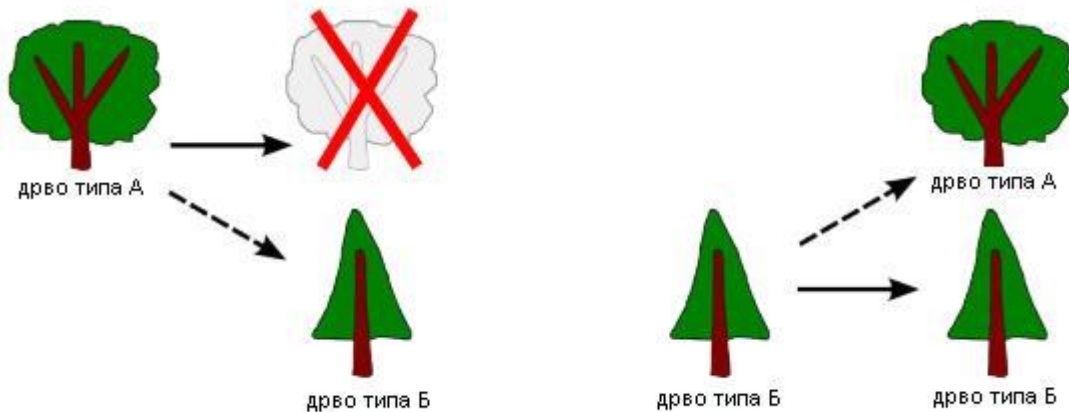
Prvo bi se proverila 10. kutija, ako nije ta, onda nam ostaje 10 preostalih kutija za drugu pretragu. Uvek bi se birala središnja, posle druge provere bi ostalo 5, posle treće dve, zatim posle četvrte bi ostala još jedna kutija i na kraju iz petog otvaranja sigurno bi se našao željeni broj cipela.

Informatička pozadina

U programiranju je važno vreme izvršenja programa. Na osnovu neke maksimalne i minimalne vrednosti treba poboljšati performanse programa. U ovom zadatku je vršen algoritam binarne pretrage.

Drveće u šumi

U šumi postoje dve posebne vrste drveća koje rastu. Drveće tipa A živi samo jednu godinu (počevši od semena), na kraju godine ono nestaje i pušta seme za drvo tipa B. Drveće tipa B živi zauvek, a na kraju svake godine pušta seme za drvo tipa A.



Pitanje:

Ako počnemo sa samo jednim semenom stabla tipa A, koliko ćemo stabala tipa A i tipa B imati nakon 10 godina?

- a) 34 A , 20 B
- b) 54 A , 144 B
- c) 34 A , 55 B
- d) 121 A , 55 B

Odgovor:

Tačan odgovor c).

Objašnjenje

Postoje dva važna zapažanja:

-Broj drveća tipa A u datoj godini je broj stabala B u prethodnoj godini.

-Broj stabala B u datoj godini je broj stabala B u prethodnoj godini, plus broj A stabala u prethodnoj godini.

Informatička pozadina

Ovde je reč o rekurzivnoj formuli, primenjuje se ista formula, ali se menjaju elementi koji učestvuju. U informatici to je princip dinamičkog programiranja (primenjuje se ista sekvenca, samo se menjaju podaci koji učestvuju u toj sekvenci).

Bezbedna lozinka

Tvoja drugarica želi da otvori nalog na socijalnoj mreži Beaverbook i moli da joj pomogneš oko izbora dobre lozinke.

Pitanje: Koji od ponuđenih saveta za izbor lozinke je najbolji?

- (a) Koristi tvoje ime iza koga sledi godina rođenja.
- (b) Iskoristi istu lozinku kao za tvoj i-mejl nalog.
- (c) Kreiraj lozinku od 8 proizvoljno izabranih malih i velikih slova engleskog alfabeta ili cifara
- (d) Uzmi rečnik sa bar 8000 reči i iz njega izaberi proizvoljno 4 reči.

Odgovor (d) je tačan.

Objašnjenje: Lozinku - odgovor (a) mogu lako pogoditi ljudi koji znaju tvoje prijatelje, kao i neka osoba čije je ime slično tvom a za starost postoji mali broj mogućnosti.

Lozinka - odgovor (b) može biti sigurnija, ali upotreba iste lozinke sa različitim naložima povećava rizik otkrivanja; tada će svi ti nalozi biti ugroženi.

Odgovor (c) izgleda kao najbolja opcija, ali broj mogućih lozinki je manji nego kod (d): engleski alphabet ima 26 slova i 10 cifara, tako da je broj mogućnosti za 1 karakter $26 + 26 + 10 = 62$, a za lozinku od 8 karaktera ima $62^8 = 218.340.105.584.896$ mogućnosti.

Odgovor (d) je stvarno najbolja opcija, jer $8000^4 = 4.096.000.000.000.000$. S druge strane, lozinka kreirana na ovaj način biće mnogo lakša za pamćenje.

Informatička pozadina: Nalozi se koriste u mnogo aplikacija i zbog toga je bezbednost vrlo važna.

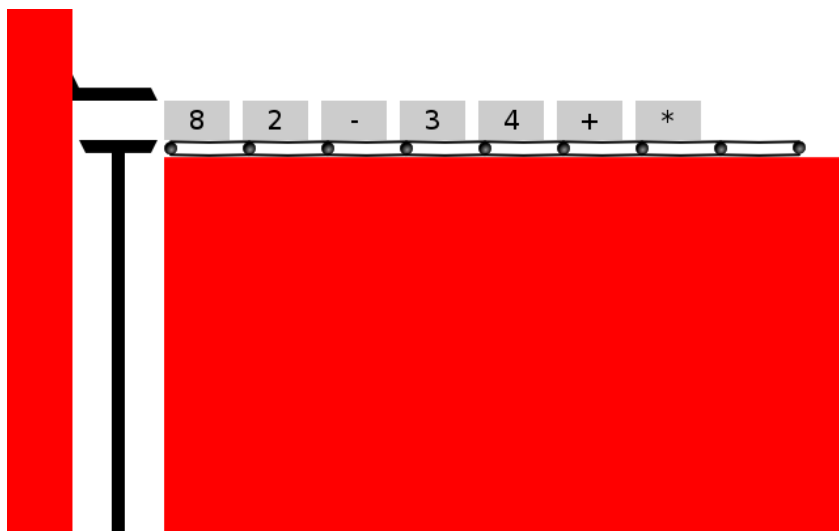
Lozinku bi trebalo birati tako da je laka za pamćenje (od strane korisnika), a teška za pogađanje (od strane računara).

Glavna mera jačine lozinke je broj raznih mogućnosti, odnosno broj bitova potrebnih za njeno čuvanje. Za (c) je $62 \approx 2^6$ pa imamo $8 \times 6 = 48$ bitova, dok je kod (d) $8000 \approx 2^{13}$, što znači da je broj bitova potrebnih za čuvanje $4 \times 13 = 52$.

Mašina za računanje

Evo jedne jednostavne mašine za računanje. Ona slaže kutije koje dolaze sdesna jednu na drugu. Kad naiđe kutija sa operacijskim simbolom (+, -, *, ili /) onda se tri kutije koje su na vrhu zamenjuju kutijom sa rezultatom računanja te operacije. Da bi koristili mašinu izraz moramo zadati drugačije. Na primer:

- $2+3$ se zadaje kao 2 3 +
- $10-2$ se zadaje kao 10 2 -
- $5*2+3$ se zadaje kao 5 2 * 3 +
- $5+2*3$ se zadaje kao 5 2 3 * +
- $(8-2)*(3+4)$ se zadaje kao 8 2 - 3 4 + *



Pitanje: Kako zadati mašini da izračuna vrednost izraza $4*(8+3)-2$?

- (A) 4 8 3 2 * + -
- (B) 4 8 3 * + 2 -
- (C) 4 * 8 3 + 2 -
- (D) 4 8 3 + * 2 -

Izraz (D) je tačan odgovor.

Prisetimo da sledeći zapisi daju isti rezultat:

- $4\ 3\ 8\ +\ *\ 2\ -$
- $8\ 3\ +\ 4\ *\ 2\ -$
- $3\ 8\ +\ 4\ *\ 2\ -$

Međutim, redosled operanada i operacija nije isti kao u datom izrazu.

S leva na desno, prvo moramo da odredimo $4*(8+3)$, tako da 4 i rezultat $8+3$ stavljamo na stek. To postizemo zapisom 4 8 3 + . Onda imamo 4 i 11 na steku, tako da dopisujemo * za množenje ta dva broja. Sada imamo 44 na steku, dodajemo na stek 2 i znak - za završno oduzimanje.

Informatička pozadina: Uobičajena način zapisivanja aritmetičkih izraza nije najjednostavniji računaru za *razumevanje*, odnosno, za obradu tako zapisanih izraza je mnogo komplikovanije napisati program. Međutim, pisanje programa koji koriste postfiksnu notaciju (kao mašina iz zadatka) je mnogo, mnogo lakše. Zato su neki prvi kalkulatori koristili ovu notaciju. Takođe, postfiksna notacija ne zahteva zagrade, bez obzira na kompleksnost izraza.

Turistička agencija

Dabar radi u turističkoj agenciji. Turistička agencija ima u ponudi sledeća putovanja:

Vrsta putovanja	Država	Vrsta smeštaja	Transport	Hrana je uključena
Poslovno putovanje	Španija	Hotel	Avion	Da
Vikend putovanje	Kanada	Hostel	Bus	Da
Istraživačko putovanje	Malezija	Hotel	Bus	Da
Medeni mesec	Južna Afrika	Hostel	Avion	Ne
Poslovno putovanje	Španija	Hotel	Avion	Ne
Poslovno putovanje	Španija	Kuća	Avion	Da
Istraživačko putovanje	Malezija	Hotel	Bus	Ne
Medeni mesec	Južna Afrika	Hotel	Bus	Da
Vikend putovanje	Kanada	Kuća	Avion	Ne
Vikend putovanje	Kanada	Hotel	Bus	Da

Šta Dabar treba da pita mušteriju, da bi sa sigurnošću znao koje putovanje treba da mu ponudi:

“Vrsta putovanja” i “Država”

“Vrsta putovanja”, “Vrsta smeštaja” i “Transport”

“Država”, “Vrsta smeštaja” i “Hrana je uključena”

“Vrsta smeštaja” i “Hrana je uključena”

Odgovor: **3**

Informatička pozadina: U informatici se baze podataka koriste za efikasno upravljanje velikim količinama podataka. Da bi se ubrzalo pronalaženje podataka neophodan je primarni ključ. Primarni jedinstveno identifikuje svaku vrstu u tabeli. U SQL standardu, primarni ključ može da se sastoji od jedne ili više kolona. Svaka kolona koja učestvuje u primarnom ključu implicitno je definisana kao NOT NULL.

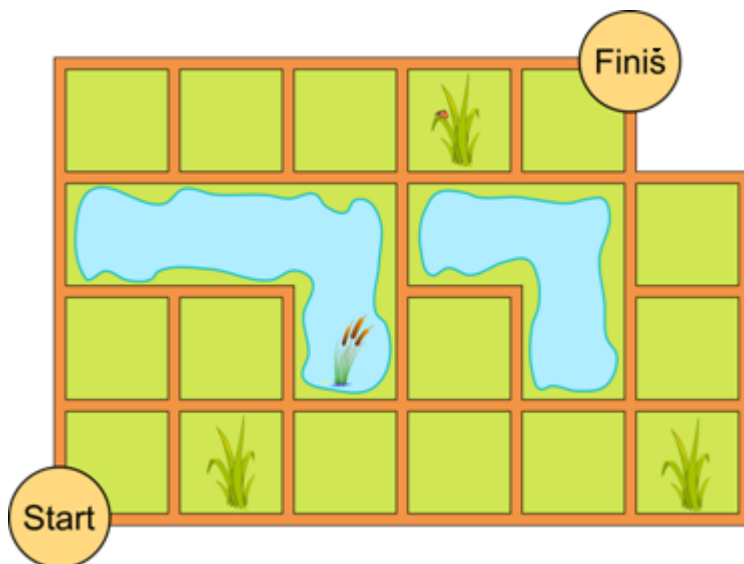
Obavezno skretanje

Kralj voli duge putovanja kočijama, tako da je rekao njegovom kočijašu pravilo da nikada ne idu pravo kada naiđu na raskrsnicu. Tako, kočijaš mora da skrene levo ili desno na raskrsnici.

Na slici vidite mapu. Svi putevi, koji povezuju dva susedne raskrsnice imaju istu dužinu od 1 kilometra.

Kralj mora da ide iz donjeg levog ugla (krug Start) do gornjeg desnog ugla (krug Finiš): obe lokacije su označene krugovima u kojima piše **Start** i **Finiš**. Međutim, kočijaš želi da stigne do finiša što je brže moguće ...

Pronađite kočijašu najkraći put od početka, držeći se kraljevog pravila. Kolika je dužina najkraćeg puta?



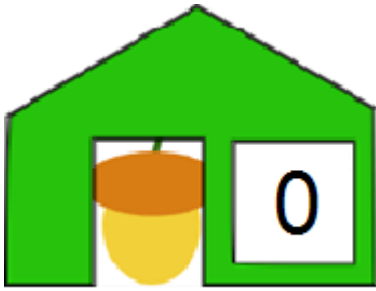
- A. 11 km
- B. 13 km
- C. 15 km
- D. Nemoguće je doći do kraja bez kršenja pravila.

Odgovor: B

Informatička pozadina: u pitanju je problem pretrage optimalnog puta. Obično je potrebno da se nađe put, minimalne udaljenosti, od jedne do druge tačke.

Sakupljanje žirova

Veverice prikupljaju žirove u jedno, zajedničko skladište. Takođe, žele i da prate brojno stanje u skladištu.



Veverice veoma sporo računaju, tako da moraju da odu do svoje kuće kako bi izvršile izračunavanja. Evo šta se dešava kada veverica odnese žir do skladišta:

- ostavlja žir u magacin
- **P**ročita broj ispred vrata
- ode do kuće i sabere broj koji je videla ispred vrata sa brojem 1
- vrati se u magacin, obriše broj koji vidi i **Z**apiše broj koji je izračunala kući

U početku, skladište je bilo prazno, pa je broj pored vrata bio 0.

Posle toga je deset veverica donelo po jedan žir, otišlo do kuće da sabere i vratila se, i to po sledećem redosledu:

P1 P2 Z1 P3 P4 Z2 Z3 P5 Z4 P6 Z5 P7 P8 P9 P10 Z6 Z7 Z8 Z9 Z10

Na primer, **P4** znači da je četvrta veverica donela žir i pročitala broj pored vrata a **Z5** znači da se peta veverica vratila od kuće sa izračunatim brojem i da ga je zalepila pored vrata.

Pitanje: koji će broj biti ispisan na kraju?

- A. 2
- B. 7
- C. 4
- D. 10

Odgovor: C, 4

Informatička pozadina: Opisana situacija je tipična kada je nekoliko procesora (veverice) rade sa istim podacima (broj pored vrata). Ključ je činjenica da se koraci (čitanje podataka, obrada i rezultat) mogu preplitati, a zatim krajnji rezultat može biti teško predvidiv i zapravo zavisi od reda preplitanja. U informatici se ovo naziva **race condition (hazard)** i postoje posebne tehnike njegovog izbegavanja.

Klavir

Dabar Taske uči da svira jednostavnu pesmu na svom klaviru. Pošto nije neki iskusni muzičar, on je pisao brojeve na dirkama klavira da ne bi zaboravio redosled pritiskanja dirki.

Dve susedne bele dirke klavira su na udaljenosti od jednog "tona". Ako nema crne dirke između dve bele dirke, onda su one na udaljenosti od samo pola tona. A "ton" je jednak dva "polutona".

Dve note bilo kojoj udaljenosti čine "Interval".

Postoje sledeći intervali (pt = pola tona):

T1 = 0 pt

K2 = 1 pt

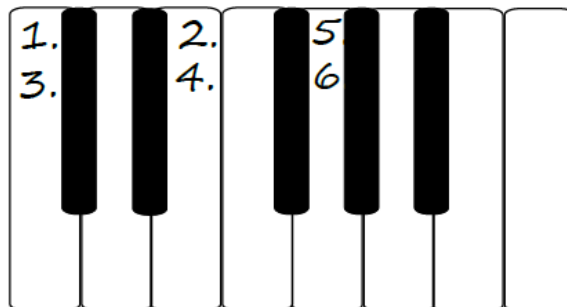
N2 = 2 pt

K3 = 3 pt

N3 = 4 pt

T4 = 5 pt

T5 = 7 pt



Koristeći gore navedene savete, odgovorite koju pesmu svira Taske:

(prvi interval čine prva i druga nota, drugi interval čine druga i treća nota itd...)

A) N2, N2, N2, N2, K2

B) N3, N3, N3, K3, T1

C) T4, T4, T4, N3, T1

D) N3, N3, N3, K3, K2

Odgovor: B

Informatička pozadina: Programeri koriste dosta algoritama, što im u mnogome olakšava rešavanje različitih zadataka. Jednostavno, algoritmi pomažu programerima. Algoritam je opis za rešavanje nekog problema. Algoritam predstavlja niz dobro definisanih pravila, kojima se ulazne vrednosti transformišu u izlazne, ili se opisuje izvršavanje nekog postupka.

Primer iznad uči nas korišćenju algoritma i jedinstvenog programskog jezika. U ovom slučaju programski jezik bi bio u stanju stanju da reprodukuje bilo koju pesmu davanjem pravaca, koraka tj. Intervala – odnosno dobro definisanih pravila gledano iz ugla algoritma.

Ko će ići na izlet?

Dabar škola organizuje izlet. Kiki je školski robot koji pomaže nastavniku u određivanju učenika koji će ići na izlet. Robot Kiki radi prema uputstvima i on ima svega dva uputstva na osnovu kojih određuje ko će ići na izlet:

Uputstvo 1: $X1 \rightarrow X2$ znači ako dabar $X1$ ide na izlet onda će i dabar $X2$ ići

Uputstvo 2: $X1 \& X2 \rightarrow X3$ znači ako isključivo i dabar $X1$ i dabar $X2$ idu na izlet, onda će i dabar $X3$ ići. U slučaju da ne idu dabar $X1$ ili dabar $X2$ tada neće ići ni dabar $X3$.

Kiki ima zadatak da utvrdi ko će ići na izlet u jednom odeljenju kog čini 9 dabrova (A, B, C, D, E, G, H, J, i L).

Nastavnik je siguran da sledećih 5 dabrova ide na izlet: A, C, E, G, H; ali je i dodelio tri uputstva robotu Kikiju da izabere još nekog dabra za ekskurziju:

Uputstvo 1: $A \rightarrow D$

Uputstvo 2: $C \& D \rightarrow L$

Uputstvo 3: $L \& B \rightarrow J$

Koliko će dabrova ići na izlet nakon što Kiki odredi ko će ići na ekskurziju?

- a. 5
- b. 6
- c. 7
- d. 8

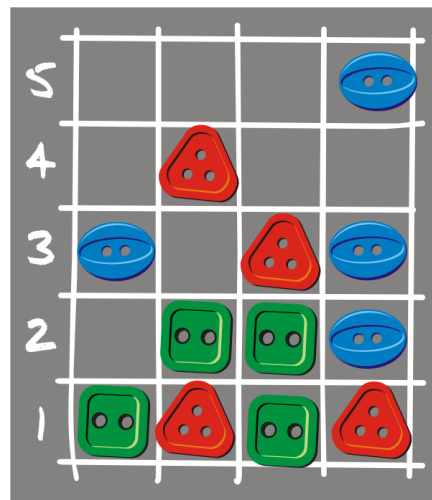
Odgovor: c.7

Informatička pozadina: Ovo je problem logičkog rasuđivanja (rasuđivanje unapred i unazad), koji se često koristi kod veštačke inteligencije.

U ovom zadatku, svi dabrovi iz klase (A, B, C, D, E, G, H, J, i L) su "događaji", i A, C, E, G, H, koji su odabrani dabrovi od strane nastavnika da ide na ekskurziju, su "istiniti događaji" ili "činjenice". Ako se zna da je $X1$ činjenica, tada je $X2$ takođe činjenica, prema prvom uputstvu. Znajući da je $X2$ činjenica, Kiki može da zaključi da je $X3$ je takođe činjenica, zasnovana na drugom uputstvu (i $X1$ i $X2$ su istine, tako da je $X3$ takođe istina).

Igra dugmićima

Ova igra se igra na specijalnoj tabli. Pored table, postoje i dugmići različitih oblika i boja. Dugmići su smešteni po različitim poljima. Dugme se može pomerati samo po jedan korak. Jedan korak znači pomeranje dugmeta na polje iznad, ispod, levo ili desno unutar table. Dozvoljeno je da se jedno dugme pomeri više puta.



Pitanje: Ako je situacija ka ona slici iznad, koji je najmanji broj koraka da bi se zeleni dugmići smestili u red broj 1?

- a. 9
- b. 10
- c. 11
- d. 12

Odgovor: 9

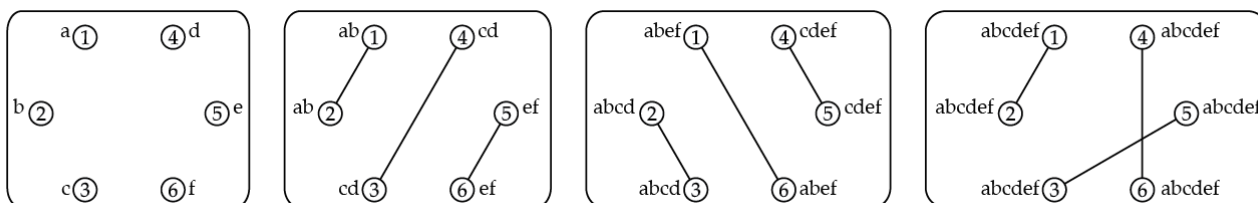
Informatička pozadina: Ovaj zadatak pokazuje koncept algoritma i optimizacije. Optimizacija se može definisati kao pronalaženje najbržeg ili najbolje ostvarljivog učinka u datim ograničenjima.

U praksi faktori, na primer vreme potrebno da kompilator izvrši svoj zadatak, postavljaju gornju granicu optimizacije koju implementator kompilatora može dati. Takođe, u prošlosti su ograničenja memorije bila glavni faktor u izboru optimizacije. U zadatku treba naći minimalan broj pokreta za postizanje cilja.

Špijuni

Svakog petka šest špijuna razmenjuju informacije sakupljene tokom nedelje. Špijuni se viđaju samo u parovima tj. špijun se nikada ne može videti sa dva ili više špijuna istovremeno. Dakle, oni moraju da sprovedu nekoliko rundi sastanaka na kojima razmenjuju informacije.

Grupi od 6 špijuna je potrebno svega 3 runde za prenos svih informacija prikupljenih tokom nedelje. Na primer ako pre sastanka svaki od špijuna zna po jednu informaciju (prvi špijun zna informaciju 'a', drugi špijun zna informaciju 'b', itd...). U prvoj rundi se sastaju špijuni 1 i 2 i razmenjuju informacije, tako da posle oboje znaju 'ab'. Dijagram ispod pokazuje sve runde sastanaka špijuna. Na dijagramu su takođe prikazane i informacije koje špijuni znaju posle sastanka. Posle tri runde sastanaka, svih 6 špijuna znaju sve informacije.



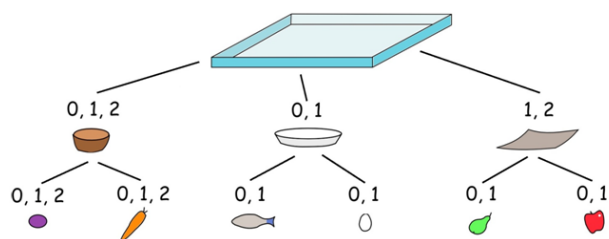
Pitanje: Nakon međunarodnog incidenta, jedan špijun je prestao da prisustvuje sastancima. Koji je minimalan broj rundi sastanaka potrebnih da bi svih 5 špijuna razmenilo informacije?

- a. 3
- b. 4
- c. 5
- d. 6

Tačan odgovor: b.4

Zdrav ručak

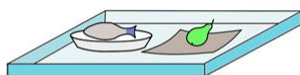
Hm, šta ćemo jesti danas? Dobar restoran daje preporuke za ručak. Preporuke su prikazane na slici ispod. Na svoj plato, za hranu, možete staviti tri različite vrste tanjira. Brojevi prikazuju koliko tanjira ovog tipa možete staviti na plato. U svaki tanjir možete staviti samo hranu koja se nalazi ispod tanjira. Brojevi, pored hrane, pokazuju koliko se određene hrane može staviti u tanjir.



Pitanje:

Jedan ručak nije sastavljen prema slici. Koji?

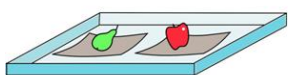
A



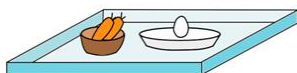
B



C



D



Tačan odgovor:D

Informatička pozadina: Slika definisana preporukama je primer dijagrama, zvanog drvo. Izgleda slično drvetu sa korenom na vrhu. Programeri koriste takve dijagrame da bi definisali strukturu agregata. Agregat je kompleksan objekat koji sadrži jednostavnije delove. Svaki deo može da se sastoji od jednostavnijih delova.

Dabar Alhemičar

Dabar Alhemičar može da pretvori jedan objekat u drugi. On može pretvoriti:

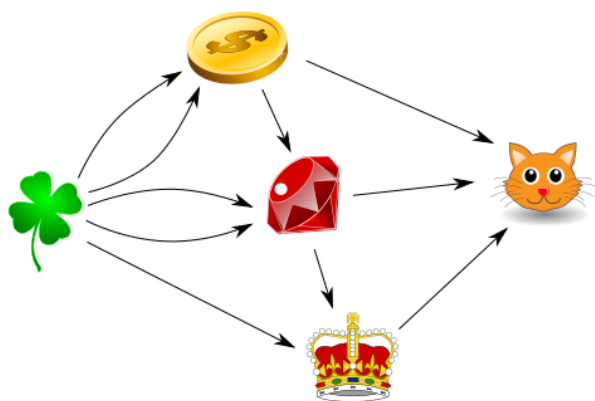
dve deteline u novčić,

novčić i dve deteline u rubin,

detelinu i rubin u krunu,

novčić, rubin i krunu u mačku.

Nakon što se objekat pretvori u drugi objekat, oni nestaje.



Pitanje: Koliko detelina je dabru Alhemičaru potrebno da kreira mačku?

A -5

B -10

C -11

D -12

Odgovor je 11. To vidimo iz sledećeg:

Novčić=2 deteline

Rubin= 2 deteline + 1 novčić=4 deteline

Kruna= 1 rubin + 1 detelina=4 deteline + 1 detelina= 5 detelina

Mačka= 1 novčić + 1 rubin + 1 kruna= 2 deteline + 4 deteline + 5 deteline = 11 detelina

Informatička pozadina: Zadatak prikazuje kako se grafici mogu koristiti da prikažu razliku između stavri. Grafik je struktura podataka koji se dosta koristi u informatici da bi prikazali odnose. Grafici takođe olakšavaju da vizualizujemo i zamislamo zadatke za razliku od čitanja odnosa u tekstu.

BD kod

Dabrovi žele da kodiraju brojeve i zato su razvili Brzi – Dabar – Kod (BD – Kod). BD – Kod je grafički kod koji se sastoji od velikog kvadrata u kome se nalaze 3x3 polja. Svako polje ima određenu vrednost. Sledeće polje, u velikom kvadratu, uvek ima duplu vrednost u odnosu na prethodno. Polja su ispunjena redom od dna prema vrhu kvadrata i uvek sa desne u levu stranu. Primer ispod prikazuje vrednosti prva 4 polja.

...
...	...	8
4	2	1

Kada kodiraju brojeve, dabrovi potamne neka polja. Kodirani broj je rezultat zbira vrednosti zatamnenih polja.

	■	
		■

Broj kodiran u ovom slučaju je 17.

Pitanje:

Na slici ispod prikazan je kvadrat sa potamnjenim poljima. Koja je najveća vrednost koju BD - Kod može prikazati, ukoliko rotiramo ceo kvadrat, bez promene položaja potamnjenih polja?

	■	
■		
		■

A -266

B -250

C -300

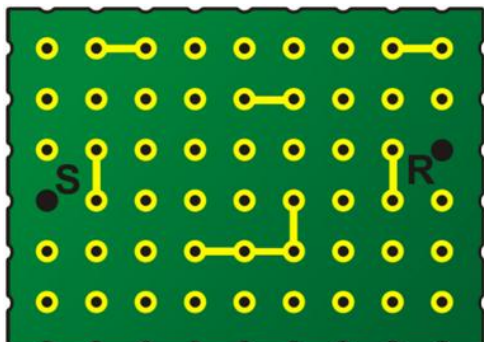
D -270

Odgovor: 266

■		
		■
	■	

Pravljenje čipa

Mali čip se sastoji od mreže kontakata (koji su označeni kao tačke). Neki su već povezani (označeni su kao linijski segmenti). Konektori (poveznici) mogu povezati samo susedne kontakte, horizontalno ili uspravno. Treba povezati S i R neprekidnim nizom konektora koji ne dotiču već povezane kontakte na pločici.



Pitanje: Koliko ima različitih puteva da se povežu S i R najmanjim brojem konektora?

A -5

B -13

C -15

D -16

Nije teško pronaći *neku* najkraću putanju (niz konektora). Može se zamisliti talas koji se širi pločicom, kontakt po kontakt, počevši od S idući ka R. Kada talas dostigne kontakt, to je svakako najmanji mogući broj konektora ("koraka" talasa).

Tabela prikazuje središte talasa tokom njegovog popunjavanja. Brojevi pokazuju redosled u kom ih je talas dostigao, a zatamnjene ćelije pokazuju konektore na koje se ne može povezati. Ovi brojevi su takođe i dužine najkraćih putanja do datog kontakta. Najveći brojevi su trenutna granica talasa.

3	#	#	6	#	#
2	3	4	5	#	#
1	#	5	6	#	R
S	#	4	5	6	#
1	2	3	#	#	#
2	3	4	5	6	

Tačan odgovor: 15 (kao zbir u polju R).

Informatička pozadina: Danas se integrisana kola (čipovi) koriste u svim elektronskim uređajima i unela su revoluciju u svet elektronike. Čipovi se mogu napraviti kompaktni sa nekoliko milijardi potrebnih tranzistora na veoma malom prostoru veličine nokta. Bez njih celokupna informatika bila bi samo teorija.

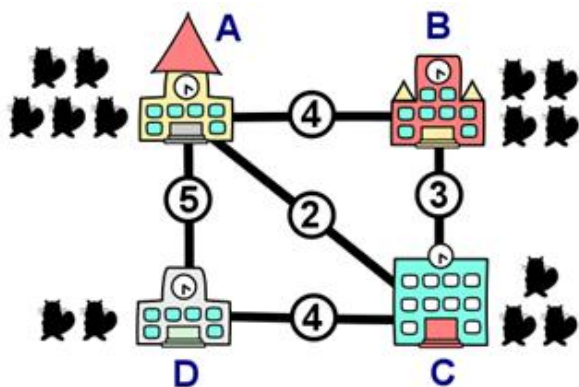
Čipovi su tako pretrpani komponentama i nije ih lako projektovati. Naučnici računarstva koriste mnoge pametne algoritme da to urade. Kada treba odlučiti gde staviti neke dve komponente, jedan kriterijum može biti broj putanja između njih: što je više raspoloživih putanja, to ima manje ograničenja za ostatak pločice, zato što neka treća komponenta lako može blokirati ove dve komponente.

Pronalaženje najkraće putanje je čest, opšti problem u informatici. Opisana procedura sa "talasom" zapravo se zove *pretraga u širinu* ("na prvi dah – *breadth-first search*"). Ovde je prilagođena da prebroji sve najkraće putanje. Nije bilo potrebno ispitati sve moguće putanje jer se postupalo sistematski od početka. Ovaj pristup se zove *dinamičko programiranje*.

Sastanak

U Dabar-gradu postoje četiri škole. Treba da se održi sastanak kome će prisustvovati po nekoliko nastavnika iz svake škole i to petoro nastavnika iz škole A, četvero nastavnika iz škole B, troje nastavnika iz škole C i dvoje nastavnika iz škole D.

Na mapi se vidi raspored škola. Mapa prikazuje i autobuske linije koje povezuju škole, kao i cene karata za svaku autobusku liniju.



Sastanak će se održati u jednoj od škola. Nastavnici iz ostalih škola će doputovati autobusom do odabrane škole i pri tom moraju platiti prevoz. Primetićete, neki nastavnici će morati da koriste dve autobuske linije kako bi došli do određene škole.

Pitanje: Vi treba da odlučite u kojoj će se školi održati sastanak, ali tako da ukupna svota novca koju će nastavnici platiti za karte bude najmanja. U kojoj školi bi se trebalo održati sastanak?

A -A

B -B

C -C

D -D

Tačan odgovor je škola C.

$$A: 0*5 + 4*4 + 2*3 + 5*2 = 32$$

$$B: 4*5 + 0*4 + 3*3 + (3+4)*2 = 43$$

$$C: 2*5 + 3*4 + 0*3 + 4*2 = 30$$

$$D: 5*5 + (3+4)*4 + 4*3 + 0*2 = 65$$

Informatička pozadina:

Problem optimizacije podrazumeva pronalaženje najboljeg rešenja između svih mogućih. Problem optimizacije ponekad je izveden iz modelovanja stvarnog životnog problema. Zahtev predstavljenog problema je dovoljno jednostavan da je lako dobiti najkraći put između bilo koje dve tačke, što u stvarnosti često nije slučaj.

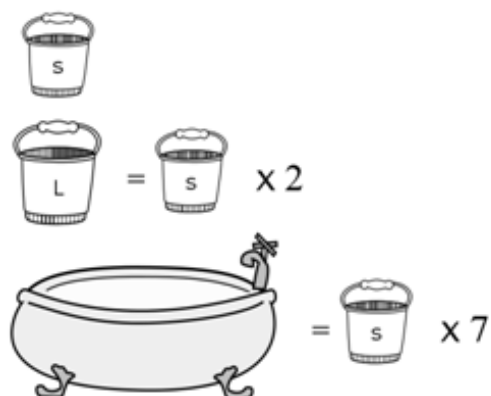
Pronalaženje najkraćeg puta uz pomoć dijagrafa (drveta odlučivanja) je jedno od opštih načina rešavanja. Problem zadovoljava nejednakost trougla gde je direktna veza uvek kraća od svake indirektna. Ovaj slučaj ne zadovoljava nejednakosti trougla. U takvim slučajevima morate biti pažljivi kako biste uporedili i proverili sva moguća rastojanja puteva.

Donošenje vode

Svakog dana posle škole Ceca puni svoju kadu za kupanje vodom.

Ona ima dve kante koje koristi za donošenje vode sa obližnjeg izvora. Velika kanta zahvata dva puta više od male. Kada za kupanje zahvata sedam malih kanti vode.

Ceci je potrebno 3 minuta hoda, od kuće do izvora, kada nosi bilo koju praznu kantu. Svaki put nosi po jednu kantu vode. Kada se vraća sa malom ili velikom kantom punom vode, potrebno joj je 4 ili 5 minuta od izvora do kuće, u zavisnosti od veličine kante.



Pitanje: Koliko je najmanje vremena potrebno Ceci da bi napunila kadu za kupanje vodom sa izvora?

A - 19 minuta

B - 31 minut

C - 28 minuta

D - 49 minuta

Tačan odgovor: B, 31 minut.

Informatička pozadina: Na zadatak se može gledati kao na specijalan slučaj Problema ranca. Da bismo ga rešili, najpre porocenjujemo brzinu popunjavanja za svaki segment. Kada Ceca koristi malu kantu, brzina punjenja je $1/7$ male kante u minutu. Velika kanta daje brzinu $2/8=1/4$ male kante u minutu. Ova brzina popunjavanja kade je brža tako da koristimo veliku kantu kada god je to moguće.

Ovakav tip algoritma naziva se Pohlepni (greedy) algoritam, jer svaki put biramo varijantu koja je najbolja za nas u tom trenutku ne obazirući se na prethodne izbore. U opštem slučaju, takvi algoritmi ne daju optimalno rešenje.

Pretpostavimo, dodali smo još jednu kantu (najveću kantu), koja je tri puta veća od male. Neka je potrebno 10 minuta da Ceca u njoj donese vodu. Njena brzina je $3/10$ male kante u minutu. Ovo je najbrža kanta, pa ćemo je koristiti dva puta, a zatim doneti vodu u maloj kanti. Potrebno vreme da se kada napuni vodom je: $10+10+7=27$ minuta.

Ako najveću kantu koristimo jednom, a zatim korstimo veliku kantu dva puta, onda će potrebno vreme biti: $10+8+8=26$ minuta, što je kraće vreme (bolje) od 27 minuta.