

Dabar 2023/24.



Školski nivo takmičenja za školsku 2023/2024. Godinu
Zadaci i rešenja za učenike starijih razreda osnovne škole i srednje škole



Sadržaj:

O takmičenju i priručniku.....	3
Bodovna tabela	4
Nizovi brojeva.....	7
Četiri pločice	9
Pretvaranje slika u brojeve.....	13
Sigurnosni sistem	15
Sakupljanje kamenja	17
Igra skakanja	21
AI sistem.....	23
Skladište brašna	25
Zlatni tiket	28
Košnica 2	30
Šifrovana poruka	32
Programerski zaliv.....	34
Fabrika robota	36
Kutija sa blagom.....	39
Lavirint	42
Tkanje čílima	44
Jagode	47
Baza podataka o dabrovima.....	50
Skladišta	52
Hamming Leming	54
Digitalni broj.....	57
Ujedinjenje plemena.....	59
Izgubljena slika	61
Tajna poruka (Srbija)	64



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 11. godinu zaredom družimo. U proteklom periodu na takmičenju je učestvovalo preko 438.502 takmičara, a na ovogodišnjem školskom nivou 64.352. 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 2023/24., ŠTO VIŠE DECE UVIDI DA SE SA INFORMATIČKIM PROBLEMIMA SUSREĆU U SVAKODNEVNOM ŽIVOTU I DA IH JE MOGUĆE SA LAKOĆOM REŠAVATI.

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

Priručnik je i deo riznice "Dabar" intelektualnih problema, koja se iz godine u godinu uvećava. Priručnik 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 riznici zadataka ćemo dodati nove. No, čak i oni će biti vrlo brzo odrađeni od strane onih koji vole takve zadatke. Šta onda?

Pozivamo vas da pratite naše aktivnosti na sajtu <https://dabar.edu.rs/> ili na sajtu Međunarodnog takmičenja "Dabar" <https://www.bebbras.org/> i da se zajedno sa nama radujete novim zadacima.

Uživajte u rešavanju zadatka!

Srdačno Vaš, Programski odbor takmičenja Dabar



Bodovna tabela

KiloDabar (5.i 6. Razred OŠ)

R.Br.	Zadatak:	Broj bodova
1	Košnica 2	+6 -4
2	Zlatni tiket	+6 -4
3	Jagode – izmenjen u kopiji	+6 -4
4	Sigurnosni sistem	+9 -3
5	Igra skakanja	+9 -3
6	Tkanje čilima	+9 -3
7	Programerski zaliv	+9 -3
8	Četiri pločice	+9 -3
9	Tajna poruka (Srbija)	+12 -2
10	Skladište brašna	+12 -2
11	Kutija sa blagom	+12 -2
12	Digitalni broj	+12 -2

BODOVI ZA TAČAN ODGOVOR + NEGATIVNI BODOVI ZA POGREŠAN ODGOVOR -

Svi učenici na početku rada dobijaju +35 bodova.

MegaDabar (7.i 8. Razred OŠ)

R.Br.	Zadatak:	Broj bodova
1	Nizovi brojeva	+6 -4
2	Tkanje čilima	+6 -4
3	Igra skakanja	+6 -4
4	Skladište brašna	+9 -3
5	Šifrovana poruka	+9 -3
6	Četiri pločice	+9 -3
7	Kutija sa blagom	+9 -3
8	Tajna poruka (Srbija)	+9 -3
9	Fabrika robota	+12 -2
10	Izgubljena slika	+12 -2
11	Digitalni broj	+12 -2
12	Lavirint	+12 -2

BODOVI ZA TAČAN ODGOVOR + NEGATIVNI BODOVI ZA POGREŠAN ODGOVOR -

Svi učenici na početku rada dobijaju +35 bodova.



GigaDabar (1. i 2. Razred SŠ)

R.Br.	Zadatak:	Broj bodova
1	Četiri pločice	+6 -4
2	Skladište brašna	+6 -4
3	Pretvaranje slika u brojeve	+6 -4
4	Hamming leming	+9 -3
5	Fabrika robota	+9 -3
6	Lavirint	+9 -3
7	Tajna poruka (Srbija)	+9 -3
8	Skladišta	+9 -3
9	AI sistem	+12 -2
10	Baza podataka o dabrovima	+12 -2
11	Sakupljanje kamenja	+12 -2
12	Ujedinjenje plemena	+12 -2

BODOVI ZA TAČAN ODGOVOR +

NEGATIVNI BODOVI ZA POGREŠAN ODGOVOR -

Svi učenici na početku rada dobijaju +35 bodova.

TeraDabar (3. i 4. Razred SŠ)

R.Br.	Zadatak:	Broj bodova
1	Četiri pločice	+6 -4
2	Skladište brašna	+6 -4
3	Pretvaranje slika u brojeve	+6 -4
4	Hamming leming	+9 -3
5	Fabrika robota	+9 -3
6	Lavirint	+9 -3
7	Tajna poruka (Srbija)	+9 -3
8	Skladišta	+9 -3
9	AI sistem	+12 -2
10	Baza podataka o dabrovima	+12 -2
11	Sakupljanje kamenja	+12 -2
12	Ujedinjenje plemena	+12 -2

BODOVI ZA TAČAN ODGOVOR +

NEGATIVNI BODOVI ZA POGREŠAN ODGOVOR -

Svi učenici na početku rada dobijaju +35 bodova.



Izbor zadataka za takmičenje i prevod, Programski odbor takmičenja:

1. **Milan Rajković** (predsednik programskog odbora):

Nastavnik računarstva i informatike u Politehničkoj školi "Milutin Milanković" u Kruševcu i nastavnik informatike i računarstva i tehnike i tehnologije u OŠ "Dragomir Marković" u Kruševcu.

2. **Saša Jevtić** (član programskog odbora):

Nastavnik tehnike i tehnologije u OŠ "Dragomir Marković" u Kruševcu.

3. **Suzana Miljković** (član programskog odbora):

Nastavnik razredne nastave u OŠ "Kralj Petar I" u Nišu.

4. **Jelena Hadži-Purić** (član programskog odbora);

5. **Bojan Milosavljević** (član programskog odbora):

Profesor strukovnih studija iz informatike i računarstva, Rukovodilac Odseka tehničko-tehnoloških studija Kruševac. Akademije vaspitačko-medicinskih strukovnih studija Kruševac.

6. **Marija Andonović Radojević** (član programskog odbora);

Nastavnik informatike i računarstva u OŠ "Svetozar Marković" u Kragujevcu.

7. **Marina Lazarević** (član programskog odbora).

Nastavnik tehnike i tehnologije i informatike i računarstva u OŠ "Nada Popović" u Kruševcu

8. **Gordana Brstina** (prevod zadataka):

Nastavnik tehnike i tehnologije i informatike i računarstva u OŠ "Marija Trandafil" u Novom Sadu (Veternik).

9. **Zorana Đorđević** (prevod zadazaka):

Nastavnik informatike i računarstva i tehnike i tehnologije u OŠ "Kosta Trifković" i OŠ "Veljko Vlahović" u Novom Sadu.

Tehnička podrška:

1. Gašper Fele-Žorž (Fakultet za računarstvo i informatiku, Univerzitet u Ljubljani)
2. Ivica Bekrić – Profesor tehnike i informatike.



Nizovi brojeva

Tekst zadatka:

Niz brojeva **3, 5, 2, 4, 1** možemo vizuelno predstaviti na sledeći način (crveni brojevi u tabeli označavaju pozicije brojeva u nizu).

	1	2	3	4	5
X	3	5	2	4	1

Pišemo **(X 2)** da opišemo broj na poziciji **2**. Dakle **(X 2)** je **5**. Slično **(X 5)** je **1**.

Pozicije se mogu naznačiti ugnježdeno. Na primer, **(X (X 3))** je **5** jer je **(X 3)** **2**, pa je **(X (X 3)) = (X 2) = 5**.

	1	2	3	4	5
A	3	5	2	4	1

Pitanje/Izazov:

Ako je:

A	3	2	4	1	5
---	---	---	---	---	---

B	5	4	1	3	2
---	---	---	---	---	---

C	2	5	4	3	1
---	---	---	---	---	---

Koji broj je opisan sa **(A (B (C 3)))?**

Ponuđeni odgovori:

- A) 4
- B) 1
- C) 2
- D) 5



Tačan odgovor i objašnjenje:

Tačan odgovor je pod A).

$$(C \ 3) = 4, \text{ dakle } (B \ (C \ 3)) = (B \ 4) = 3, \text{ dakle } (A \ (B \ (C \ 3))) = (A \ (B \ 4)) = (A \ 3) = 4$$

Informatička pozadina:

Strukture podataka su neophodne za programiranje. Posebno su korisni oni koji mogu da sadrže čitav spisak podataka. Strukture podataka, kao i mnogi drugi entiteti u računarstvu i programiranju, mogu biti međusobno povezani, tako da element jedne liste može opisati poziciju na drugoj listi. Ovaj indirektni način opisivanja pozicija je moćan koncept. U početku može izgledati zbumujuće, ali nije teško ako izračunate vrednosti korak po korak.



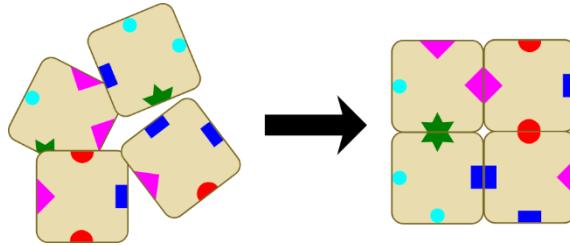
Četiri pločice

Tekst zadatka:

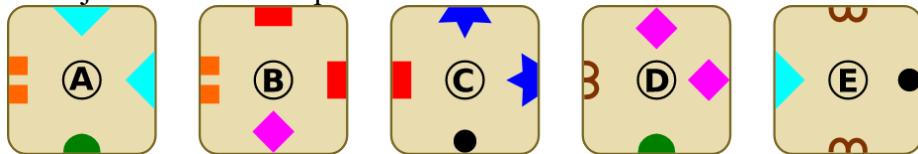
Od vas se traži da rasporedite četiri pločice u kvadrat 2×2 , prema sledećem pravilu:

- Pločice se mogu dodirivati samo na stranama koje nose potpuno isti simbol!

Slika ispod pokazuje primer:



Sada vam je dato sledećih 5 pločica.



Morate da rasporedite četiri od ovih pet pločica u kvadrat veličine 2×2 koji sledi pravilo iznad. (U ovom slučaju postoji samo jedan mogući izbor od četiri pločice koji to dozvoljavaju.)

Pitanje/Izazov:

Koju pločicu **NEĆETE** koristiti?

Ponuđeni odgovori:

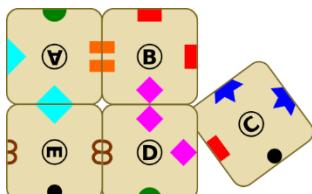
- A) A
- B) B
- C) C
- D) D



Tačan odgovor i objašnjenje:

Tačan odgovor je C.

Moguće je napraviti kvadrat 2×2 koji prati pravilo koristeći pločice A, B, D i E, kao što je prikazano ispod:

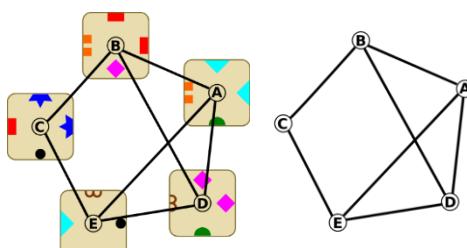


Isprobavanje mnogih mogućnosti dok se jedna ne uklopi je opcija, ali postoji način da se smanji broj mogućnosti koje treba proveriti.

Ovo radimo u dva odvojena koraka.

- 1) Pravimo dijagram od pet pločica i povezujemo ih linijom ako dele bar jedan simbol.

Objašnjenje dijagrama: C i E su povezani linijom jer obe pločice imaju mali crni krug na jednoj od svojih strana. A i C nisu povezani jer se nijedan od simbola na A ne pojavljuje na C. (Pojednostavljeni dijagram sa desne strane naziva se graf.)



Objašnjenje pojma, ciklus dužine 4: putanja od pločice do pločice koja prati linije i završava se na početku nakon 4 koraka od jedne pločice do druge. Važno je shvatiti da četiri pločice u kvadratu 2×2 koje prate pravilo odgovaraju 4 ciklusa.

- 2) Pomoću dijagrama saznajemo sve moguće cikluse dužine 4. U našem dijagramu postoje tri ciklusa dužine 4.:

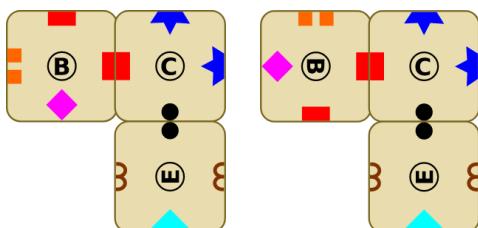
- a) Prvi ide od A do B do D do E, a zatim nazad do A (Taj 4-ciklus odgovara našem rešenju).
- b) Drugi je A-B-C-E-A.
- c) Treći je B-C-E-D-B.



Naravno, sve ciklične permutacije pet slova (tj. počevši od bilo kog slova, ali držeći redosled) su ekvivalentne, kao i njihov obrnut poredak.. Na primer, **A-B-C-E-A**, **B-C-E-A-B** i **E-C-B-A-E** su ekvivalentni.

Da li je moguće da drugi ili ciklus dužine 4 dovedu do drugog rešenja naše slagalice?

Oba ciklusa sadrže **B-C-E** i postoje samo dve mogućnosti da se ove pločice spoje tim redosledom:



Ni pločica **A** ni **D** se ne mogu koristiti za dovršavanje ovih kvadrata **2 x2** (i dalje u skladu sa pravilom).

Informatička pozadina:

Za rešavanje računarskog problema često je korisno (privremeno) zanemariti neke njegove detalje. (Ovo se zove apstrakcija i važna je veština računarskog razmišljanja). U našem prvom koraku zanemarili smo oblik i boje simbola i njihove pozicije na pločicama. Zanimalo nas je samo da li dve pločice dele simbol ili ne.

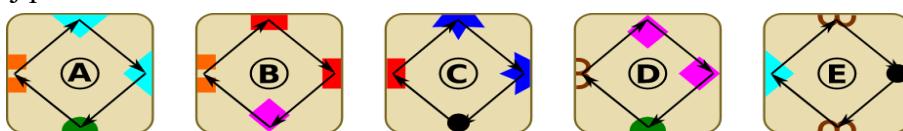
Dobijeni graf nam je već pružio dovoljno informacija da malo skratimo našu potragu za rešenjem. Ostala su samo tri moguća rešenja.

Grafovi se često koriste u računarstvu kao apstrakcija podataka koje treba obraditi. Kroz istoriju mnogi ljudi su napravili pametne i brze metode za određivanje različitih svojstava takvih grafova, na primer, da bi lako pronašli sve cikluse dužine 4.

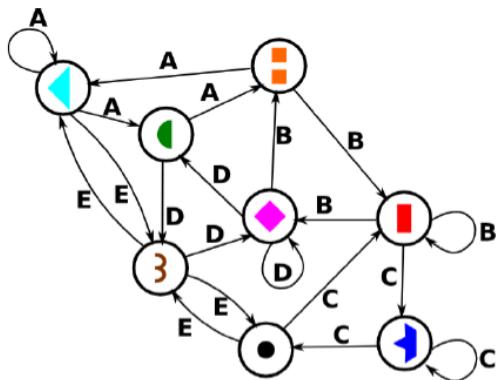
Da bismo dalje rešili problem, bio nam je potreban drugi korak, koji je ponovo koristio graf kao apstrakciju problema. Podela teškog problema na zasebne korake koji se mogu lakše rešiti naziva se dekompozicija - opet važna veština računarskog razmišljanja.

Alternativno rešenje:

Možemo koristiti drugi dijagram, koji konstruišemo na sledeći način: povežite simbole na istoj pločici strelicom na sledeći način:



Sada koristite ove strelice kao veze u novom dijagramu u kojem se svaki simbol pojavljuje samo jednom (ovaj dijagram se naziva usmereni graf):



Stavili smo 'ime' pločice (**A,B,C,D,E**) na odgovarajuću strelicu.

Ponovo valjano rešenje se sastoji od 4-ciklusa na ovom dijagramu, ali ovog puta sve strelice moraju pratiti pravac u kome bismo hodali duž 4-ciklusa. Postoji takav ciklus **A-E-D-B-** koji odgovara našem rešenju.

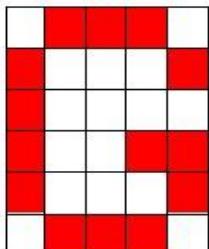
Postoji i ciklus **E-C-B-D-** koji je 'skoro rešenje', osim što je **D**-strelica u pogrešnom smeru. Postoji put **A-B-C-E-**, ali ne završava tamo gde je počeo (dakle nije ciklus) i svejedno, **C**-strelica je u pogrešnom smeru. Naše rešenje je dakle jedino ispravno.



Pretvaranje slika u brojeve

Tekst zadatka:

Sliku **6x5** koja se sastoji od crvenih i belih polja možemo prikazati (kodirati) nizom brojeva koji predstavljaju broj uzastopnih crvenih i bela polja u jednom redu naizmenično. Prvi broj u nizu uvek odgovara broju uzastopnih belih polja na početku reda.



1, 3, 1
0, 1, 3, 1
0, 1, 4
0, 1, 2, 2
0, 1, 3, 1
1, 3, 1

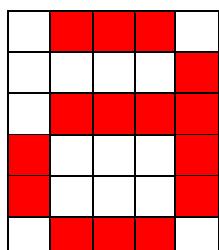
Na ovaj način možemo prikazati sliku nizom brojeva.

Ako sve navedene brojeve zapišemo u jedan niz, to će izgledati ovako:

1, 3, 1, 0, 1, 3, 1, 0, 1, 4, 0, 1, 4, 0, 1, 3, 1, 1, 3, 1.

Pitanje/Izazov:

Koji niz brojeva opisuje sliku ispod?



Ponuđeni odgovori:

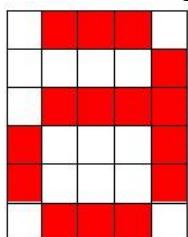
- A) 0, 1, 3, 4, 1, 1, 3, 1, 0, 2, 2, 1, 0, 1, 3, 1, 2, 2, 1
- B) 1, 3, 1, 4, 1, 1, 4, 0, 1, 3, 1, 0, 1, 3, 1, 1, 3, 1
- C) 1, 3, 1, 0, 1, 4, 1, 4, 0, 1, 3, 1, 0, 1, 3, 1, 1, 3, 1
- D) 1, 3, 1, 4, 1, 1, 4, 1, 3, 1, 1, 3, 1, 1, 3, 1



Tačan odgovor i objašnjenje:

Tačan odgovor je **B**.

Da biste pronašli pravi odgovor potrebno je konvertovati najmanje četiri reda slike u brojeve, a zatim ih spojite.



1, 3, 1
4, 1
1, 4
0, 1, 3, 1
0, 1, 3, 1
1, 3, 1

Dobijeni niz je 1, 3, 1, 4, 1, 1, 4, 0, 1, 3, 1, 0, 1, 3, 1, 1, 3, 1, što je odgovor B. Ostala tri odgovora predstavljaju druge slike, jer za svaku postoji jedinstvena reprezentacija slike, fiksirajući veličine mreže i boju koju treba koristiti.

Informatička pozadina:

Da biste sačuvali ili preneli slike između elektronskih uređaja, potrebno ih je pretvoriti u brojeve. Postoji mnogo načina da se to uradi, a ovaj zadatak predstavlja jedan od njih, poznat kao kodiranje dužine rada. Korišćenjem jednog broja za predstavljanje niza kvadrata iste boje, ovaj metod vrši neku kompresiju podataka, a to je važno polje u računarstvu.

Ovaj metod se može poboljšati na mnogo načina, na primer, dozvoljavanjem broja da se odnosi na niz kvadrata koji obuhvata više od jedne linije.

Pronalaženje načina za predstavljanje podataka predstavljalo je izazov još od prvih elektronskih mašina, jer izbori mogu uticati na vreme potrebno za obradu ili slanje i primanje informacija. Trenutno, ovo je i dalje problem posebno na internetu: stvaranje novih načina kodiranja slika, video zapisa i drugih medijskih datoteka može ubrzati naše iskustvo pregledanja. Metod opisan u pitanju koristili su stari faks aparati za prenos sadržaja dokumenata.

<http://desplugged.ime.unicamp.br/atividade2/index.html>

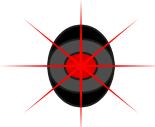
<https://classic.csunplugged.org/activities/image-representation/>



Sigurnosni sistem

Tekst zadatka:

Banci je potreban novi sigurnosni sistem da bi zaštitila prostoriju ispred svog sefa od uljeza. Kupili su pet detektora. Svaki detektor šalje laserske snopove u osam smerova.

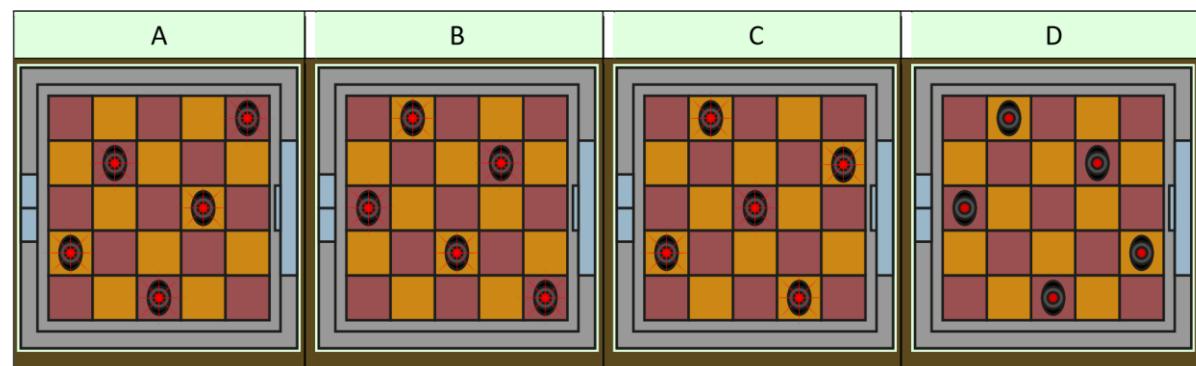


Alarm se aktivira ako uljez prekine laserski snop bilo kog detektora. Alarm se takođe aktivira, ako jedan od detektora stane na put laserskom snopu drugog.

Banka želi da postavi pet detektora na kvadratne podne pločice tako da pokrivaju sve preostale pločice u prostoriji, ali da ne aktiviraju jedni druge. Oni predlažu četiri različita načina da se to uradi. Međutim, jedan **ne ispunjava** ove uslove.

Pitanje/Izazov:

Koji raspored detektora ne ispunjava uslove?



Ponuđeni odgovori:

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



Tačan odgovor i objašnjenje:

Tačan odgovor je **D**.

Za rešenje koje zadovoljava sve uslove, prvo se može primetiti da će u svakom redu i u svakoj koloni biti postavljen najviše jedan detektor, pošto oni odašilju horizontalne i vertikalne snopove i ne smeju da detektuju jedan drugog. To bi bilo dovoljno da se otkrije bilo koji uljez. Ali pošto detektori takođe šalju zrake dijagonalno, mora se voditi računa da dva detektora ne budu postavljena na istoj dijagonali.

Uopšteno govoreći, u ispravnom rešenju, sledeći uslovi moraju da važe za bilo koji par dva različita detektora. **K** je broj kolone u kojoj se nalazi prvi detektor i **I** broj reda tog detektora, a **A** je broj kolone u kojoj se nalazi drugi detektor i **B** je broj reda tog detektora.

Onda:

- **K** i **A** ne mogu biti isti zbog vertikalnih laserkoh snopova;
- **I** i **B** ne mogu biti isti zbog horizontalnih laserskih snopova:

- kada oduzmete manje od **K** i **A** od većeg i uradite isto za **I** i **B**, rezultati ta dva oduzimanja ne mogu biti isti, zbog dijagonalnih nopa.

Svi ovi uslovi važe za odgovore A, B i C. Za odgovor D, dva detektora (sa koordinatama 1,3 i 3,5) će uključiti alarm, pošto su postavljeni na istoj dijagonali.

Informatička pozadina:

Problem koji se ovde rešava je varijacija problema “Osam kraljica”, koji je primer problema sa zadovoljenjem ograničenja: imate nekoliko stvari koje mogu imati različite vrednosti i za svaku stvar morate da izaberete jednu od mogućih vrednosti u način da se određeni skup pravila ne prekrši. Takvi problemi se često javljaju u automatizovanom planiranju, koje se koristi da roboti rade; računarska lingvistika, koja je neophodna za prepoznavanje jezika; ili alokaciju resursa, kada preduzeća moraju da odluče ko će raditi sa kojim sredstvima koja su im pri ruci. Mogu se veoma brzo zakomplikovati, tako da je potreban računar da pokuša da ih reši. Računari obično rade na isti način na koji ste verovatno radili: pokušavaju da izgrade rešenje “Malo po malo”, ali čim se pravilo prekrši, vraćaju se jedan ili više koraka unazad i zatim pokušavaju sledeću opciju dok ne pronađu rešenje koje odgovara svim ograničenjima. Ovaj proces se naziva “Vraćanje unazad”.

Constraint satisfaction problem, eight queens puzzle, backtracking

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

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

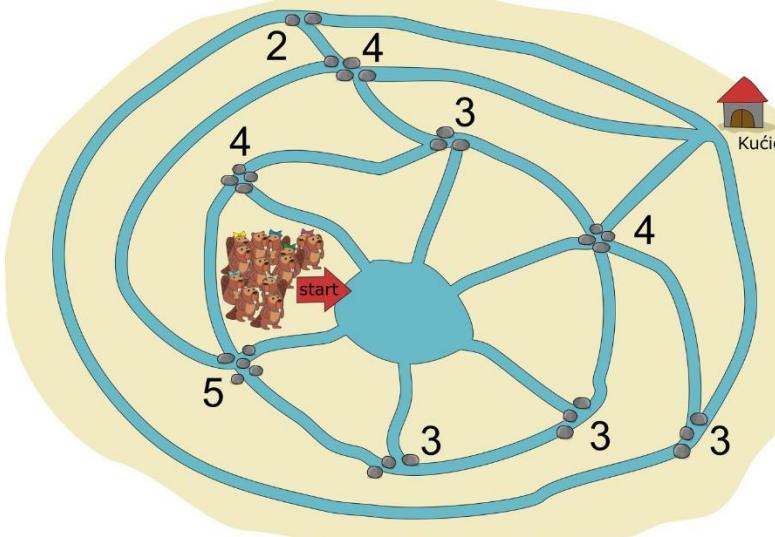
<https://en.wikipedia.org/wiki/Backtracking>



Sakupljanje kamenja

Tekst zadatka:

Porodično imanje dabrova sastoji se od 21 kanala. Na pojedinim raskrsnicama, gde se kanali seku, nalazi se kamenje. Dabrovi žele da to kamenje prebace u kućicu. Kada su brojali kamenje koje treba prebaciti u kućicu, uvideli su da ih je ukupno 31.



Kamenje je teško pa dabar može da nosi samo jedan ili dva kamaena istovremeno. Iznenadujuće, dabrovi plivaju tačno jedan sat od jedne do druge raskrsnice.

Pitanje/Izazov:

Dabrovi kreću istovremeno i potebno je da odnesu u kućicu kamenje u roku od četiri sata. Koliko je najmanje dabrova potrebno kako bi posao završili u roku?

Ponuđeni odgovori:

- A) 14 dabrova
- B) 18 dabrova
- C) 20 dabrova
- D) 24 dabrova

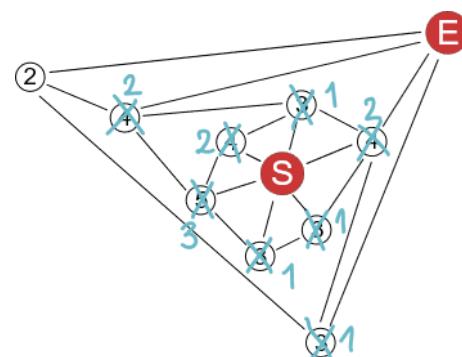


Tačan odgovor i objašnjenje:

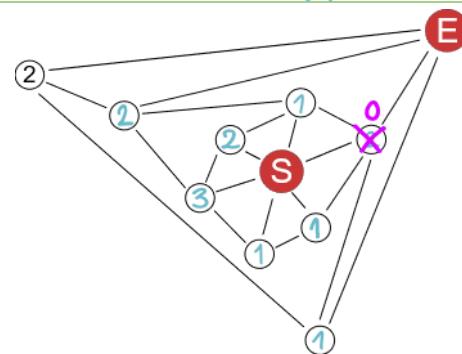
Tačan odgovor je A).

Postoji samo jedan put kojim dabrovi mogu da stignu do kućice za 2 sata. Na ovom putu su 4 kamena. 2 dabra mogu nositi ova 4 kamena do raskrsnice i mogu nastaviti da nose dodatna 4 kamena do raskrsnice. Tada je ostalo $23 - 8 = 15$ kamena i potrebno je još najmanje 12 dabrova, pošto preostalo kamenje leži na stazama dužine 3 ili 4 i u roku od 4 sata nakon odlaganja kamenja nema vremena da se nastavi i doneše još. Pregledom dabrove kuće možemo primetiti da preostala 23 kamena u kućicu može doneti 12 dabrova, od kojih 11 nosi 2 kamena, a 1 nosi samo jedan kamen.

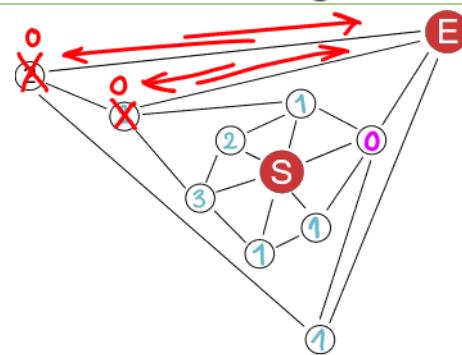
Prvo, razmatramo sve raskrsnice sa najmanje tri kamena. Na svakoj takvoj raskrsnici dabar skuplja dva kamena i donosi u kućicu. U tu svrhu nam je potrebno 8 dabrova. Sakupili su 16 kamenova. Ostalo je još $31 - 16 = 15$ kamenja. Svi dabrovi osim jednog bili su na putu najmanje tri sata. Nisu u stanju da se vrate u kanale i da sakupe dalje kamenje. Ne možemo ih „ponovno koristiti“.



Jedan od dabrova je, međutim, za dva sata sakupio dva kamena. Imao bi dovoljno vremena da se vrati na raskrsnicu, da sakupi još dva kamena i da ih doneše do raskrsnice. Stoga možemo kasnije da „ponovo upotrebimo“ ovog dabra.



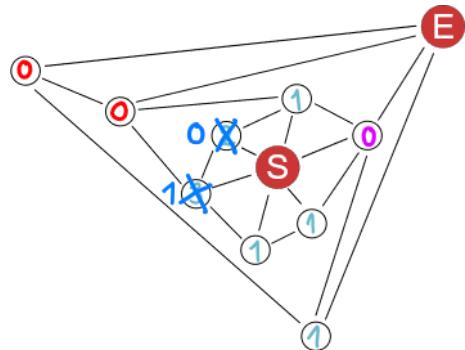
Sada se dva dabra mogu „ponovno koristiti“. Svaki od njih sakuplja dva kamena.



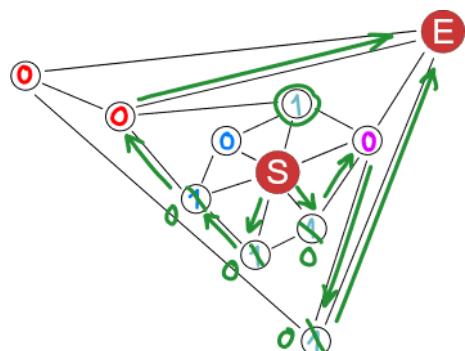
Ostalo je još $15 - 4 = 11$ kamenčića. Do sada smo koristili 9 dabrova. Niko od njih se ne može dalje „koristiti“.



Primenjujemo istu strategiju i dalje razmatramo svu raskrsnicu sa najmanje dva kamena. Trebaju nam 2 dabra. Sakupljaju 4 kamena. Ostalo je još $9-4=5$ kamenčića. Do sada smo koristili $9+2=11$ dabrova.



Poslednji korak se sastoji u pronalaženju rešenja (moguće pokušajima i greškama) koje koristi što manje dabrova. Dva dabra sakupljaju po dva kamena. Treći i poslednji dabar sakuplja poslednji kamen. Najbolje rešenje je A): $11+3= 14$ dabrova.



Ima 31 kamen, samo dva dabra mogu doneti po 2 kamena i onda se vraćaju po još, ostali dabrovi mogu doneti najviše 2. Dakle, potrebno je najmanje 13,5 dabra. Najbolje rešenje zahteva 14 dabrova.

Informatička pozadina:

U očima čitaoca, priča je privlačna i bogata atraktivnim detaljima. Ipak, prva radnja koju preduzima kompjuterski naučnik sastoji se u identifikaciji informacija koje su zaista potrebne za rešavanje problema i u uklanjanju svih ostalih.

Značajne informacije obuhvataju:

- kanali i raskrsnice kanala (kamenje, start i dom).
- brojčana ograničenja: dabar pliva tačno 1 h od jedne raskrsnice do druge; dabrovi moraju biti kod kuće u roku od 4 sata; moraju da sakupe 31 kamen; mogu nositi jedan ili dva kamena.

- Moramo da sakupimo svih 31 kamenčića koristeći što manje dabrova.

S druge strane, ne moramo da razmatramo precizan tok kanala i njihov tačan položaj na karti, izgled dabrova itd. Ovaj podatak se može izostaviti.

Informatičari takav formalni prikaz nazivaju grafom. Graf se sastoji od čvorova i ivica između čvorova. Udaljenost između dva čvora može se izraziti na nekoliko načina, na primer brojanjem ivica koje moramo da pređemo da bismo došli od jednog čvora do drugog. Grafikoni su jedan od ključnih instrumenata za rešavanje problema u životu kompjuterskog naučnika. Navigacioni sistemi su dobro poznati primer aplikacija koje u velikoj meri koriste grafikone.

U našem zadatku, od dabrova se očekuje da sakupe svih 31 kamenčića. Moguće je da postoji više od jednog načina da se postigne ovaj cilj. Moglo bi se zamisliti da se koristi 14, 18, 20 ili čak 31 različit dabar. Međutim, pitanje eksplisitno zahteva rešenje koje koristi što



manje dabrova. Iz ove perspektive, trebalo bi pažljivo razmotriti da li postoji način da se „ponovno koristi“ dabar ne.

Problemi koji traže najbolje moguće rešenje su veoma česti. U terminologiji informatike, to su problemi optimizacije. Od navigacionog sistema se očekuje da izabere najbolji (npr. optimalan) način da stigne od A do B. Naravno, tačno značenje „najboljeg“ rešenja treba unapred definisati. U slučaju navigacionog sistema, kriterijum bi mogao da se sastoji u minimiziranju udaljenosti u km, ili vremena potrebnog automobilu da stigne do cilja.

U smislu računarskog razmišljanja, učenici će verovatno malo pojednostaviti prikaz zadatog zadatka – umesto da crtaju kamenje, pisaće brojeve i kanale će predstavljati linijama. Ipak, slika je već neformalna, ali tačna forma grafikona. Od učenika se zahteva da razlože problem na manje delove kojima se može upravljati, da reše (moguće pokušajima i greškama) svaki od njih i da zaključe da li njihovo rešenje zadovoljava sva ograničenja i izuzetke koji bi potencijalno mogli da se pojave.

mathematical modeling, graph, optimization problem:

[https://en.wikipedia.org/wiki/Graph_\(discrete mathematics\)](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics))

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

Decomposition: <https://www.csunplugged.org/en/computational-thinking/>



Igra skakanja

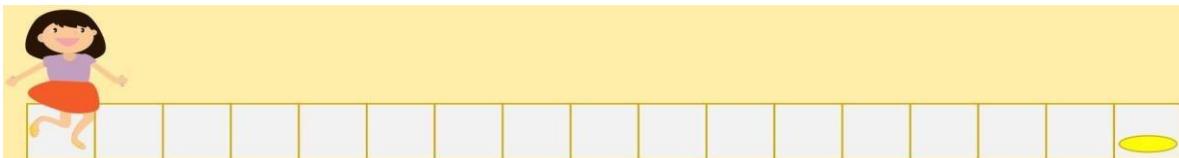
Tekst zadatka:

Valentina voli da skakuće i smišlja pravila za nove igre. U gradskom parku primetila je 17 pločica u nizu i osmisnila pravila za novu igru.

Stavila je novčić na kraj reda pločica i stala na suprotni kraj, tako da je bila okrenuta prema novčiću (vidi sliku).

Postoje dva pravila za skakanje:

- Ako se nalazi na pločici označenoj sa X, ona će preskočiti naredne dve pločice i stati na treću.
- Ako se nalazi na pločici označenoj sa O, ona će skočiti na pločicu unazad.



Pitanje/Izazov:

Koji raspored X i O pločica ce je dovesti tačno do pločice sa novčićem?

Ponuđeni odgovori:

A)

X O X O X O X O X O X O X O X O

B)

X O O X O O X O O X O O X O O X

C)

X O O X X O O X X O O X X O O X

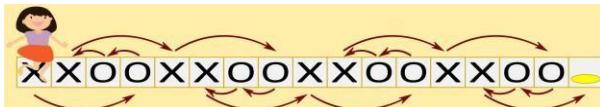
D)

X X O O X X O O X X O O X X O O



Tačan odgovor i objašnjenje:

Tačan odgovor je D.



Strelice pokazuju kako se skače. Odgovor A nije tačan jer je promašila drugu pločicu sa leve strane. Odgovori B i C nisu tačni jer je promašila drugu i treću pločicu sa leve strane.

Informatička pozadina:

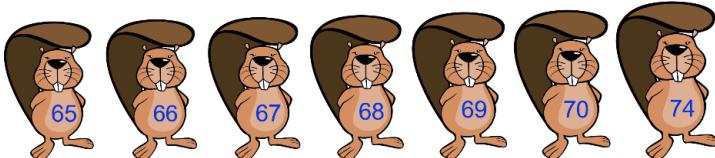
Pronalaženje načina uzastopnih akcija do postizanja cilja prema datim pravilima naziva se algoritmizacija. Robot ili kompjuterska aplikacija mogu se programirati tako da rade po zadatim pravilima. Ako nam je tada potreban određeni izlaz iz njega, moramo postaviti odgovarajući ulaz.

Na primer, ako imamo dugačak uski hodnik oblika L i naš robot za čišćenje može samo da ide pravo i jednom skrene udesno za 90° , moramo da ga stavimo na kraj kraćeg dela „L“ inače ne bi mogla da očisti ceo hodnik.



AI sistem

Tekst zadatka:



Dabrovi konstruišu "zapanjujuće inteligentan" (AI) sistem. Sistem meri visinu životinja i samo na osnovu tog podatka odlučuje da li je životinja dabar ili ne. Sistem veštačke inteligencije uči da donosi odluke na osnovu primera.

Na početku, AI sistem uči na primeru životinja ovih visina:

- 65, 66, 67, 68, 69 => dabar, dabar, dabar, dabar, dabar
- 11, 101, 110, 120, 130 => nije dabar, nije dabar, nije dabar, nije dabar, nije dabar

Zatim, dabrovi puštaju sistem veštačke inteligencije da odluči. Rezultat je sledeći:

- 70, 74 => dabar, dabar
- 86, 38 => nije dabar, nije dabar
- 40, 80 => dabar, dabar

AI je napravio grešku jer dve životinje visine 40 i 80 zapravo nisu dabrovi! AI je primetio da životinja veličine 11 nije dabar, a životinja veličine 65 jeste dabar. Gledajući razliku u veličinama, AI je odlučio da su samo životinje više od 38 i manje od 85 dabrovi. Stoga, da bi poboljšao AI, dabar mu daje novi primer: životinja veličine 42 nije dabar.

Pitanje/Izazov:

Nakon nove obuke, kako AI klasificiće dve životinje veličine 48 i 84?

Ponuđeni odgovori:

- A) dabar, dabar
- B) dabar, nije dabar
- C) nije dabar, dabar
- D) nije dabar, nije dabar



Tačan odgovor i objašnjenje:

Tačan odgovor je: C.

Kako je AI zaključio interval 38-85 korišćen za identifikaciju dabrova? U stvari, ne znamo tačno. Ali pošto među primerima koje je AI videla 11 nije bio dabar, a 65 je bio dabar, AI je izračunao prag da li je dabar ili ne i stavio ga negde između 11 i 65. Razumna tačka za taj prag može biti prosečna vrednost $(11+65)/2 = 38$ i slično rezonovanje se može primeniti na drugu krajnost: $(69+101)/2 = 85$. Koristeći isti postupak, nakon što smo videli novi primer (42 nije dabar) prag od 85 ne bi trebalo da se menja, ali novi levi prag treba da postane $(42+65)/2 = 53,5$. Stoga bi životinja veličine 48 identifikovana kao „ne dabar“, dok bi životinja veličine 84 bila identifikovana kao dabar.

Informatička pozadina:

Sistem „Zapanjujuće inteligentan“ u ovom Bebrasovom zadatku primenjuje algoritam mašinskog učenja. U oblasti mašinskog učenja (ML) kompjuterski naučnici proučavaju kompjuterske algoritme koji mogu da unaprede njihovo donošenje odluka automatski. ML algoritmi grade model zasnovan na podacima primera, poznatim kao podaci o obuci, kako bi mogli da predviđaju ili donose odluke bez eksplicitnog programiranja za to. Dakle, njihov kvalitet može u velikoj meri zavisiti od kvaliteta korišćenih podataka za obuku, kao što možete videti iz ovog zadatka. ML algoritmi se koriste u širokom spektru aplikacija, kao što su medicina, filtriranje epošte, prepoznavanje govora i kompjuterski vid, gde je teško ili neizvodljivo razviti konvencionalne algoritme za obavljanje potrebnih zadataka.

Mašinsko učenje se smatra podoblasti veštačke inteligencije. Možete pronaći onlajn simulaciju mašinskog učenja na <https://machinelearningforkids.co.uk> or at AI for Oceans.

https://en.wikipedia.org/wiki/Machine_learning <https://machinelearningforkids.co.uk>

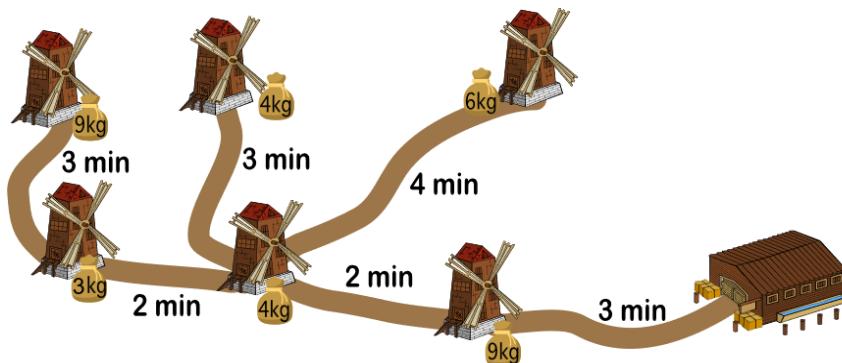


Skladište brašna

Tekst zadatka:

BenBen je vredan vodeničar. Svakog dana, pre zalaska sunca, skuplja džakove brašna iz mlinova i nosi ih u skladište.

Može da nosi više džakova brašna istovremeno, ali njihova ukupna težina ne sme biti veća od 15 kg. Brojevi pokazuju vreme potrebno za prelazak tog dela puta.



Pitanje/Izazov:

BenBen kreće iz skladišta. Želi da prenese brašno koje se nalazi ispred mlinova na najbrži mogući način do skladišta. Za koliko će minuta najbrže završiti ovaj zadatak?

Ponuđeni odgovori:

- A) 50 minuta
- B) 31 minut
- C) 44 minuta
- D) 54 minuta

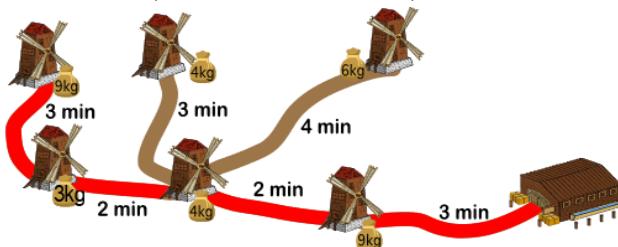


Tačan odgovor i objašnjenje:

Tačan odgovor je A).

Ukupni kilogrami brašna koje treba pokupiti je 35 kg. Pošto dabar BenBen može da ponese najviše 15 kg po putovanju, onda bi morao da preduzme najmanje 3 putovanja da unese sve džakove. Da bi minimizirao svoje vreme, morao bi da pokupi džakove u susednim mlinovima.

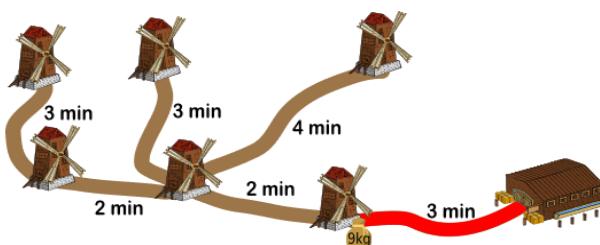
Na primer, prilikom preuzimanja vreće od 9 kg iz gornjeg levog mlinu, BenBen takođe može pokupiti vreću od 3 kg u svom susednom mlinu (ukupna težina: 12 kg). Svi ostali džakovi su teži od 3 kilograma, tako da on ne može pokupiti nijednu drugu vreću. Prvo putovanje bi trajalo 20 minuta ($3+2+2+3+3+2+2+3$).



Da bi pokupili preostale vreće u četiri preostala mлина, drugi put bi morao da donese tri najudaljenija vreća, 6 kg u gornjem desnom mlinu i dva od 4 kg u srednjem mlinu (Ukupna težina: 14 kg). Drugo putovanje bi trajalo 24 min ($3+2+4+4+3+3+2+3$).



Treće putovanje će preuzeti preostalih 9 kg iz mлина najbliže skladištu i trajaće 6 min ($3+3$). Dakle, ukupno putovanje bi trajalo 50 minuta.



Gornji redosled prvog, drugog i trećeg putovanja se može zameniti; ono što je važno je grupisanje od džakova koje treba pokupiti.



Opcija B ne može biti odgovor jer bi preuzimanje vreće od 9 kg iz gornjeg levog mлина već trajalo 20 minuta (pogledajte prvo putovanje iznad). Preostalih 10 min dovoljno je samo za odlazak do drugog mлина i nazad, tako da Ben Ben neće moći da pokupi dva džaka u gornjim mlinovima.

Opcija C ne može biti odgovor jer je 44 min dovoljno vremena za prvo i drugo putovanje, tako da 9 kg u najbližem mlinu ne bi bilo pokupljeno.

Opcija D ne može biti odgovor jer je ovo duže od opcije A.

Informatička pozadina:

U računarskoj nauci, problemi optimizacije su problemi za pronalaženje najboljeg rešenja među svim mogućim rešenjima. Obično se to odnosi na pronalaženje najveće ili najmanje vrednosti funkcije. U ovom zadatku, na primer, od nas se traži da pronađemo najmanje ukupno vreme.

Rešavanje problema korišćenjem računara takođe je često povezano sa optimizacijom. Na primer, želimo da računar može da završi svoje zadatke za kratko vreme i koristeći minimalne resurse kao što je memorija.

U ovom zadatku mi (ili računar) možemo pronaći najbolje rešenje tako što ćemo isprobati sve moguće rute, a zatim izabrati rutu sa najmanjim ukupnim vremenom kao rešenje. Ova metoda je poznata kao tehnika grube sile. Ova tehnika je pogodna za probleme sa malim veličinama (u ovom slučaju, broj mlinova i staza koje se može proći). Ako je problem veliki, onda će tehnika grube sile raditi sporo jer postoji toliko mogućih rešenja koja se moraju ispitati.

Stoga, postoje i druge tehnike koje mogu efikasnije pronaći rešenja za probleme, kao što su pohlepni algoritmi i dinamičko programiranje.

Optimization. [https://simple.wikipedia.org/wiki/Optimization_\(computer_science\)](https://simple.wikipedia.org/wiki/Optimization_(computer_science))



Zlatni tiket

Tekst zadatka:

Armadilo Marija, ima poklon kod koji vredi četiri kukuruza. Marija, zajedno sa kapibarom Huanom, odlazi do automata da preuzeme kukuruz.



Dok idu ka automatu, gledaju poklon kod, ali ga ne razumeju.



Kada vide tabelu kodova na automatu, sve postaje jasno.



Pitanje/Izazov:

Koja četiri slova treba da unesu da bi dobili svoj kukuruz?

Ponuđeni odgovori:

- A) R S T F
- B) R S T D
- C) R S X D
- D) R Y T D



Tačan odgovor i objašnjenje:

Tačan odgovor je B.

Zamena svake šifre na karti sa slovom:

oo ..o	R
o. ..o	S
o. o.o	T
.o ooo	D

Informatička pozadina:

Tabela kodova koja se pojavljuje na automatu je replika Baudotovog koda. Kod Bautačke je binarni kod koji koristi krstove i tačke. Bodoov kod je kodiranje znakova koje je za telegrafiju izmislio Emil Bodo 1870-ih. Korišćen je za teleprinterske poruke umesto Morzeove azbuke.

U kriptografiji, ova vrsta kodne tabele se naziva ključ. Ključ se može koristiti za šifrovanje pisane poruke u formatu koji je nemoguće pročitati bez ključa. Pomoću ključa, poruka se može dekodirati nazad u čitljiv format. Šifrovanje se koristi, na primer, za digitalne potpise, verifikaciju radi zaštite privatnosti podataka, pregledanje Interneta i poverljivu komunikaciju kao što su transakcije kreditnim karticama i e-pošta.

Bodotov kod je prvobitno koristila Klementina, koji je bio prvi računar korišćen u naučne svrhe koji je stigao u Argentinu 1961. godine i nalazio se u Paviljonu i Univerziteta u Buenos Ajresu. Uputstva su davana rolni perforiranog papira i rezultati su vraćeni na isti način.

Baudot code, encoding, encryption, cryptography.

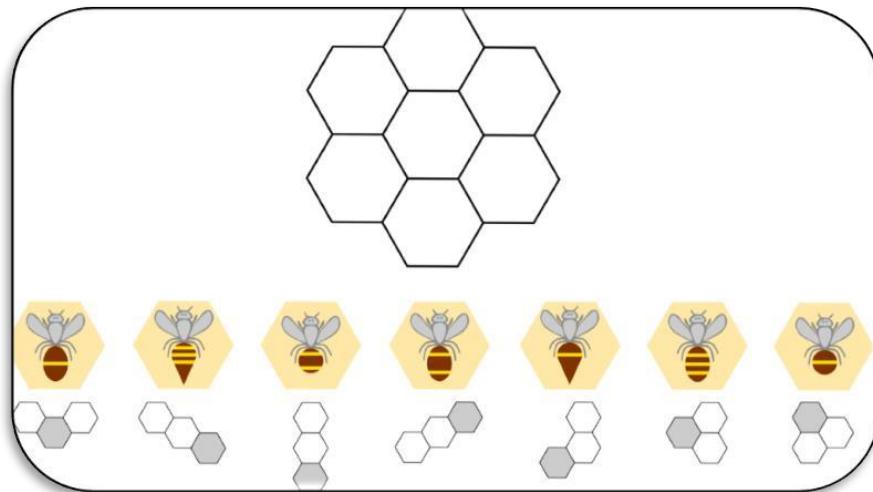
https://en.wikipedia.org/wiki/Baudot_code <https://en.wikipedia.org/wiki/Cryptography>
[https://es.wikipedia.org/wiki/Clementina_\(computadora\)_\(in Spanish\)](https://es.wikipedia.org/wiki/Clementina_(computadora)_(in_Spanish))



Košnica 2

Tekst zadatka:

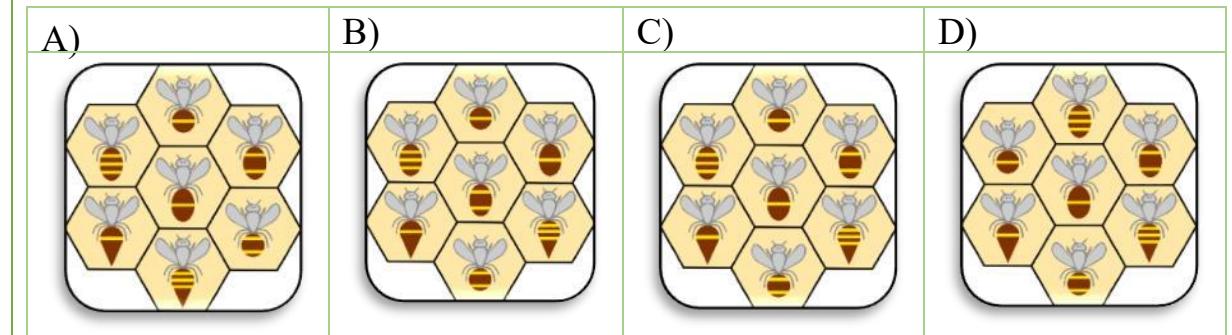
Dabru pčelaru je potrebna pomoć da rasporedi pčelice u košnici.



Ispod svake pčelice je prikazano pravilo: Pčelica mora da se smesti u sivo polje.

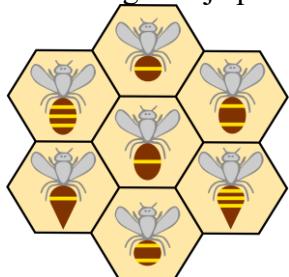
Pitanje/Izazov:

Koji od prikazanih rasporeda pčelica odgovara zadatim pravilima?



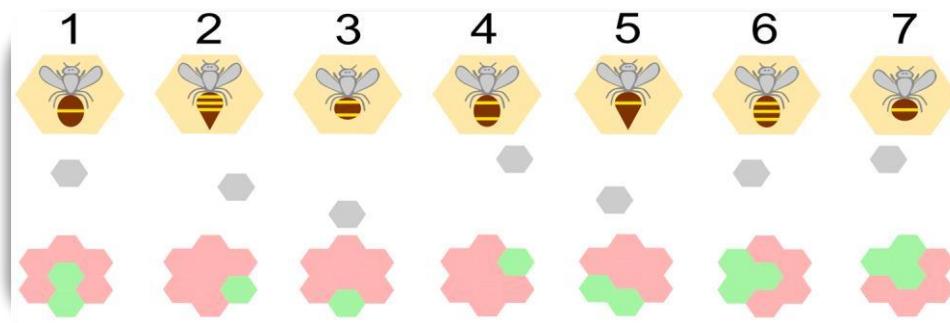
Tačan odgovor i objašnjenje:

Tačan odgovor je pod C).



Zadatak možete rešiti samo tako što ćete ga isprobati. Ali ovo može potrajati mnogo vremena. Da biste pronašli brži način, pogledajte bliže pravila rasporeda pčela. Na sledećoj slici vidite svaku pčelu i njeno pravilo. Ćelije, u koje se pčela može smestiti po svom pravilu, obojene su zelenom bojom.

Vidite da neke pčele mogu biti smeštene u samo jednu ćeliju košnice, a druge u različite ćelije. Na primer, pčela 2 se može staviti samo u jednu ćeliju.



Za rešavanje zadatka postupite ovako: Prvo postavite pčele za koje je moguć samo jedan raspored, odnosno pčele 2, 3 i 4.

Tada ostaje samo jedno moguće mesto za pčele 1 i 5. Na isti način rasporedite pčelu 6 i na kraju pčelu 7.

Informatička pozadina:

U ovom zadatku morate postaviti sedam pčela na sedam različitih mesta. Broj kombinacija za to je veoma veliki. Ako uzmete u obzir pravila, broj mogućnosti je već smanjen, ali i dalje veliki da bi isprobavanje svih njih izazvalo priličnu količinu posla. Ključ za rešavanje zadatka je da se nastavi pravim redosledom koji još više smanjuje broj mogućih rasporeda pčela. U ovom slučaju smo počeli tako što smo se bavili elementima koji su najviše ograničeni kako bismo ograničili broj slučajeva koje treba istražiti.

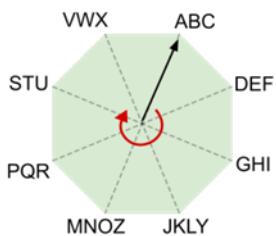
Razmatranja poput ovog zadatka mogu pomoći da se eliminišu problemi koje bi inače bilo nemoguće rešiti.



Šifrovana poruka

Tekst zadatka:

Vrhovi osmougaonika su označeni sa tri ili četiri slova. Na početku šifrovanja nove poruke, strelica uvek pokazuje na slova ABC. Strelica se rotira samo u smeru kazaljke na satu kao što je prikazano na slici. Šifrujemo poruku pomoću strelice i osmougla.



Šifrujemo svako slovo poruke tako da:

- Prvi broj označava za koliko vrhova osmougla strelica treba da se okreće od trenutne pozicije.
- Drugi broj označava poziciju šifrovanog slova u grupi slova na koju pokazuje strelica
- Šifrovana slova su odvojena znakom „-“.

Na primer, poruka „TRI“ je šifrovana nizom 62-73-53

Pitanje/Izazov:

Kako šifrujemo poruku „VODA“?

Ponuđeni odgovori:

- A) 72-53-41-31
- B) 62-53-62-51
- C) 71-43-26-22
- D) 71-53-51-71



Tačan odgovor i obaveštenje:

Tačan odgovor je: 71-53-51-71

Tačan odgovor dobijamo tako što postepeno kreiramo šifrovani tekst za poruku VODA. Šifrovani tekst će sadržati 4 koda:

- Na početku šifrovanja, strelica pokazuje grupu slova ABC. Slovo V je unutar grupe slova na koje će strelica pokazivati nakon rotacije 7 vrhova. Slovo V je prvo slovo u grupi slova VVKS, tako da je prvi kod 71.
- Drugo slovo šifrovane poruke je O. Ovo slovo je u slovnoj grupi MNOZ, na koju će strelica pokazivati nakon što se pet puta okrene sa svoje trenutne pozicije u grupi slova VVKS. O je na trećem mestu, tako da je drugi kod 53.
- Do grupe slova koja sadrži D dolazimo okretanjem 5 vrhova i slovo D je prvo slovo, pa je treći kod 51.
- Slovo A je u grupi slova na koju će strelica pokazivati nakon što strelica rotira 7 vrhova i A je prvo slovo u grupi, tako da je kod 71. Ceo šifrovani tekst je 71-53-51-71

Informatička pozadina:

Jedan od načina zaštite podataka od neovlašćenih lica je šifrovanje. Kriptografija je započela svoj razvoj pre 3500 godina. Jedna od najjednostavnijih metoda šifrovanja je zamena svakog znaka drugim karakterom.

U našem zadatku je osmišljen metod šifrovanja koji zbog rotacije strelice daje različitu šifrovani poruku za slovo u zavisnosti od toga koje je slovo ispred tog slova u reči. Dakle, svaki kod zavisi od toga na koju grupu slova je strelica prethodno pokazivala i, prema tome, koliko strelica mora da se okrene da bi pokazala na novu grupu slova za prvo slovo reči koja se šifruje. Iako preostala slova reči imaju iste kodove, postoji izvesna varijabilnost u kodu dizajniranom na ovaj način i stoga ga je teže otkriti. Međutim, ovako dizajniran kod je dovoljno jednostavan da čoveku nije problem da ga zapamti.

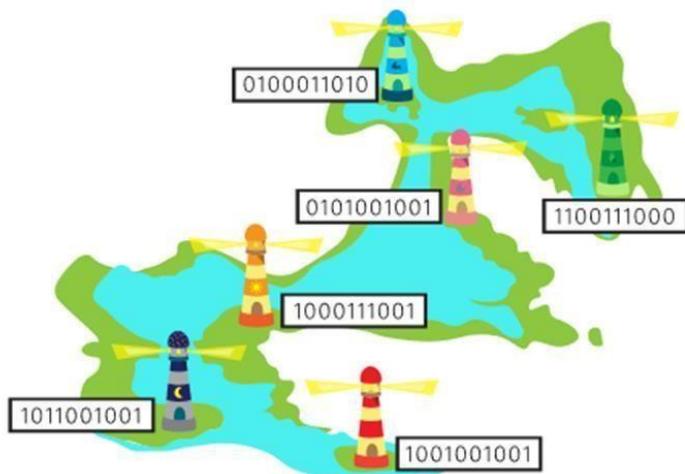
U ovom problemu, ako promenimo slova u svakom tjemenu ili ako ih stavimo u različite trojke, dobili bismo druge metode šifrovanja. Kriptoanalitičari koji pokušavaju da razbiju kodove mogu imati problema sa dešifrovanjem naše



Programerski zaliv

Tekst zadatka:

U Programerskom zalivu, svetla na svakom svetioniku se uključuju i isključuju na jedinstven način. Svaki svetionik ima svoj jedinstveni kod koji opisuje kako se svetlo uključuje i isključuje, tako da pomorci mogu lako da kažu u koji svetionik gledaju.



Na primer, ako svetionik ima kod "1011001001", kod opisuje da se tokom prvih 10 sekundi rada, svake sekunde svetlo uključuje i isključuje na sledeći način: uključeno, isključeno, uključeno, uključeno, isključeno, isključeno, uključeno, isključeno, uključeno, uključeno.

Svakih 10 sekundi svetlo na svetioniku se uključuje i isključuje na isti način.

Pitanje/Izazov:

Ako svi svetionici počnu da rade u isto vreme, u kojoj sekundi će se svi svetionici prvi put uključiti istovremeno?

Ponudeni odgovori:

- A) 5
- B) 6
- C) 7
- D) 8



Tačan odgovor i objašnjenje:

Ako napišemo kodove svetionika jedan ispod drugog, lako ćemo videti da se u svim kodovima nalazi 1 na sedmom mestu: (1 određuje da će svetlo biti upaljeno) u sedmoj sekundi (pročitajte cifre s leva nadesno):

0100011010
0101001001
1100111000
1000111001
1011001001
1001001001

Informatička pozadina:

U računarskoj nauci često kodiramo određene događaje ili stanja ciframa, slovima, brojevima u različitim sistemima brojeva itd. Na primer, ako se neki događaj desi, možemo da dodelimo odgovarajuću vrednost promenljivoj u našem računarskom programu da bismo označili da je događaj je došlo ili je stanje dostignuto. Kasnije, ispitivanjem trenutne vrednosti ove promenljive u našem programu, biće izvršene odgovarajuće grane programa.



Fabrika robota

Tekst zadatka:

Fabrika u Dabrogradu proizvodi robote koji mogu da obavljaju kućne poslove. Robot se sastoji od tri dela: glave, tela i nogu. Neki delovi robota mogu biti različitih oblika. Postoje 4 različita oblika za telo i tri različita oblika za glavu i noge kao što je prikazano u tabeli ispod. Mašina pravi robote, birajući redom delove (glava, telo, noge) kao na slici ispod:

GLAVA	plava	zelena	narandžasta	(ponovljaj opet)
TELO	krug	kvadrat	trougao	petougaon (ponovljaj opet)
NOGE	dve noge	četiri noge	jedna noge (ponovljaj opet)	

Svakog dana mašina pravi robote na isti način prema tabeli: prvi robot ima plavu glavu, okruglo telo i dve noge...

Sinoć je u fabrici nestalo struje, zbog čega se mašina pokvarila. pa je ujutru počela da radi na sledeći način:

- prvog robota je napravila ispravno, a zatim je krenula po sledećim pravilima:
 - za glavu i noge robota – preskače jedan deo unapred u tabeli,
 - za telo robota – preskače dva dela unapred u tabeli.

Dakle, drugi robot sada izgleda ovako:

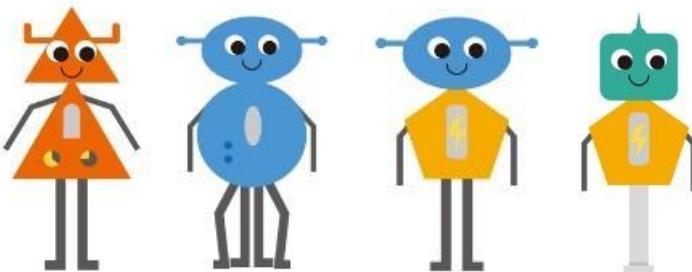


Pitanje/Izazov:

Radnici u fabrici su shvatili da je mašina pokvarena i isključili su je, ali tek nakon što je napravljen deseti robot. Kako je izgledao taj deseti robot? Odaberi tačan odgovor!



Ponuđeni odgovori:



Tačan odgovor i objašnjenje:



plava glava, petougaono telo, dve noge.

OBJAŠNJENJE

Prvi robot je i dalje isti - plava glava, okruglo telo i dve noge.

Glava: Mašina je odabrala svaku drugu glavu na listi. Dakle, novi obrazac je bio – plava → narandžasta → zelena (ponavlja se ponovo). Deseti robot bi imao plavu glavu. 

Telo: Mašina je izabrala svaki treći deo tela na listi. Dakle, novi obrazac je bio – krug → petougao → trougao → kvadrat (ponovno). Deseti robot bi imao petougaono telo. 

Noge: Mašina je odabrala svaki drugi set nogu na listi. Dakle, danas je obrazac bio – dve noge → jedna noga → četiri noge (ponavlja se ponovo). Deseti

robot bi imao dve noge.



Informatička pozadina:

U računarskom razmišljanju, koncepti prepoznavanja obrazaca i algoritama su važni. Prepoznavanje obrazaca je metoda koja se koristi u analizi podataka gde se izdvajaju sličnosti između objekata ili skupova podataka, što nam omogućava da analiziramo trendove ili predvidimo buduće podatke. Algoritmi su skupovi pravila ili uputstava koja objašnjavaju kako proces treba da se izvede. U ovom zadatku, mašina u fabrici prati postavljeni obrazac/algoritam prilikom sklapanja delova tela robota. Računar radi na sličan način – sledi data uputstva da bi proizveo rezultat. Ako se pravila/obrazac promene, promeniće se i krajnji rezultat.

Kao i u stvarnim situacijama, moguće je da algoritmi ili procesi unutar računara budu poremećeni. Iako ovo može biti problem koji korisnik može primetiti tek nakon nekog vremena, postoje koraci koje računari preduzimaju da pokušaju da otkriju kada se ove greške pojave. Jedan primer je provera parnosti prilikom prenosa podataka, koja proverava da li podaci imaju paran ili neparan broj od 1bita. Na osnovu ovoga se zaključuje da li je došlo do promene u toku prenosa podataka (1 na 0 ili 0 na 1). Ovaj metod može otkriti samo promenu u jednom bitu (jedna greška). Zbog toga računari koriste druge, složenije metode provere grešaka prilikom prenosa podataka.



Kutija sa blagom

Tekst zadatka:

Dabrica Ivana pronašla je zaključanu kutiju sa skrivenim blagom. Ivana želi da sazna šta se nalazi u kutiji, ali da bi otključala kutiju treba da zna tačnu kombinaciju tri različita oblika.

Prateći opise desno od svake ponuđene kombinacije, pomozite Ivani da otkrije ispravnu kombinaciju.



1. Jedan oblik je pravilno odabran i dobro postavljen.



2. Nema pravilno odabranih oblika.



3. Dva oblika su pravilno odabrana, ali su na pogrešnim mestima.



4. Jedan oblik je ispravno odabran, ali je na pogrešnom mestu.

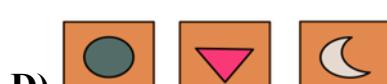


5. Jedan oblik je ispravno odabran, ali je na pogrešnom mestu.

Pitanje/Izazov:

Kojom kombinacijom će Ivana otključati kutiju?

Ponuđeni odgovori:



Tačan odgovor i objašnjenje:

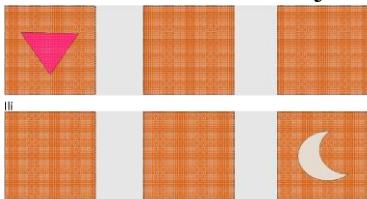
Počećemo da objašnjavamo tačan odgovor tako što ćemo eliminisati oblike koji ne odgovaraju kombinaciji da bismo otključali tajnu kutiju.

Za obrasce iz drugog reda navodi se da nijedan od njih ne pripada traženoj kombinaciji. Dakle, možemo odbaciti te oblike, jelku , Dijamant i Strelu , jer sa njima sigurno nećemo otključati kutiju.

U poslednjem redu opisa kombinacija (u tekstu zadatka) vidimo da je jedan od prikazanih oblika pravilno izabran, ali je postavljen na pogrešno mesto. Prethodno smo već zaključili da jelka i Strelica sigurno ne pripadaju traženoj kombinaciji pa sledi da je upravo oblik Zvezdica oblik koji treba smestiti na neko drugo mesto. Dakle mogući položaji Zvezdice su sledeći:



Nastavljamo da tražimo preostala dva oblika. Iz prvog reda opisa kombinacija (iz teksta zadatka) vidimo da je jedan oblik pravilno izabran i postavljen na odgovarajuće mesto. Oblik strelice odbacujemo jer smo već zaključili da ne pripada traženoj kombinaciji, pa proizilazi da je jedna od sledećih kombinacija ispravna:

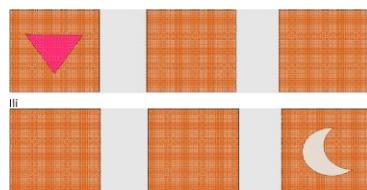


Iz trećeg opisa kombinacija (iz teksta zadatka) vidimo da su dva oblika pravilno izabrana, ali pogrešno postavljena. Svakako da je Zvezda deo naše kombinacije, ali zaključujemo da njena pozicija neće biti u sredini, odnosno Zvezda mora biti na prvom mestu, kao što je prikazano na sledećoj slici:



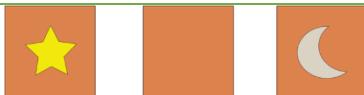
Dakle, pronašli smo odgovarajuću poziciju za jedan oblik. Nastavljamo da tražimo preostala dva oblika i njihove pozicije.

Iz prvog reda opisa (iz teksta zadatka) vidimo da na prvom mestu mora biti oblik Trougla ili na poslednjem mestu oblik Meseca.



S obzirom da smo već zaključili da je oblik Zvezde na prvoj poziciji, oblik Meseca mora biti na poslednjoj poziciji, tako da naša željena kombinacija trenutno izgleda ovako:





Iz četvrtog opisa (iz teksta zadatka) vidi se da jedan oblik u kombinaciji tačan, ali se nalazi na pogrešnom mestu. Već smo zaključili da Trougao nije deo tražene kombinacije, pa je jedino preostalo mesto u kombinaciji drugo (srednje) mesto. Oblik Srce ne može biti oblik koji je deo tražene kombinacije, jer nalazi se u sredini, pa sledi da se u sredini mora smestiti Krug .



Informatička pozadina:

U informatici je često potrebno posmatrati odabrani skup znakova/simbola da bi se otkrilo da li posmatrani znakovi/simboli pripadaju nekom zajedničkom obrascu.

U našem zadatku pratili smo opise kombinacija i kreirali šablon. Svaki put kada napišemo ispravan obrazac, odnosno rasporedimo odgovarajuće oblike u ispravnom redosledu, kutija sa blagom će biti otključana.

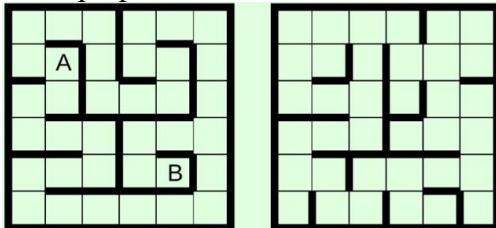
U stvarnom životu, prepoznavanje/podudaranje obrazaca se koristi, na primer, kada se proverava ispravnost napisanih reči.



Lavirint

Tekst zadatka:

Mala Dabrica je u lavirintu. Lavirint je napravljen na dva sprata, svaki sprat sa svojim rasporedom prepreka, kao na slici.



Mala Dabrica može se kretati između susednih celija unutar jednog sprata ako nema prepreke (podebljane linije) između celija; takav potez traje jednu sekundu. Mala Dabrica može takođe koristiti svoj čarobni štapić da bi se pomerila na odgovarajuću celiju drugog sprata; ovo traje pet sekundi.

Na primer, ako je Mala Dabrica u celiji A, postoje tri moguća poteza:

1. Potez levo. Ovaj potez traje 1 sekundu.
2. Potez dole. Ovaj potez traje 1 sekundu.
3. Potez na odgovarajuću celiju drugog sprata. Ovaj potez traje 5 sekundi. Mala Dabrica počinje iz celije A i želi doći do celije B što je pre moguće.

Pitanje/Izazov:

Koje je najkraće vreme potrebno Maloj Dabrići da dođe do celije B, ako počinje iz celije A?

Ponuđeni odgovori:

- A. 16
- B. 17
- C. 18
- D. 20



Tačan odgovor i objašnjenje:

Tačan odgovor je C.

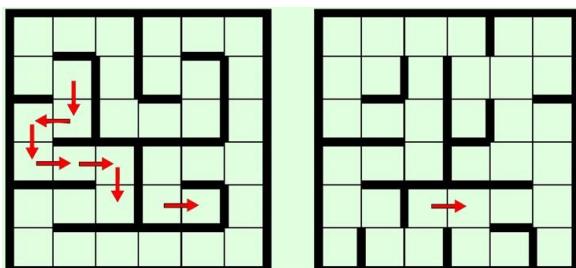
Dati problem je problem najkraće putanje. Postoje različiti pristupi da se dobije rešenje, od kojih je jedan da se primeni Dijkstrin algoritam za nalaženje najkraće putanje od A do B.

Slika ispod pokazuje dužine optimalnih putanja do svih ćelija ako se podje iz ćelije A.

2	3	4	11	12	13
1	0	5	10	9	14
2	1	6	7	8	15
3	4	5	18	17	16
8	7	6	17	18	15
9	10	11	12	13	14

7	6	7	8	11	12
6	5	8	9	10	11
7	6	7	10	11	12
8	9	8	13	12	13
9	10	11	12	13	14
10	11	12	13	14	15

Može se videti da je dužina najkraće putanje 18. Jedna od mogućih optimalnih putanja je sledeća:



Odgovor D (20) odgovara optimalnoj putanji kada se kreće samo unutar jednog sprata. Odgovor A (16) odgovara donjoj granici vremena da se dođe do B iz A ako se ide preko drugog sprata i pod pretpostavkom da nema prepreka (tj. [Menhetni rastojanje od A do B] + [vreme za prelaz između spratova]).

Informatička pozadina:

Problem najkraće putanje je dobro proučen problem teorije grafova. Njegove primene obuhvataju pronalaženje optimalne putanje u računarskoj mreži, na mapi grada itd.

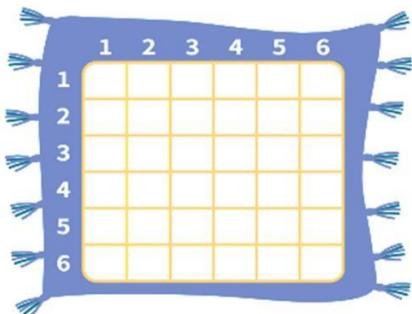
Primer problema najkraće putanje u mrežnom grafu labyrintha ima razne primene, npr. vođenje žica u dizajnu integrisanih kola visoke integracije (VLSI) (tzv. „trkač kroz labyrin“). U datom problemu kretanje između spratova zahteva više vremena u poređenju sa kretanjem po istom spratu (5 prema 1). Ovo odgovara činjenici da je u višeslojnim VLSI integrisanim kolima ožičenje između različitih nivoa mnogo skuplje nego ožičenje unutar istog nivoa. Obično su VLSI integrisana kola postavljena na 10-onivovskom kristalu silicijuma, dok ovaj zadatak uvodi samo dvoslojni problem. Štaviše, projektovanje VLSI integrisanih kola zahteva određivanje putanje miliona i miliardu žica za povezivanje tranzistora, ulaza, izlaza i memorijskih ćelija da bi se dobilo VLSI integrisano kolo za digitalne uređaje.



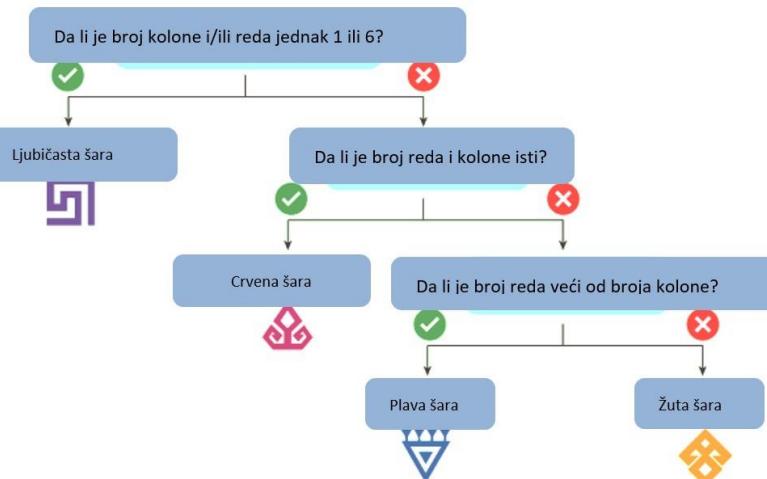
Tkanje čilima

Tekst zadatka:

Azra je tkalja čilima iz Turske. Ona tka čilim oblika kvadrata, koji se sastoji od 6 kolona i 6 redova.



Ona stavlja simbol na svaki kvadrat čilima koristeći sledeća pravila:

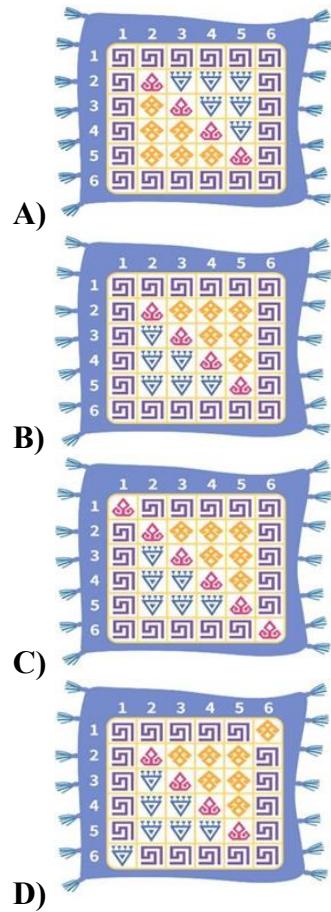


Pitanje/Izazov:

Koristeći ovaj metod, kako će izgledati gotov čilim?



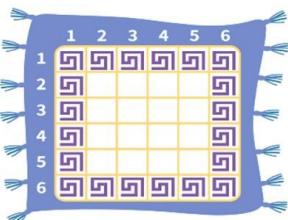
Ponuđeni odgovori:



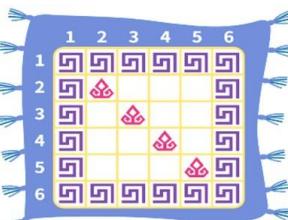
Tačan odgovor i objašnjenje:

Tačan odgovor je B.

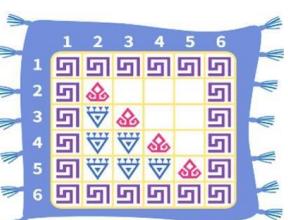
Prvo pitanje koje Azra postavlja, dovodi do toga da prvi i šesti red, kao i prva i šesta kolona treba da imaju ljubičasti oblik. To daje sledeći raspored:



Drugim pitanjem se određuje da će po dijagonali biti crveni simbol. Tako dobijamo sledeći raspored simbola:



Sledećim pitanjem izdvajamo sve redove čiji je broj veći od broja kolone, i tako plavom šarom popunjavamo levi deo kvadrata ispod dijagonale.



Na kraju, preostali kvadrati desno od dijagonale imaju brojeve redova koji nisu veći od broja kolona, i popunjavaju se žutim simbolom, pa dobijamo tačan odgovor.

Informatička pozadina:

Drvo grananja je jednostavan način da se prikaže struktura algoritma koji sadrže IF/ELSE izjave. Ovo znači da svaki red drveta sadrži pitanje ili izjavu i ako je tačno, problem se rešava jednom granom drveta, a ako nije, drugom granom. To se nastavlja do izlaska iz algoritma.



Jagode

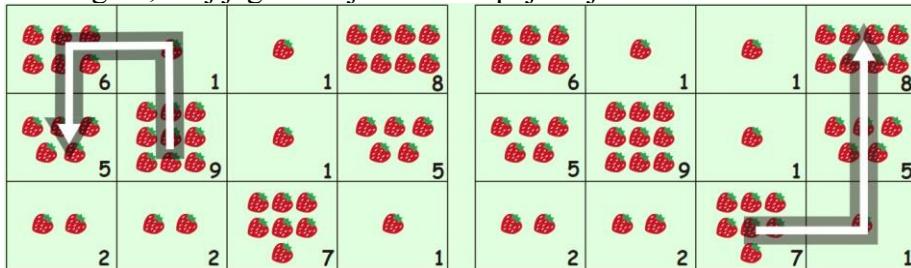
Tekst zadatka:

Na poljima su jagode, a dabrovi ih obožavaju. Međutim, dabar može da stane na neko polje, a zatim sa tog polja može napraviti svega još tri koraka (može da ide gore, dole, levo i desno). Dabar će pojesti sve jagode sa polja sa kog je krenuo i sa svih ostalih polja na kojima je bio.

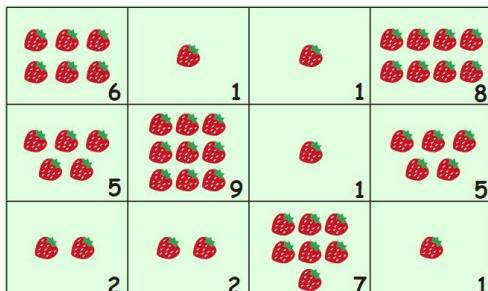
Na slikama su prikazana dva primera:

Na prvom, broj jagoda koje će dabar pojesti je $9 + 1 + 6 + 5 = 21$.

Na drugom, broj jagoda koje će dabar pojesti je $7 + 1 + 5 + 8 = 21$.



Mali dabar se kreće po sledećem polju:



Pitanje/Izazov:

Koliko najviše jagoda može da pojede?

Ponuđeni odgovori:

- A) 20 B) 21 C) 22 D) 23

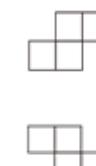


Tačan odgovor i objašnjenje:

Tačan odgovor je 23.

Iako postoji 51 različit način da se naprave 3 koraka na zadatom polju, ne moramo računati broj pojedenih jagoda za svaki način.

Kako se mladi dabar može kretati samo po 4 susedna polja (praveći 3 koraka), mogući oblici puta kretanja su:

Oblik	Kvadrat	Linija	L oblik	Horizontalni L oblik	Z oblik	Vertikalni Z oblik
Izgled puta				   	 	 
Broj mogućnosti	6	3	12	16	8	6

Ispitujući sva moguća kretanja, maksimalan broj jagoda se može postići samo u slučajevima ako staza kretanja obuhvata polja na kojima se nalazi 8 i 9 jagoda. Tako se dolazi do maksimalnog broja jagoda od 23.



	Number of strawberries Little Beaver can eat on a path		
	including the field '9'	including the field '8'	largest of the two
square 	$9+5+6+1=21$	$8+5+1+1=15$	21
line 	$5+9+1+5=20$	$6+1+1+8=16$	20
L shape 	$7+2+9+1=19$	$8+5+1+7=21$	21
Horizontal L shape 	$9+1+5+8=23$	$9+1+5+8=23$	23
Z shape 	$5+9+2+7=23$	$8+1+1+9=19$	23
Horizontal Z shape 	$6+5+9+2=22$	$8+5+1+7=21$	22

Informatička pozadina:

U računarskoj nauci, mnogi problemi imaju za cilj pronalaženje maksimalne ili minimalne vrednosti, kao što je raditi najviše posla, trošiti najmanje novca ili – u ovom Bebrasovom zadatku – pojeti najviše jagoda. U ovim problemima postoji mnogo opcija da se uradi posao, da se potroši novac ili da se pojedu jagode. U okviru ovih opcija, izazov je pronaći one sa maksimalnim ili minimalnim vrednostima; nazivaju se optimalnim rešenjem.

Obično postoji mnogo načina da se pronađe optimalno rešenje. Jedan od njih je tako što ćete navesti sve moguće opcije, izračunati njihove vrednosti jednu po jednu i izabrati one sa najboljim vrednostima. Ovu metodu zovemo brute-force pretraga ili iscrpna pretraga.

Međutim, ovaj metod može oduzeti ogromno vreme računara ako postoji mnogo opcija. Stoga je korisno analizirati problem i pronaći ograničenja koja mogu smanjiti broj opcija i brže pronaći optimalno rešenje. Za mnoge probleme kompjuterski naučnici znaju metode za efikasno pronalaženje optimalnih rešenja. Ali za druge probleme je poznato da se ne mogu efikasno rešiti.



Baza podataka o dabrovima

Tekst zadatka:



U selu dabrova živi desetak porodica. Informatičar Jelenko kreirao je bazu podataka seljana, beležeći podatke o svakom dabru u obliku 16-bitne binarne reči od bita b15 (levo) do bita b0 (desno), na sledeći način:

b15 _ b12: četiri bita za porodični broj;

b11: jedan bit za pol (0 = ženski, 1 = muški);

b10 _ b4: sedam bitova za masu (prirodni brojkilograma);

b3 _ b2: dva bita za „vrednog radnika“ (00 = izgradnja kuće, 01 = izgradnja brana, 10 = skladište hrane, 11 = obrazovanje mladih dabrova);

b1 _ b0: dva komada za omiljenu hranu (00 = kora drveta, 01 = vodene biljke, 10 = trava, 11 = šaš).

Na primer, binarna reč 0100 0 0100101 10 01 označava da dabar pripada porodici 4, da je ženka, da ima masu 37 kilograma, da je kvalifikovan radnik u skladištu hrane i voli vodene biljke.

Pitanje/Izazov:

Dabar Jelenko postavlja upite bazi podataka formulisanjem Bulovih izraza (u dvovalentnoj logici: 0 = netačno, 1 = tačno). Koji skup dabrova označava sledeći izraz?

b11 and not (b10) and b9 and b7 and (not (b3) or not (b2))

Ponuđeni odgovori:

- A. Ženke težine najmanje 16 kilogram, vredni radnici u skladištu hrane.
- B. Muškarci težine najmanje 64 kilograma, vredni radnici na izgradnji kuća ili brana.
- C. Muškarci težine od 39 do 63 kilograma, vredni radnik u izgradnji (nečeg) ili vredni radnik u skladištu za hranu.
- D. Muškarci čija masa je najviše 39 kilograma, vredni radnik na izgradnji brana.



Tačan odgovor i objašnjenje:

Tačan odgovor je 3, pošto: b11 = 1 označava „muški pol“, b10 = 0 označava „maksimalna masa 63 kilograma“, b9 = 1 i b7 = 1 znači „masa najmanje (32 + 8) kilograma“, i na kraju b3 = 0 ili b2 = 0 isključuje samo “vredan radnik u obrazovanju mladih dabrova” (b3 = 1 i b2 = 1).

Imajte na umu da je prema De Morganovim zakonima iskaz (not (b3) or not (b2)) je ekvivalentan iskazu not (b3 and b2).

Ponuđeni odgovor 1 nije tačan: b11 = 1 označava “muški pol”, ali odgovor se odnosi na ženski pol.

Ponuđeni odgovor 2 nije tačan: b2 = 0 označava “vredan radnik u skladištu hrane ”

Ponuđeni odgovor 4 nije tačan: b3 = 0 označava “vredan radnik u izgradnji kuća”

Informatička pozadina:

Formulacija rešenja opisanog zadatka može se predstaviti iskaznom logikom ili Bulovom algebrrom. Na taj način može se obaviti pretraga jednostavne baze podataka, odnosno može se formulisati „ključ za pretragu“.



Skladišta

Tekst zadatka:

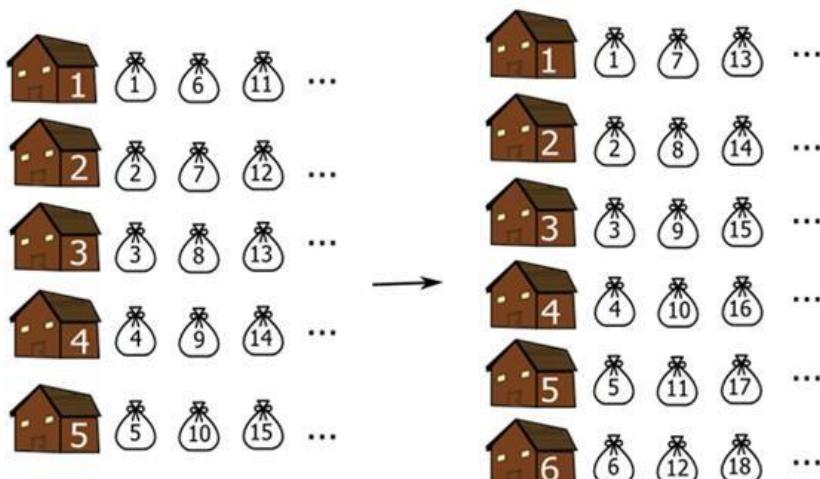
Postoje 42 ježa koji svoje stvari čuvaju u 5 skladišta. Prvi jež čuva svoje stvari u prvom skladištu, drugi u drugom skladištu, ..., šesti ponovo u prvom skladištu itd.

Jednog dana ježevi su izgradili novo skladište i numerisali ga brojem 6. Onda su odlučili da presele stvari između skladišta da bi ponovo uprostili njihovu raspodelu, i to na sledeći način: prvi jež čuva svoje stvari u prvom skladištu, drugi u drugom skladištu, ..., sedmi ponovo u prvom skladitu itd.

Pitanje/Izazov:

Koliko ježeva nije moralo seliti svoje stvari iz jednog u drugo skladište?

Na primer, prvi jež uvek čuva svoje stvari u prvom skladištu, on ne mora da se seli. Ali šesti jež je prvo bio u prvom skladištu, ali se onda preselio u šesto skladište.



Ponuđeni odgovori:

- A.10
- B.11
- C.12
- D.14



Tačan odgovor i objašnjenje:

Tačan odgovor je 10.

Ježevi označeni sa 1, 2, 3, 4, 5 i 31, 32, 33, 34, 35 ne moraju se seliti. Da bismo ovo videli, možemo prvo pojednostaviti brojanje: neka ježevi i skladišta budu numerisani od 0. Tako imamo ježeve od 0 do 41. Jež broj x ne treba da se seli ako je $x \bmod 5 = x \bmod 6$. Ovo je moguće ako $30 \mid x - r$, pri čemu je $r = x \bmod 5 = x \bmod 6$. Zato, ovo je jedino moguće za $x = 0 + 0, 1, 2, 3, 4$ and $x = 30 + 0, 1, 2, 3, 4$.

Informatička pozadina:

Ovo je zadatak koji se odnosi na distribuirano skladištenje resursa: da bi se došlo do sadržaja skladišta može se koristiti strategija koja se naziva hešing (hashing). Po ovom pristupu, svaki resurs se povezuje sa hash ključem koji se može efikasno izračunati; resursi se onda dobijaju izračunavanjem ovog ključa i pretraživanjem samo među onim resursima sa istim hash ključem. Na primer, ako su resursi osobe sa svojim imenima, strategija raspršivanja može koristiti prva slova imena kao hash ključ: ako neko treba da dobije podatke o osobi "Mattia", hash ključ biće "M"; da bi se osoba pronašla, može se pretražiti samo među manjom grupom osoba sa istim hash ključem (na primer "Mattia", "Mathilde" i "Marcello"). U ovom zadatku hash ključ se izračunava korišćenjem funkcije mod (koja predstavlja ostatak celobrojnog deljenja), ali kada se broj skladišta poveća, ključ se menja. Dobre hash funkcije imaju osobinu da se ne menjaju ključevi kada se promeni nešto u strukturi podataka u skladištu, i za njih se kaže da imaju osobinu doslednog heširanja (consistent hashing).



Hamming Leming

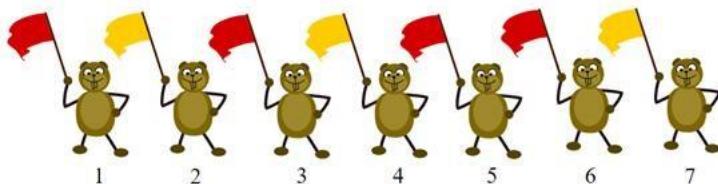
Tekst zadatka:

Kralj dabrica BenBen, želi da pošalje poruku svojoj kraljici Flekici koja živi u drugom zamku. Bira četiri glavna dabra da dostave njegovu poruku i svakom od njih daje zastavicu: crvenu ili žutu, prema poruci koju želi da pošalje. Ali BenBen se brine da bi tokom putovanja moglo nešto da podje po zlu, pa bira još tri dabra koji će kontrolisati glavne dabrice i daje im zastave prema sledećim pravilima:

- Peti dabar prati dabrice br. 1, br. 2 i br. 3: ako je broj crvenih zastavica koje nose neparan, tada će dabar br. 5 nositi crvenu zastavu, inače će nositi žutu zastavu.
- Šesti dabar prati dabrice br. 1, br. 2 i br. 4: ako je broj crvenih zastavica koje nose neparan, onda će dabar br. 6 nositi crvenu zastavu, inače će nositi žutu zastavu.
- A sedmi dabar prati dabrice br. 2, br. 3 i br. 4: ako je broj crvenih zastava koje nose neparan, onda će dabar br. 7 nositi crvenu zastavu, inače će nositi žutu zastavu.

Na putu je, jedan od glavnih dabrica izgubio zastavu. Da bi prikrio svoju krivicu, brzo je napravio novu. Na žalost, on se ne seća boje zastave koju je u početku imao, i nije siguran da li je njegova nova zastava tačna ili ne.

Kada su dabrici stigli u kraljičin zamak, poruka je izgledala ovako:



Pitanje/Izazov:

Tačno jedan glavni dabar je izgubio zastavu, ali ne znamo da li njegova nova zastava ima pravu boju. Jedna od izjava u nastavku je tačna. Koja?

Ponuđeni odgovori:

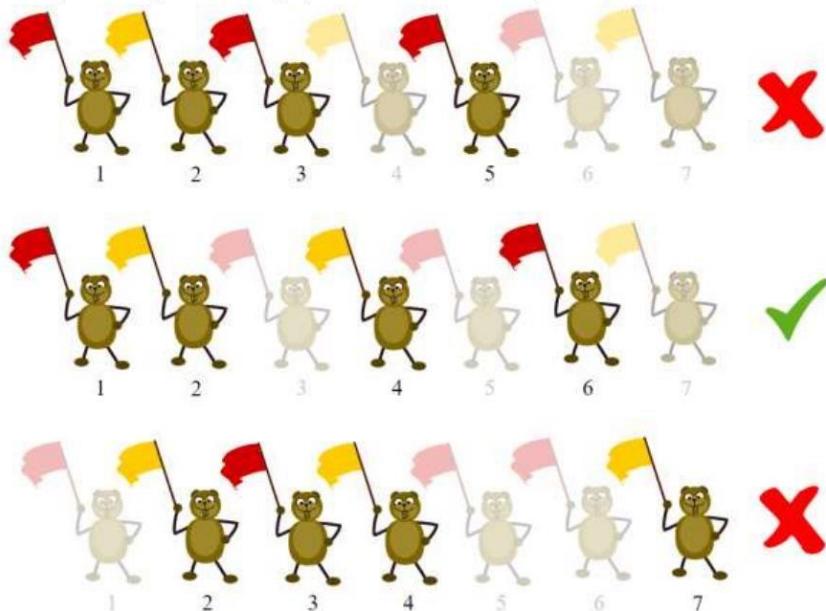
- A. Nova zastava ima pravu boju.
- B. Dabar br. 1 je izgubio zastavu, a nova zastava ima pogrešnu boju.
- C. Dabar br. 2 je izgubio zastavu, a nova zastava ima pogrešnu boju.
- D. Dabar br. 3 je izgubio zastavu, a nova zastava ima pogrešnu boju.



Tačan odgovor i objašnjenje:

Tačan odgovor je D. Tačan odgovor je D. Dabar br. 3 je izgubio zastavu, a nova zastava ima pogrešnu boju. Nazovimo pomoćnika dabara i daborove kojima on pomaže dabrova grupa i označimo ovu grupu brojem dabrovog pomoćnika. Na primer, grupu br. 5 čini dabar br. 5 i daborvi kojima on pomaže, naime daborvi br. 1, br. 2 i br. 3. Iz pravila koja je kralj upotrebljavao da izdaje zastave pomoćnim daborvima, možemo primetiti sledeće: pre početka putovanja, broj crvenih zastava koje je nosila bilo koja grupa trebalo bi da bude paran. Zapravo, prema pravilima, ako je broj nosećih crvenih zastavica dabara br. 1, br. 2 i br. 3 neparan, onda dabar # 5 mora nositi crvenu zastavu, čineći tako ukupan broj crvenih zastavica parnim. Ako je broj crvenih zastavica među daborvima br. 1, br. 2 i br. 3 bio paran, tada je dabar # 5 imao žutu zastavu, zadržavajući ukupan broj crvenih zastavica parnim. Ista osobina važila je za grupu br. 6 i za grupu br. 7.

Sada da proverimo pravila za sve grupe nakon što su dabrovi stigli u kraljičin zamak.



Prvo, vidimo da pravila ne važi za sve grupe. To znači da je dabar koji je izgubio zastavu napravio zastavu pogrešne boje, a ne onu koja mu je uopšte data. Drugo, možemo primetiti da ako je pomoćnik nosio pogrešnu zastavu, samo jedna grupa bi prekršila pravila, odnosno grupa u kojoj je pomagač. Budući da imamo više grupa koje krše pravila, možemo zaključiti da je jedan od dabrova br. 1, br. 2, br. 3 ili br. 4 izgubio zastavu. Već znamo da je dabar koji je izgubio zastavu zamenio svoju originalnu zastavu pogrešnom. Svaka grupa koja sadrži ovaj dabar prekršila bi pravila. Moramo da pronađemo dabara koji je u grupama koje krše pravila (naime, grupe 5 i 6), a nije u grupi koja ne krši pravila (naime, grupa 7). Dabar br. 3 je jedini takav dabar.



Informatička pozadina:

Ovaj zadatak i u njemu opisana pravila za pomoćne dabrove su primer koda za ispravljanje grešaka. Jedan od prvih kodova koji su kreirani radi ispravljanja grešaka je Hamingov kod koji radi prema pravilima sličnim onima koje je kralj dao u ovom zadatku.

Kada šaljemo poruke, obično u binarnom formatu, putem komunikacionih kanala (žice ili radio talasi), zbog šuma koji dolazi iz okoline, uvek postoji šansa da se neki bitovi u poruci preokrenu (promene sa 0 na 1 ili obrnuto).

Obično želimo da možemo da otkrijemo da li je poruka izmenjena i eventualno utvrdimo koja je originalna poruka.

Ovde su korisni kodovi za ispravljanje grešaka.

Jednostavan način da to uradite je slanje svakog bita poruke nekoliko puta. Na primer, ako svaki bit pošaljemo tri puta, i dalje možemo utvrditi kakav je originalni bit, čak i ako je jedna od tri kopije okrenuta. Svaki bit možemo poslati više puta kako bismo zaštitali poruku od još više okretanja. Međutim, ovaj pristup zahteva da pošaljemo tri ili više puta više podataka.

Srećom, mnogi pametni algoritmi omogućavaju slanje što manje dodatnih bitova, a istovremeno omogućavaju proveru i ispravljanje poruke. U ovom zadatku se koristi jedan takav metod. To je primer Hammingovog koda. Hammingovi kodovi se oslanjaju na paritet (parnost ili neparnost) zbiru grupa bitova u poruci, uključujući i dodatne bitove. Ako su pariteti svih grupa tačni, onda možemo pretpostaviti da je poruka stigla netaknuta. Ako su neki od pariteta netačni, tada su neki bitovi (možda i dodatni bitovi) okrenuti tokom prenosa.

Ako znamo da se može preokrenuti samo najviše jedan bit i znamo koje grupe imaju pogrešan paritet, možemo reći koji je bit preokrenut i koja je prvobitna vrednost tog bita.

Svaki kod za ispravljanje grešaka ima svoja ograničenja. Neki kodovi su dizajnirani samo da otkriju da je došlo do promene u poruci bez načina da se utvrdi koji je bit promenjen. Neki kodovi rade samo ako nije promenjena više od određene količine bitova. Na primer, kod u ovom zadatku radi samo ako više od jednog leminga nije pogrešno zamenilo njegovu zastavu. Ako bi dva leminga pogrešno zamenila svoje zastave, kraljica ne bi mogla da utvrdi koje su zastave promenjene. U zadatku smo samo prepostavili da je samo jedan leming pogrešno zamenio svoju zastavu. Da to nismo znali, možda bi slučaj bio da je poruka stigla netaknuta, čak i ako su pariteti pogrešni (na primer, ako su oba leminga br. 5 i br. 7 promenila zastave). Ispravno je pretpostaviti da je manje verovatno da će se okrenuti puno bitova nego samo jedan ili nekoliko, a kada je ta pretpostavka tačna, kodovi za ispravljanje grešaka poput Hammingovih kodova rade dovoljno dobro. Međutim, stvarni komunikacijski sistemi su obično složeniji od toga, a razvijeni su još složeniji kodovi za rešavanje težih situacija.

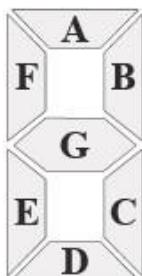
Slični algoritmi se koriste za verifikaciju tačnosti informacija na nekim drugim mestima, kao što su bar kodovi i identifikacioni brojevi na pasošima i ličnim kartama.



Digitalni broj

Tekst zadatka:

Sofija želi da prikaže brojeve pomoću sijalica. Ona koristiti 7 sijalica za prikazivanje brojeva. Sijalice su označene slovima A, B, C, D, E, F i G, kao što je prikazano na slici ispod.



Da bi uključila sijalice, Sofija mora odgovarajuće ćelije u redu tabele da oboji u crveno. Na primer, sledeća tabela prikazuje trocifreni broj 103:

	A	B	C	D	E	F	G
↓	Red	White	Red	Red	White	Red	White
	Red	Red	Red	Red	White	Red	Red



Pitanje/Izazov:

Koji broj će se prikazati ako Sofija kreirala sledeću tabelu?

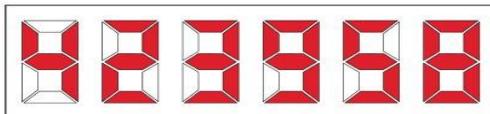
	A	B	C	D	E	F	G
↓	Red	Red	White	Red	White	Red	Red
	Red	Red	Red	Red	White	Red	Red

Ponuđeni odgovori:

- A) 423958
- B) 423598
- C) 425938
- D) 425928



Tačan odgovor i objašnenje:



Iz primera uočavamo da svaki red u tablici predstavlja jedan digitalni broj. Za prvi red u tablici pitanja odabrani su samo segmenti B, C, F i G, koji predstavljaju broj 4. Za drugi red odabrani su segmenti A, B, D, E, G, koji predstavljaju broj 2. Nastavljajući ovaj postupak za svaki red, dobijamo broj 423958.

Informatička pozadina:

Promena prikaza jednog objekta skupom drugih objekata naziva se kodiranje. Kodiranje je u računarskoj nauci vrlo česta stvar. Svaki broj zamenjujemo nizom od sedam ćelija koje mogu imati jednu od dve moguće vrednosti: crvenu ili belu.

Ekrani poput ovih često se koriste u današnjem modernom dobu. Zbog male veličine diode (LED), nekoliko njih se može spojiti u 7-segmentni ekran.

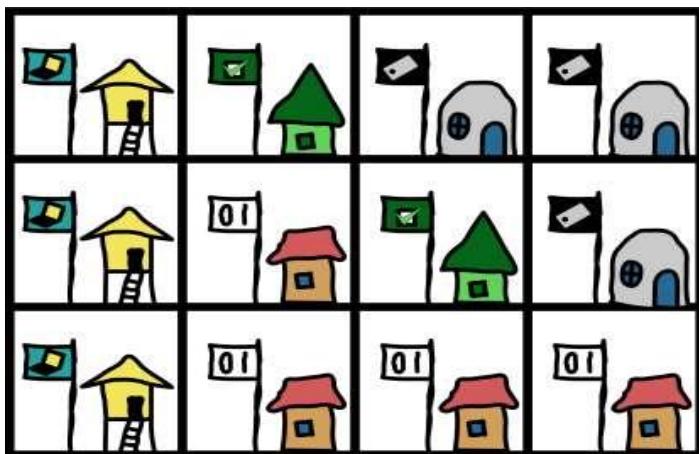
Kad koristimo 7-segmentne ekrane, možemo poslati niz od 7 vrednosti, bilo jedinice (crvene) ili nule (bele), za ABCDEFG gde nula isključuje LED diodu, a jedan uključuje. Na primer, broj "4" na 7segmentnom ekranu opisuje se sa 0110011.



Ujedinjenje plemena

Tekst zadatka:

U dalekoj zemlji živela su četiri plemena koja su se sastojala od nekoliko sela, kao što je prikazano na slici: Binarijanci , Smafonci , Čekboksi i Laptopci. Jednog dana su svi odlučili da se ujedine. Međutim, da ne bi došlo do haosa, odlučeno je da se samo dva plemena mogu ujediniti istovremeno. Vreme potrebno da se dva plemena ujedine u mesecima jednako je broju sela u ta dva plemena. Nakon toga, dva plemena postaju jedno pleme, a proces ujedinjenja se ponavlja sve dok ne ostane samo jedno ujedinjeno pleme.



Pitanje/Izazov:

Koliko je minimalno meseci potrebno kako bi se sva plemena ujedinila?

Ponuđeni odgovori:

- A) 16
- B) 24
- C) 20
- D) 28

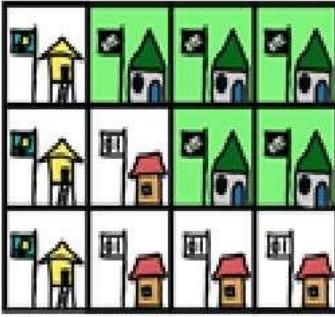
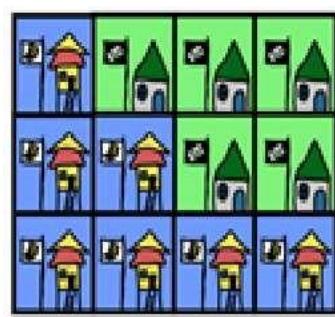
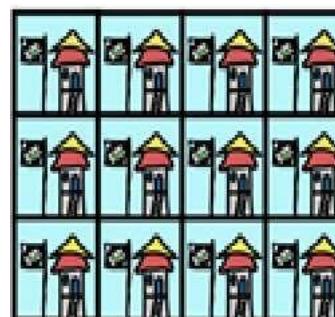


Tačan odgovor i objašnjenje:

Tačan odgovor (B) 24

Optimalna strategija za smanjenje ukupnog broja meseci potrebnih za ujedinjenje svih plemena je minimiziranje broja uključivanja svakog sela u procesu ujedinjenja. Dakle, možemo uočiti da bi najveća plemena trebala biti dotana poslednja, jer će tada najveća količina sela biti dodana najmanji mogući broj puta. Kako bi se to napravilo, svaki korak ujedinjenja trebao bi se provesti s dva plemena s najmanje sela.

To je prikazano u donjoj tablici::

1. Čekboksi imaju najmanji broj sela, pa njih biramo prve za ujedinjenje. Obzirom da postoje dva plemena koja imaju 3 sela, možemo izabrati jedno od njih, na primer, Čekbokse i Smatrfonce. Nakon ujedinjenja možemo ih nazvati Čekfonci.	2. Sada su plemena sa najmanje sela Binarijanci (4) i Laptopsi(3). Nakon ujedinjenja možemo ih nazvati Bintopsi.	3. Na kraju, 5 Čekfonca i 7 Bintopsa se ujedinjuje u jedno veliko pleme.
		

Ovo traje 5 meseci i rezultira sa 3 sela Laptopsa, 4 Binarijanca i 5 Čekfonca.

Ovo traje 7 meseci, a rezultat je 5 Čekfona i 7 Bintops sela.

Ovo traje 12 meseci

Zato je minimalni broj meseci za ujedinjenje sva četiri plemena iz zemlje Dabrovije $5+7+12 = 24$.

Informatička pozadina:

Ovaj izazov je primer problema optimizacije, zadatka čiji je cilj da dođe do strategije koja maksimizira ili minimizira određenu količinu, podložna određenim ograničenjima. Problemi optimizacije su sveprisutni u našem svakodnevnom životu: pronalaženje najkraćeg puta do odredišta, kreiranje rasporeda koji obuhvata najveći broj aktivnosti koje se ne preklapaju, itd. Postoji nekoliko načina za pristup problemu opzimizacije, a oni uključuju "pohlepne" algoritme.

Pohlepni algoritmi počivaju na pretpostavci da će donošenje najboljeg izbora u svakoj fazi (lokalni optimum) rezultirati najboljim konačnim ishodom (globalni optimum). U ovom problemu, ova pretpostavka je zadovoljena: plemena moraju da minimiziraju broj meseci za svako ujedinjenje da bi minimizirala broj meseci za ceo proces ujedinjenja. Međutim, mora se naglasiti da pohlepna paradigma nije univerzalno rešenje za sve vrste problema optimizacije. Ipak, obično pruža pristojnu aproksimaciju u razumnom roku.



Izgubljena slika

Tekst zadatka:

TransArt je kompanija specijalizovana za transport slika. Slike se donose u kompaniju, a zatim ih kuriri prevoze do krajnjeg odredišta.

Slike se ređaju jedna na drugu. Svaka nova pristigla slika stavlja se na vrh. Kurir uzima sliku sa vrha i nosi je na odredište.

Iz bezbednosnih razloga TransArt vodi evidenciju o svim slikama koje se donose i iznose iz kompanije.



Slike donete u kompaniju

Vreme Naziv slike

11:40	“Dabrovi na travi”
12:15	“Srećan dabar
12:55	“Sunce i mesec”
13:30	“Začarana šuma”
14:18	“Hrast i Breza”
15:10	“Močvarna romansa”

Slike odnete iz kompanije

Vreme Kurir

12:25	A
13:35	C
14:35	A
14:40	B
15:20	C
15:35	D



Jedne večeri TransArt-u je rečeno da slika „Sunce i Mesec“ nije stigla do odredišta. Kurir koji je dostavljao sliku je izgubio sliku.

Pitanje/Izazov:

Koji kurir je izgubio sliku „Sunce i Mesec“?

Ponuđeni odgovori:

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

Tačan odgovor i objašnjenje:

Postoje dve važne vrste događaja: neko stavlja sliku na gomilu, a neko uzima sliku sa vrha gomile. Od tabele u zadatku kreiramo novu tabelu koja prikazuje događaje i rezultirajuće stanje gomile slika, sortirano po vremenu.

Vreme	Događaj	Slike na gomili
11:40	Slika "Dabrovi na travi" je doneta u kompaniju	Dabrovi na travi
12:15	Slika "Srećan dabar" je doneta u kompaniju	Srećan dabar Dabrovi na travi
12:25	Kurir A uzima sliku "Srećan dabar"	Dabrovina travi
12:55	Slika "Sunce i mesec" je doneta u kompaniju	Sunce i mesec Dabrovi na travi
13:30	Slika "Začarana šuma" je doneta u kompaniju	Začarana šuma Sunce i mesec Dabrovi na travi
13:35	Kurir C uzima sliku "Zalarana šuma"	Sunce i mesec Dabrovi na travi
14:18	Slika "Hrast i breza" je doneta u kompaniju	Hrast i breza Sunce i mesec Dabrovi na travi



14:35	Kurir A uzima sliku "Hrast i šumu"	Sunce i mesec Davrovi na travi
14:40	Kurir B Uzima sliku "Sunce i Mesec"	Dabrovi na travi
... i ovde možemo stati i zatražiti od kurira B da se javi.		

Informatička pozadina:

U ovom zadatku pojavljuju se tri informatičke ideje.

Jedna je koncept "steak". Slaganje slika nije samo gomila slika, već i struktura podataka, organizovana tako da je poslednji element koji se stavlja na gomilu prvi koji će se preuzeti iz nje („Last In - First Out“ ili skraćeno LIFO).

Drugi je spajanje: da bismo došli do rešenja, morali smo da uzmemo dve sortirane liste (događaje razvrstane po vremenskim oznakama) i spojimo ih u jednu sortiranu listu - onu koju prikazujemo ovde u rešenju. Ovaj korak je osnova jednog od najbržih algoritama za sortiranje podataka, sortiranje spajanjem.

Konačno, čitavu hronologiju možete smatrati izvršenjem programa. Krađa slike je poput događaja koji uzrokuje zaustavljanje pokrenutog računarskog programa („rušenje“). Ovo se naziva izuzetkom. Da bismo pronašli uzrok izuzetka (kurir koji se loše ponaša ili linije u programu), moramo da pratimo izvršavanje programa dok ne dođemo do tačke gde se on ruši. Ovo se zove praćenje. Tada bi programer pokušao da pronađe način kako da postupi sa izuzetkom kako bi sprečio pad programa.



Tajna poruka (Srbija)

Tekst zadatka:



Dabrovi **BenBen** i **Tom** razmenjuju poruke koje se sastoje od **12 cifara 0 i 1**. Pošto Dabar **Boni** razume njihove poruke, odlučili su da ih šifruju. U prvom koraku šifrovanja oni zamenjuju par uzastopnih cifara slovom **A**, **B**, **C** ili **D**:

00	01	10	11
A	B	C	D

U drugom koraku zamenjuju par uzastopnih slova koristeći sistem ispod:

AA	AB	AC	AD	BA	BB	BC	BD	CA	CS	CC	CD	DA	DS	DC	DD
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Na kraju dobijaju konačnu poruku **C13**.

Pitanje/Izazov:

Konačna poruka je **C13**. Koja je bila početna poruka?

Ponuđeni odgovori:

- A) 110000010011
- B) 110010101111
- C) 110000011011
- D) 111011001001



Tačan odgovor i objašnjenje:

Tačan odgovor je A).

Idemo unazad: C postaje DA, 1 postaje AB i 3 postaje AD čime se dobija DAABAD. Sada, kako je D zamenjeno sa 11, A sa 00 i B sa 01, dobijamo konačni odgovor 110000010011, što je odgovor A).

Informatička pozadina:

Kriptografija je nauka koja se bavi metodama očuvanja tajnosti informacija. Kada se poruka prenosi preko otvorenog kanala sa jednog mesta na drugo, svako (u slučaju našeg zadatka Boni) može da je pročita. Da bi se zaštitio njegov sadržaj, on je šifrovan, što ga čini nerazumljivim. Osnovna tehnika sakrivanja poruka je kodiranje, što se dešava u našem zadatku koristeći proces u dva koraka. Pravo šifrovanje je robusnije. Da bi se poruka šifrovala, kodiranje bi trebalo da bude parametrizovano pomoću ključa – zajednička tajna između BenBena i Toma. Moguća parametarizacija može biti način na koji se permutira drugi red svake tabele.

