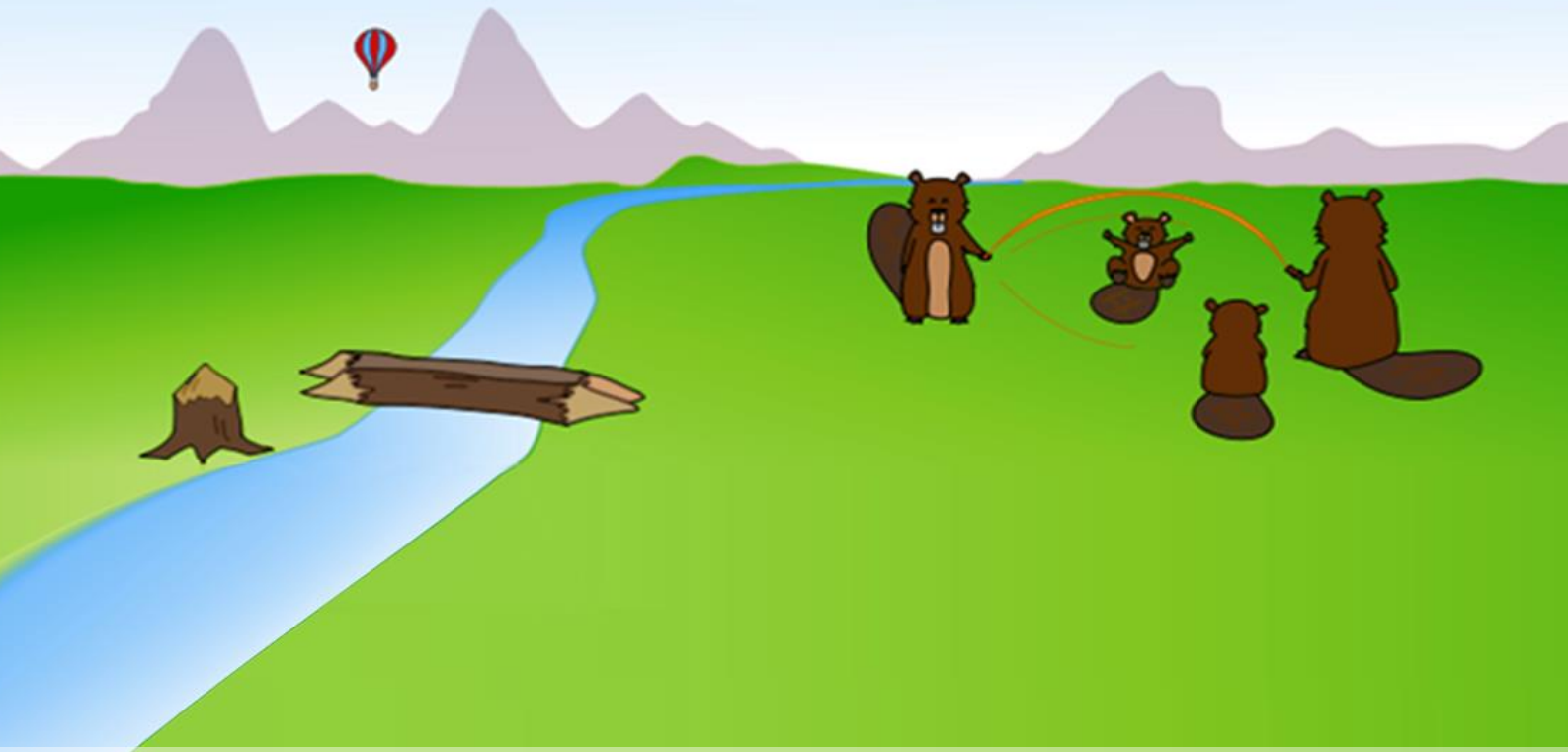


Bober 2016/17

Naloge in rešitve

Naloge za tekmovanje je izbral, prevedel, priredil in oblikoval Programski svet tekmovanja:

Špela Cerar (PeF, Univerza v Ljubljani)
Alenka Kavčič (FRI, Univerza v Ljubljani)
Andreja Filipič (Osnovna šola Spodnja Idrija)
Janez Demšar (FRI, Univerza v Ljubljani)
Matej Črepinšek (FERI, Univerza v Mariboru)
Nataša Kermc (Osnovna šola Brežice)

Razvoj tekmovalnega sistema: Gašper Fele Žorž (FRI, Univerza v Ljubljani)

Kazalo nalog

Nakup zelenjave	5
Vlak Bobopuh	6
Šaman	7
Razbito steklo	9
Polica	10
Zdravstveni dom	11
Ekipni dresi in številke	12
Policisti	13
Barve rož	14
Bobrova koda	15
Čarobni valj	16
Ples	17
Rojstnodnevna torta	18
Skrivna sporočila	19
Vrni se nazaj	20
Igrače iz zamaškov	21
Labirint	22
Sistem cevi	23
Očisti park	25
Nogomet	27
Barvanje	28
Razpored sob	29
Registrske tablice	31
Čarobni napoj	32
Dieta	34
Dvigalo	36
Gasilci	38
Go	40
Kje je zaklad?	42
Mosta	43
Robot	45
Trak	46
Vrni se	47
Vrstni red kuhanja	48
Avtonomni taksi	50
Besedna sestavljanka	52
Igra s frnikolami	53
Načrtovanje poti	55
Pot na večerjo	56
Predor	57
Rokovanje igralcev hurlinga	58
Štirje opravki	59
Večerja	60
Zbirka skodelic	61
KIX koda	63
Padajoče frnikole	64
Prevoz hlodov	66
Sočasni premiki	67
Pravo jutranje zaporedje	69
Zlata frnikola	71
Frankova stikala	73
Opisi zastav	75
Segway	76
Tri v vrsto	77
Prazgodovinska šifra	80
Poteze skakača	82
Rekurzivna umetnost	84
Kopičenje napak	87
Hitro potenciranje	88
Načrt opravil	90
Izračuni	92
Programi za e-pošto	93
Raziskovanje poti	94
Vezalke	95

Vrnitev na začetek 97
Čitalca slik 99
Filter 101
Iskanje tatu 103
Jame 104
Ligra 106
Opica 109
Pranje oblačil 111
Slika v sliki 113
Svetilke 114
Zamenjave 116
Rdeče in modre frnikole 119

Nakup zelenjave

2. – 3. razred

MAMA POŠLJE TOMA V TRGOVINO. KUPIL BO VSO **ZELENJAVO**, KI **NI ORANŽNE ALI RUMENE BARVE**, SAJ TEH DVEH BARV NE MARA.

POMARANČA
11 BEVROV
(SADJE, ORANŽNA)

REDKVICA
2 BEVRA
(ZELENJAVA, RDEČA)

KRUH
17 BEVROV
(KRUH, RJAVA)

BROKOLI
3 BEVRE
(ZELENJAVA, ZELENA)

KLOBASA
19 BEVROV
(MESO, RDEČA)

PAPRIKA
5 BEVROV
(ZELENJAVA, ZELENA)

KORENJE
13 BEVROV
(ZELENJAVA,
ORANŽNA)

REPA
7 BEVROV
(ZELENJAVA, BELA)

KAJ BO KUPIL? KOLIKO BEVROV POTREBUJE?

REŠITEV

KUPIL BO REDKVICO, BROKOLI, PAPRIKO IN REPO. PLAČAL BO 17 BEVROV.

KUPIL PA NE BO POMARANČE, KRUHA IN KLOBASE, KER NISO ZELENJAVA. PRAV TAKO NE BO KUPIL KORENJA, KER JE ZELENJAVA ORANŽNE BARVE.

Računalniško ozadje

Računalniki shranjujejo velike količine podatkov. Za večjo preglednost jih lahko organiziramo v zbirke podatkov. Po zbirkah podatkov lahko iščemo s posebnimi jeziki, v katerih je preprosto opisovati pogoje, kakršne je moral v tej nalogi upoštevati Tom.

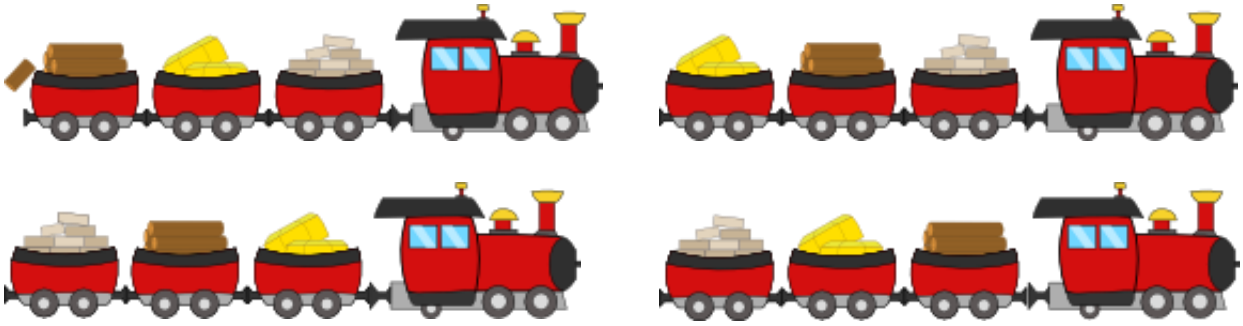
Vlak Bobopuh

2. – 3. razred

VLAK VOZI SKOZI MESTA V TAKŠNEM ZAPOREDJU:

- LESNO MESTO POTREBUJE LES.
- SENENO MESTO POTREBUJE SENO.
- OPEČNO MESTO POTREBUJE OPEKE.

V KAKŠNEM VRSTNEM REDU MORAJO BITI VAGONI, DA BO LAHKO VLAK V VSAKEM MESTU PUSTIL ZADNJI VAGON?



REŠITEV

VAGON, KI JE PRITRJEN ZADNJI, BO PRVI ODKLOPLJEN. NA DRUGI POSTAJI BODO ODKLOPILI DRUGI VAGON S SLAMO. NA TRETJI, ZADNJI POSTAJI, V OPEČNEM MESTU, BODO IZPRAZNILI PRVI VAGON Z ZIDAKI.

Računalniško ozadje

Vlak je potrebno sestaviti po vrstnem redu, ki ga računalnikarji imenujejo zadnji noter prvi ven (angl. Last In, First Out, krajše LIFO). Velikokrat jim pride prav.

Šaman

SLIKAR JE NASLIKAL OSEM RAZLIČNIH SLIK ŠAMANA. NAJBOLJ MU JE VŠEČ TISTA, NA KATERI

- IMA PAPIGO NA SVOJI LEVI ROKI,
- NIMA PALICE IN
- NE MANJKA NOBEN GUMB.

KATERA SLIKA JE TO?

1.

2.

3.

4.

5.

6.

7.

8.

REŠITEV

PRAVILNA REŠITEV JE NA PRVI SLIKI.  .

SLIKE 1, 2, 3, IN7 IZPOLNJUJEJO PRVI POGOJ, DA JE PAPIGA NA ŠAMANOVI LEVI ROKI. NA SLIKI 6 PAPIGA SEDI NA PALICI, KI JO IMA ŠAMAN V ROKI.

SLIKE 1, 3, 4, 5, 7, IN 8 IZPOLNJUJEJO DRUGI POGOJ, DA NIMA PALICE.

SLIKE 1, 2, 4, 5, 6, IN 8 IZPOLNJUJEJO TRETJI POGOJ, DA SO VSI GUMBI NA PLAŠČU ZAPETI.

EDINA SLIKA, KI IZPOLNJUJE VSE TRI POGOJE, JE SLIKA 1.

Računalniško ozadje

Problem v nalogi je povezan z logiko. Pri reševanju moramo uporabiti eno od osnovnih logičnih operacij konjunkcija (IN). Ime je učeno, operacija preprosta: dve stvari sta hkrati res, če je res tako prva kot druga. ☺

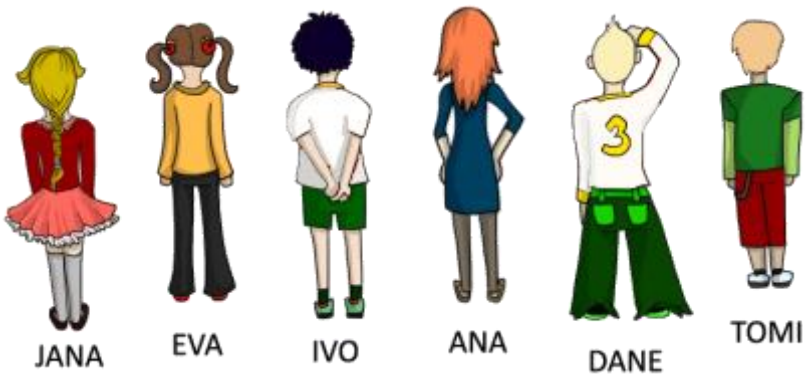
	Slika 1	Slika 2	Slika 3	Slika 4	Slika 5	Slika 6	Slika 7	Slika 8
Papiga na levi roki	✓	✓	✓	×	×	×	✓	×
Nima palice	✓	×	✓	✓	✓	×	✓	✓
Ne manjka gumb	✓	✓	×	✓	✓	✓	×	✓

Kot opazimo, so vsi trije pogoji izpolnjeni le na prvi sliki.

Razbito steklo

2. – 3. razred

EDEN IZMED OTROK JE Z ŽOGO RAZBIL OKNO IN ZBEŽAL. VIDELI SMO LE, DA JE IMEL HLAČE, SVETLO MAJICO IN TEMNE LASE. KDO JE BIL?



REŠITEV

OKNO JE RAZBIL IVO.

SAMO DVA OTROKA NOSITA BELO MAJICO, TO STA IVO IN DANE. OBA NOSITA HLAČE. DANE IMA SVETLE LASE. IVO USTREZA OPISU KRIVCA.

Računalniško ozadje

Naloga je podobna nalogi Šaman, le z bolj zapletenimi pogoji.

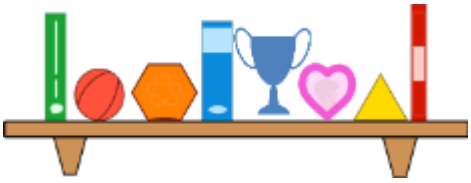
Polica

2. – 5. razred

BETI JE UREDILA SVOJO POLICO TAKO, DA NOBEN PREDMET PRAVOKOTNE OBLIKE NI ZRAVEN DRUGEGA PRAVOKOTNEGA ALI ZRAVEN OKROGLEGA PREDMETA. KATERA POLICA JE NJENA?

A. 

C. 

B. 

D. 

REŠITEV

ODGOVOR D.

NA POLICI A SO PREDMETI PRAVOKOTNE OBLIKE SKUPAJ.

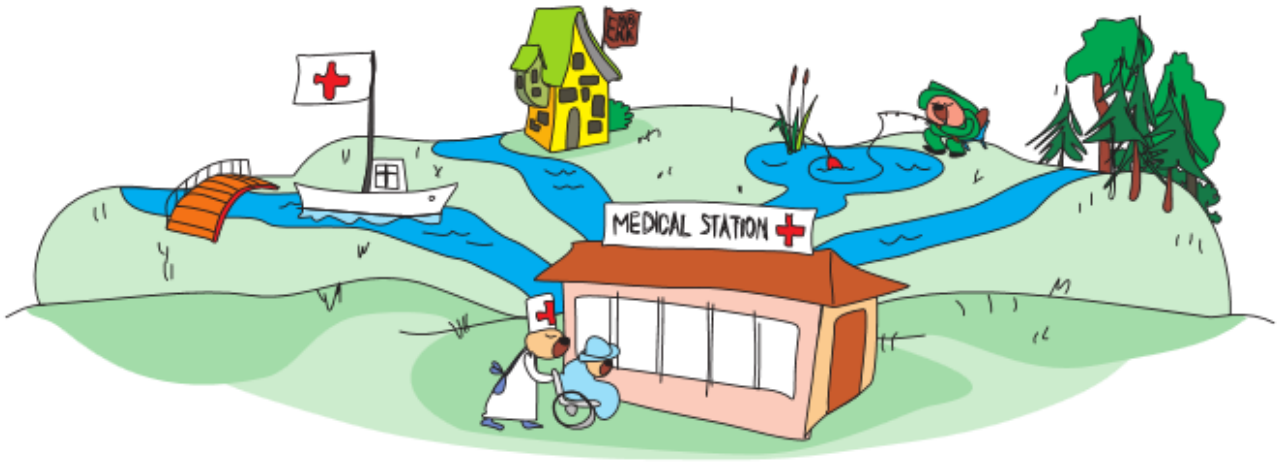
NA POLICAH B IN C JE OKROGEL PREDMET POLEG PRAVOKOTNEGA.

Računalniško ozadje

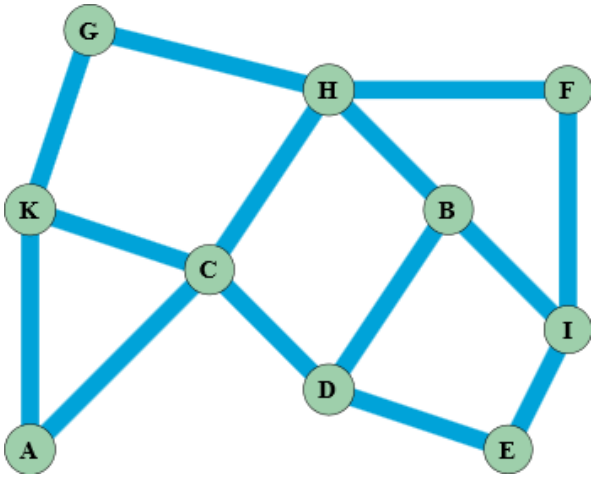
Opisovanje vzorcev in preverjanje, ali določen vzorec ustreza pravilu, je ena od osnovnih veščin v računalništvu. Vzorci so lahko preprosti, kot tale tu, in opisujejo povezave med »resničnimi« stvarmi, lahko pa so tudi veliko abstraktnejši.

Zdravstveni dom

3. – 5. razred



Bobri živijo na križiščih kanalov, označenih s črkami. V mestu bodo zgradili tri zdravstvene domove. Postaviti jih želijo tako, da bodo bobri do najbližjega plavali skozi največ en vodni kanal, ne glede na katerem križišču so. Na katera križišča naj jih postavijo?



Rešitev

Možnih je več pravilnih rešitev, recimo E, H in K.

- Na kraj E lahko plavajo s položaja D, E in I.
- Na kraj H lahko plavajo s položaja B, C, F, G in H.
- Na kraj K lahko plavajo s položaja A, C, G in K.

Druge možne rešitve so: A E H, B I K, C E H, C G I, C H I, C I K in D F K. Reševanja se lahko lotimo, na primer, tako, da naključno izberemo prvi položaj in nato označimo vse kraje, ki so dosegljivi s plavanjem po enem vodnem kanalu. Nato to ponovimo še dvakrat. Če ostane kak kraj nepokrit, rešitev popravljamo...

Računalniško ozadje

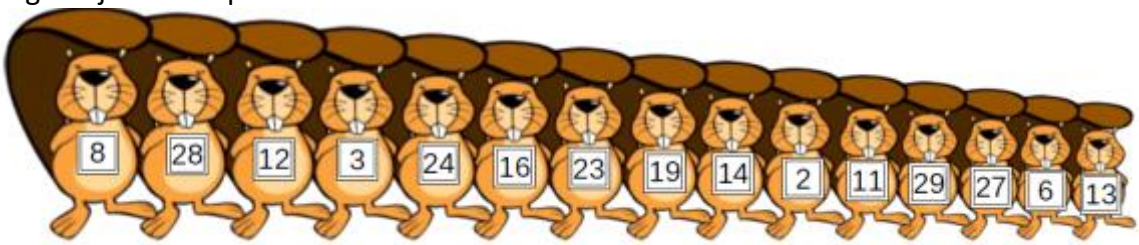
Problem, ki ga opisuje naloga, je eden znanih problemov, za katere računalnikarji še niso odkrili dobrih rešitev. Če bi bilo bobrov več in bi potrebovali več zdravstvenih domov, bi tudi najhitrejši računalniki potrebovali milijarde in milijarde let, preden bi našli rešitev, za katero bi bili lahko prepričani, da zahteva najmanjše možno število zdravstvenih domov.

Ekipni dresi in številke

3. – 5. razred

Prva ekipa se je postavila po vrsti glede na številke na dresih.

Druga se je uredila po velikosti.



Koliko številk, ki so jih uporabili v prvi ekipi, so uporabili tudi v drugi ekipi?

Rešitev

Pravilen odgovor je 3. Številke 14, 19 in 23 se pojavljajo v obeh ekipah.

Računalniško ozadje

Kot si opazil, je iskanje v seznamu, ki je urejen po tistem, kar iščeš (ekipa 1), veliko hitrejše kot v seznamu, ki ni urejen ali pa je urejen po čem, po čemer ne iščeš (ekipa 2, ki je urejena po velikosti).

Nisi prvi, ki je to spoznal. Računalnikarji to vedo že dolgo, zato podatke vedno shranjujejo tako, da so urejeni po lastnosti, po katerih jih bodo preiskovali. Domislili so se celo načinov, na katere lahko shranijo različne vrstne rede za iste podatke. Največji računalniki, ki shranjujejo največje zbirke podatkov (od YouTuba do spletnih iskalnikov) tako porabijo največ časa prav za urejanje in iskanje.

Policisti

3. – 5. razred

Enemu od bobrov je ime Robi.

Bober z obema rokama v zraku stoji zraven Petra.

Andrej stoji zraven bobra brez značke.

Matej nima obeh rok v zraku.

K vsakemu bobru zapiši njegovo ime.




_____ _____ _____ _____

Rešitev

Matej Andrej Robi Peter

Iz druge izjave lahko sklepamo, Peter ni prvi ali tretji bober.
Iz tretje izjave lahko sklepamo, da je Andrej drugi bober.
Iz četrte izjave lahko sklepamo, da je Matej ni tretji bober.
Tako lahko sklepamo, da je Peter četrti bober in Robi je tretji bober.
Končno določimo, da je Matej prvi bober.

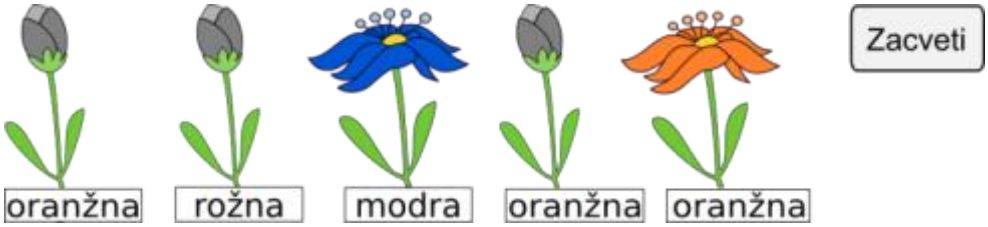
Računalniško ozadje

Logika in logično sklepanje je eno od osnov računalništva.

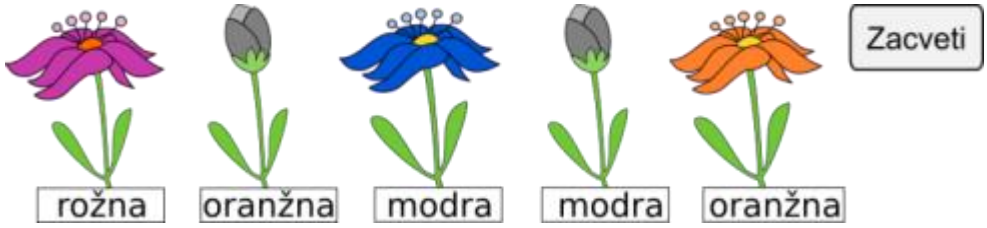
Barve rož

4. – 5. razred

Jana igra igrico, v kateri mora ugibati barve rož. Rože so lahko oranžne, rožne ali modre. Pod vsako mora napisati barvo in pritisniti tipko »Zacveti«. Rože, katerih barvo je uganila, se razcvetijo. Rezultat prvega poskusa je takšen.



Barvo tretje in pete rože je že uganila. V drugem poskusu je poskusila druge barve za preostale tri.



Zdaj že pozna barve vseh petih rož. Kakšne so, po vrsti?

Rešitev

Rožna, modra, modra, rožna, oranžna.

Druga ni ne rožna ne oranžna, torej je modra. Četrta ni ne oranžna ne modra, torej je rožna.

Računalniško ozadje

Kaj bi lahko bilo, če ne more biti ne to ne ono, lahko pa je tisto, vendar ne tole? Ko računalnikar išče napako v programu, varnostno luknjo v sistemu, razlog, da računalnik ne počne tistega, kar bi moral ... je kot detektiv, ki poskuša odkriti morilca. Ali pa kot Jana, ki poskuša z izločanjem napačnih odgovorov odkriti barvo rože.

Bobrova koda

4. – 5. razred

Bobrovka Barbara je za rojstni dan dobila dve štampiljki. Domislila se je, kako z njima zapisovati črke B, A, R, E in Y.

ČRKA	B	A	R	E	Y
KODA					

Ime Barbara bi zakodirala takole: 

Nato zapiše še imena svojih prijateljev. Toda veter je pomešal njene zapise. Pomagaj ji. Poveži ime z ustreznim zapisom s štampiljkami.

Abby 

Arya 

Barry 

Ray 

Rešitev

Pravilni odgovor je:

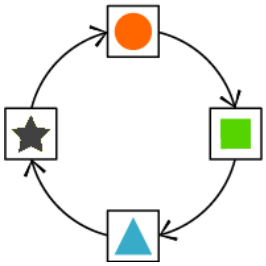
Abby	
Arya	
Barry	
Ray	

Računalniško ozadje

Barbara je zvita: črki A in B, ki sta najbolj pogosti, je predstavila z enim oziroma dvema znakoma. Podoben trik uporabljamo pri stiskanju datotek. Datoteke .zip so narejene tako, da znake, ki so najpogostejši, zapišejo z najmanj biti.

Čarobni valj

4. – 5. razred



Čarobni valj zamenja vse kroge s kvadrati, kvadrate s trikotniki, trikotnike z zvezdami, zvezde pa s krogi.

Spodnja slika kaže, kaj dobimo, če povaljamo krog in trikotnik: krog se spremeni v kvadrat, trikotnik pa v zvezdo.

Kaj dobimo, če povaljamo spodnje zaporedje?

Rešitev

Zvezdo, krog, trikotnik, kvadrat in zvezdo.

Računalniško ozadje

V nekaterih preprostih programskih jezikih, namenjenih preprostim nalogam, podamo seznam sprememb, ki naj jih izvede program.

Ples

Bojan se uči plesa na desni.

Njegov začetni položaj je tak:

V katerih od spodnjih položajev je bil med plesom? Obkroži.

1. Pokrči kolena.
2. Dvigni obe roki.
3. Nagni glavo na eno stran.
4. Iztegni kolena.
5. Nagni glavo na drugo stran.
6. Spusti obe roki.
7. Poravnaj glavo.

Rešitev

Pravilni odgovor je drugi položaj z leve: 

Začetni položaj 

1. Pokrči kolena. 

2. Dvigne obe roki. 

3. Nagne glavo na eno stran. 

4. Iztegne kolena. 

5. Nagne glavo na drugo stran. 

6. Spusti obe roki. 

7. Poravna glavo. 

Računalniško ozadje

Ko program ne naredi tistega, kar bi moral, poskušamo slediti njegovemu izvajanju korak za korakom – tako, kot si moral pri reševanju te naloge slediti plesnemu koraku za plesnim korakom.

Rojstnodnevna torta

4. – 5. razred

Beno praznuje enajsti rojstni dan. Njegova mama bo število 11 zapisala s petimi svečkami.

- Prižgana svečka čisto desno predstavlja število 1.
- Prižgana svečka levo od nje predstavlja število 2, to je dvakrat 1. Tretja svečka predstavlja število 4, to je dvakrat 2. In tako dalje.
- Če gori več kot ena svečka, vrednosti svečk seštejemo. Če prižgemo najbolj desni dve svečki, to pomeni 1 + 2 in dobimo številko 3.

Na desni sliki označi, katere svečke mora prižgati, da bodo kazale število 11?

Rešitev

Prižgati mora svečke z vrednostjo 8, 2 in 1, saj je 8 + 2 + 1 = 11.

Računalniško ozadje

Računalniki shranjujejo števila v dvojiški obliki, to je, z zaporedji, ki jih navadno zapisujemo kot 0 in 1. Če bi bili računalniki narejeni iz tort, pa bi števila shranjevali z ugasnjenimi in prižganimi svečkami. Če.

Skrivna sporočila

4. – 5. razred

Bober Boris želi bobrovki Branki poslati skrivno sporočilo:

DOBIVASEOBPOLOSMIH

Vsako črko zapiše v tabelo s štirimi stolpci od leve proti desni, vrstico za vrstico tako, da začne zgoraj levo, kot kaže slika. Prazen prostor zapolni z X.

Nato ustvari skrivno sporočilo tako, da bere znake z vrha navzdol, stolpec za stolpcem:

DVOLIOABOHBSPSXIEOMX

Bobrovka Branka mu je odgovorila: PPLMRRA!AIBXVŠOX

Kaj mu je sporočila?

Rešitev

Odgovor lahko razbereš tako, da v preglednico od vrha proti dnu stolpec po stolpec zapišeš prejeto sporočilo.

PRAVPRIŠLABOM!

Računalniško ozadje

Računalniške komunikacije pogosto zaščitimo tako, da jih zašifriramo. Danes seveda ne uporabljamo več tako preprostih postopkov šifriranja, saj strokovnjakom (in nepridipravom) razkrivanje takole skritih sporočil ne bi bil noben izziv.

Vrni se nazaj

4. – 5. razred

Robot ima na hrbtu pet tipk. Tipka  ga premakne za eno polje naprej (gor, dol, levo ali desno – kamor je pač obrnjen). Tipka  ga premakne za eno polje nazaj.  in  ga zasukata v levo ali desno, pri čemer ostane na istem polju.

S pritiskanjem na te gumbe sestavimo program. S tipko  ga poženemo. Če jo pritisnemo večkrat, robot večkrat ponovi program.

Štirje mali bobri so se igrali s svojimi roboti. Vsak od njih je ustvaril svoj program. Vsak je **dvakrat** pritisnil gumb GO. Eden od robotov je končal pot na polju, na katerem je začel. Kateri?

A. 

B. 

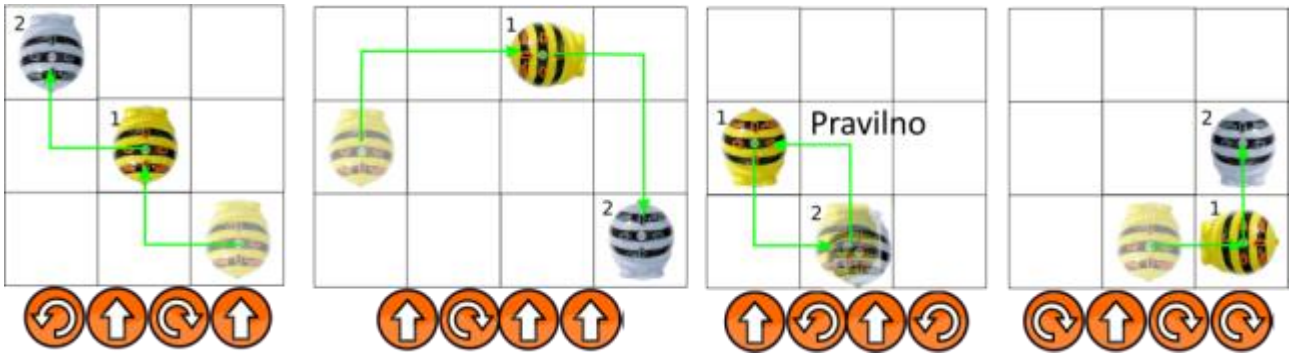
C. 

D. 

Rešitev

Pravilni odgovor je C.

Spodaj so predstavljeni premiki čebel glede na dane ukaze.



Računalniško ozadje

Vsak program, ki ga napišemo, je potrebno preskusiti. V resnici pravi programerji porabijo več časa za iskanje in popravljanje napak kot za programiranje.

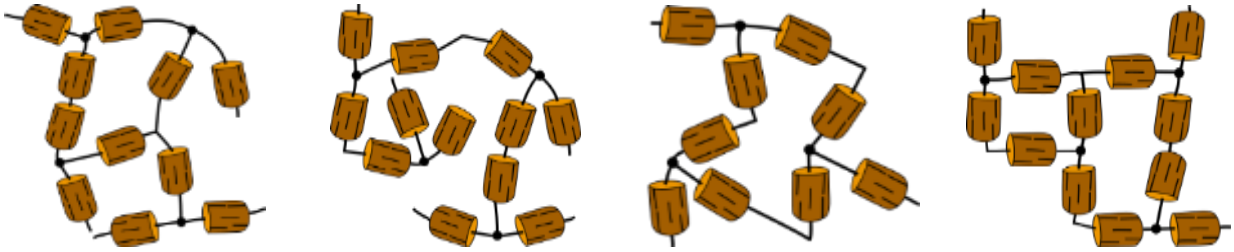
Igrače iz zamaškov

4. – 7. razred

Bober je našel veliko škatlo z zamaški. Po trije zamaški so povezani z vrvicami, kot kaže slika. K vsakemu takemu delu lahko na vrvice, ki grejo skozi zamaške, pritrdiš največ tri druge dele.

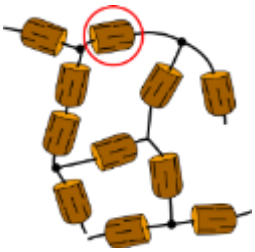
Bober želi iz delcev sestaviti igračo, ne da bi razdrl obstoječe delce.

Katere igrače **ne** more sestaviti?



Rešitev

Prve igrače.



Na obkroženem mestu bi morala biti dva zamaška.

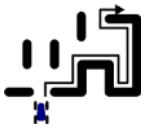
Računalniško ozadje

Prepoznani vzorci pomagajo najti podobnosti v stvareh, ki so sprva videti različne, a imajo kaj skupnega. Ob spoznanju, da je nov problem podoben problemu, ki ga že znamo rešiti, lahko na podoben način poskusimo rešiti tudi novi problem. To je uporabno tudi v matematiki in na drugih področjih znanosti.

Labirint

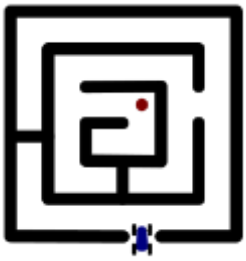
4. – 7. razred

Robot vozi skozi labirint tako, da vedno, ko je mogoče, zavije desno. Če ne more desno, gre naravnost. Če ne more naravnost, gre levo. Pozorno si oglej primer na desni!

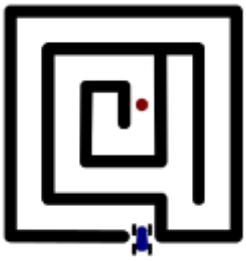


V katerih od spodnjih labirintov bo prišel do rdeče pike?

A



B



C

D

Rešitev

V vseh razen v C. V njem robot ne zapelje v osrednji del in ne doseže pike.

A

B

C

D

Računalniško ozadje

Gre za preprost algoritem, ki omogoča vožnjo skozi katerikoli labirint. Z upoštevanjem pravila se robot nikoli ne izgubi – vedno se vrne na začetno točko gibanja. Pravilo pa ne zagotavlja, da bo robot prevozil celoten labirint.

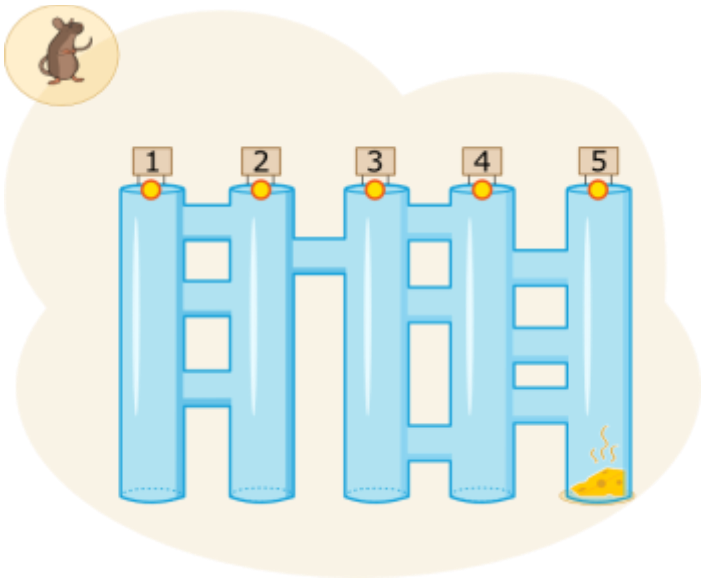
Sistem cevi

4. – 7. razred

Miš je pri vrhu sistema cevi in želi priti do sira na dnu cevi 5.

Miš vedno upošteva te ukaze:

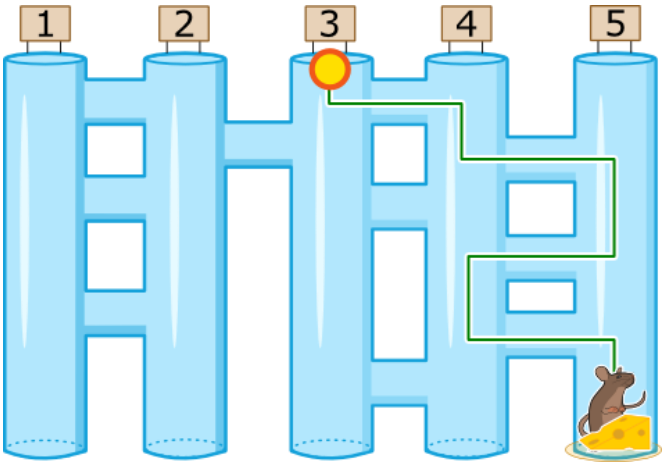
1. pojdi navzdol, **dokler** nisi pri prehodu (levem ali desnem),
2. **pri prehodu** pojdi po vodoravni cevi; nadaljuj z ukazom 1.



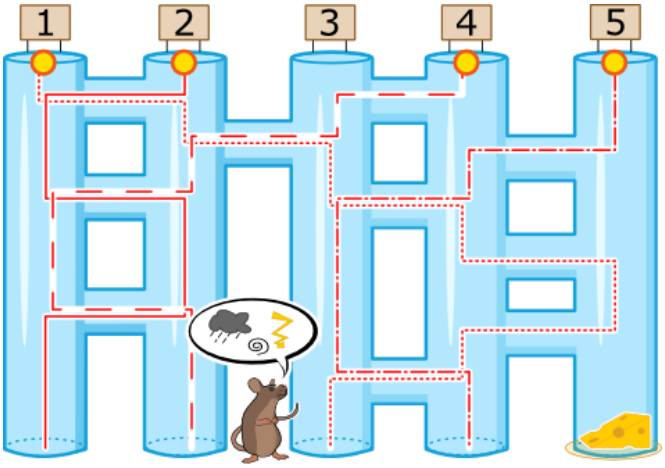
Pri kateri cevi naj miš začne, da bo prišla do sira na dnu cevi 5?

Rešitev

Začeti mora pri cevi 3.



Če bi začela pri kateri od preostalih cevi, ne bi končala na koncu pete cevi.



Ozadje naloge

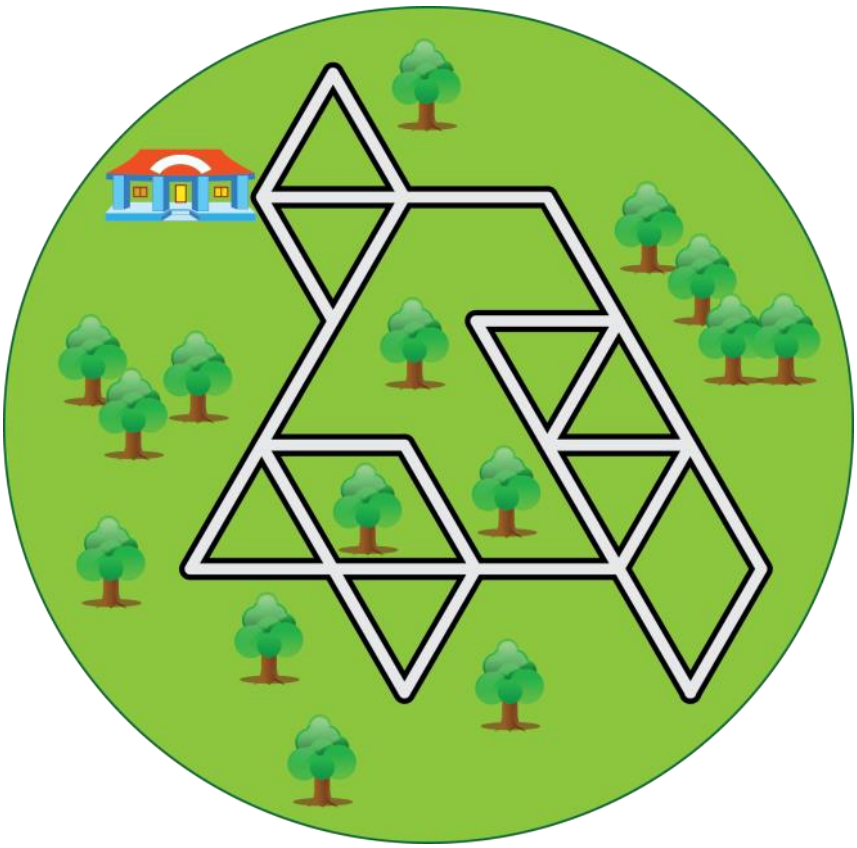
Sledenje navodilom je podobno kot izvajanje programa. Tako kot miš v nalogi mora računalnik slediti ukazom, ki so zapisani v programu. Računalnik izvaja ukaze iz programa, dokler se ta ne konča.

Očisti park

4.- 9. razred, srednja šola

Hišnica Jana skrbi za poti v šolskem parku. V parku je 27 odsekov poti, ki jih ločujejo križišča in ovinki. Vsak odsek je dolg 10 metrov.

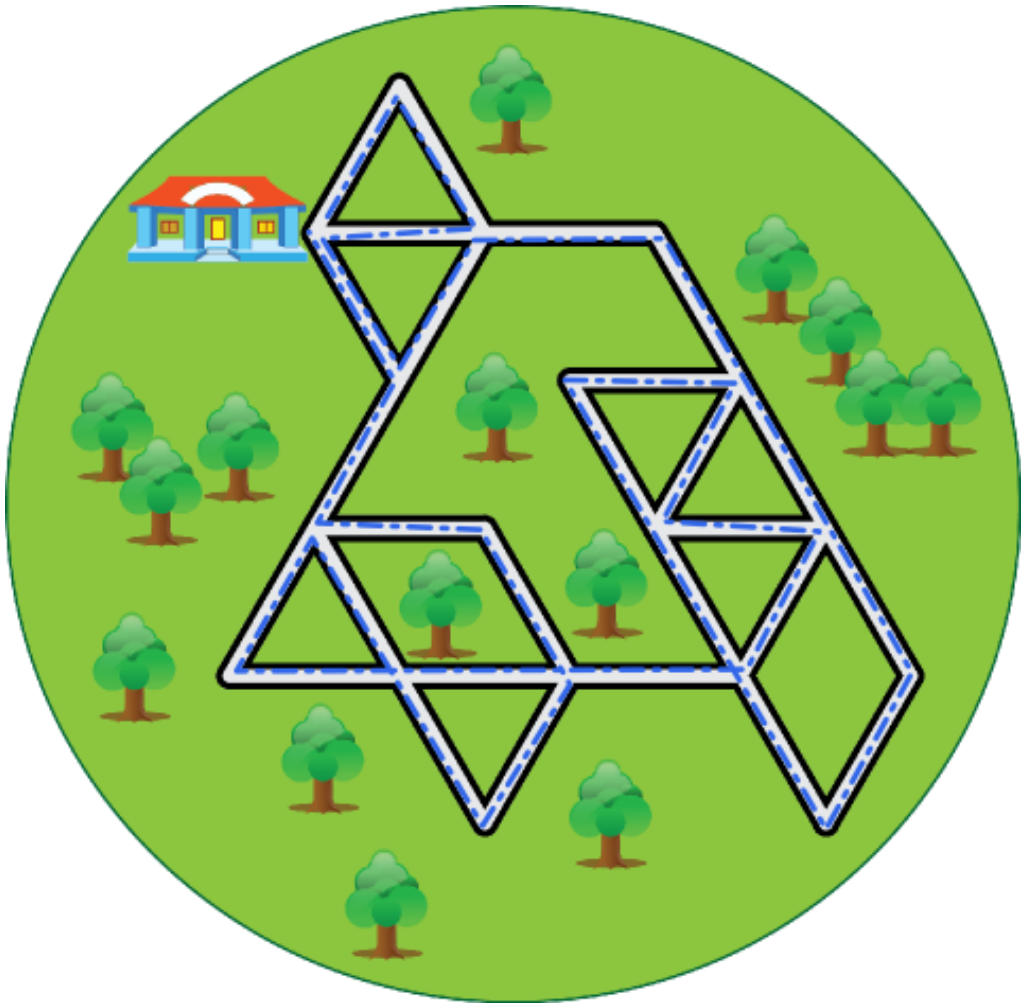
Nagajivi otroci se pogosto igrajo z debli in z njimi zapirajo pot. Vsako jutro gre Jana na obhod, ki ga začne in konča pri šoli. Med obhodom pregleda vse odseke poti. Ker želi biti učinkovita, vedno izbere svojo pot tako, da prehodi čim manj.



Kako dolg je njen jutranji sprehod?

Rešitev

Janin sprehod je dolg 280 metrov.



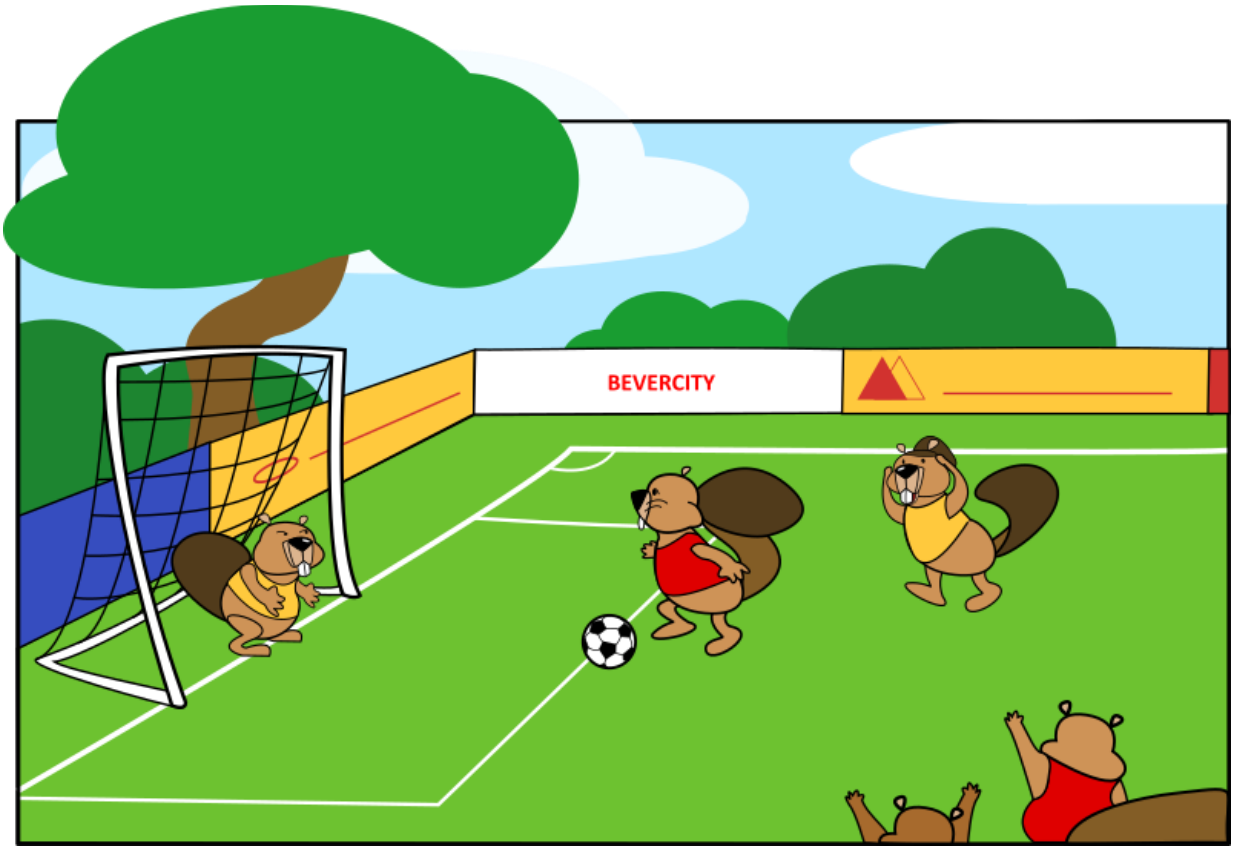
Ozadje naloge

V računalništvu se pogosto srečamo z obhodi na grafu. V tej nalogi iščemo Eulerjev obhod. Eulerjev obhod na grafu pomeni, da pri risanju poti skozi graf obiščemo vse poti, ampak vsako od njih le enkrat, začeti in končati se mora v isti točki. To se zgodi natanko takrat, kadar iz vsake točke vodi sodo število poti. V našem primeru to ne drži, saj iz šole potekajo tri poti in mora vrtnarka eno pot prehoditi dvakrat.

Nogomet

6 - 7. razred

Na tekmi med Ostrozobimi plavalci in Gozdnimi plenilci je padlo šest golov. Le eni ekipi je uspelo dati dva gola zapored. Kakšni so možni rezultati?



Rešitev

4:2, 3:3 ali 2:4.

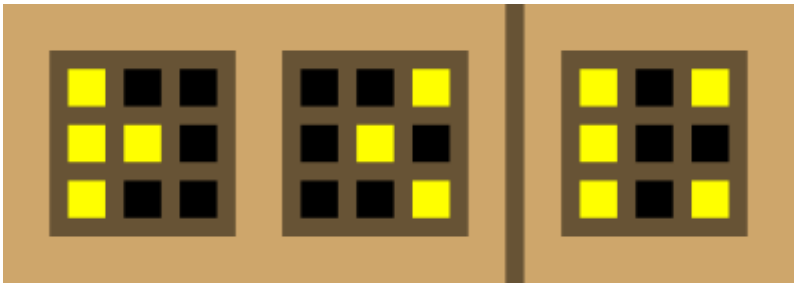
Ekipi označimo z A in B, pri čemer je A tista, ki je prva zadela gol. Možna zaporedja zadetkov so AABABA, ABBABA, ABAABA, ABABBA, ABABAA. Vidimo, da sta ekipi bodisi izenačeni, ali pa je dala ena štiri gole in druga dva.

Računalniško ozadje

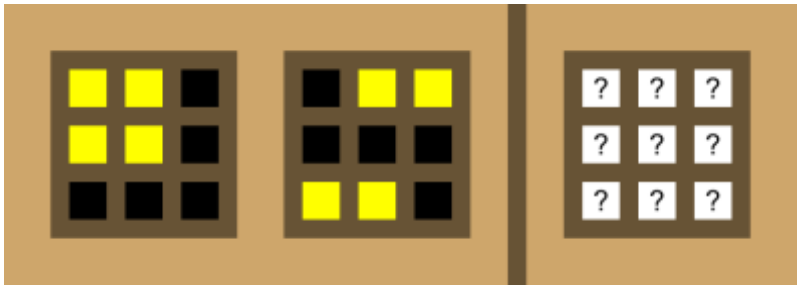
Programer mora pogosto razmisliti o vsem, kar se lahko zgodi v določenem programu in kakšen bo njegov rezultat.

Barvanje

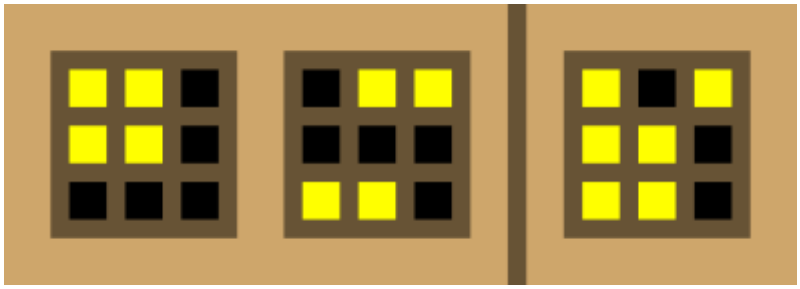
Če združimo sliki na levi, dobimo sliko na desni.



Po kakšnem pravilu deluje združevanje? Kaj dobimo, če združimo spodnji levi sliki? Katera polja bodo rumena?



Rešitev



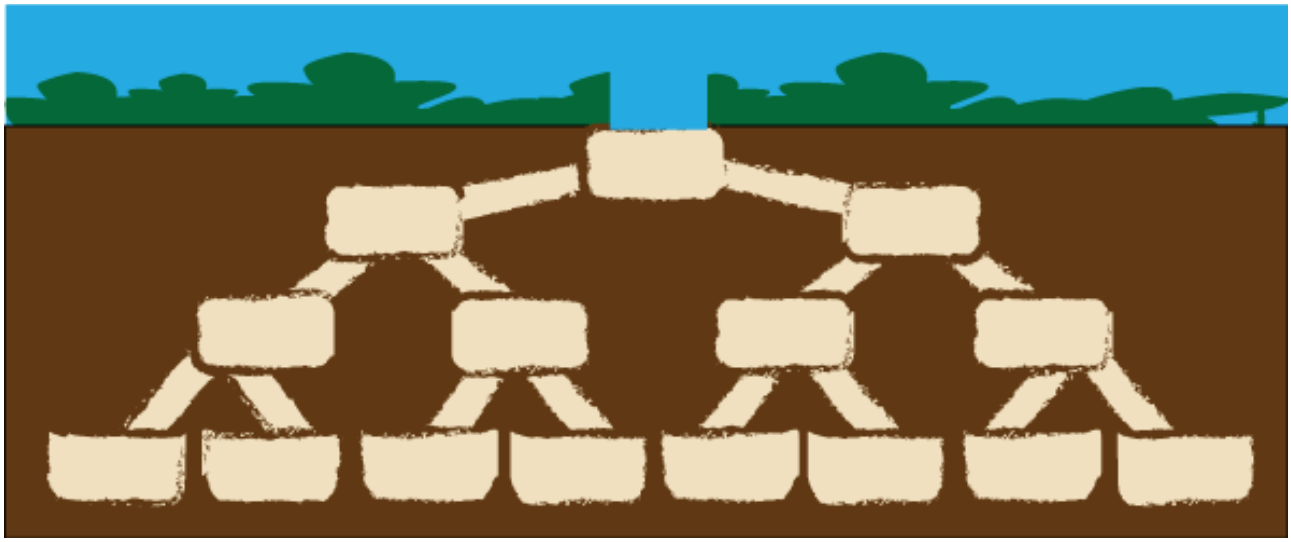
Polje na desni sliki je rumeno, če sta pripadajoči polji na levih slikah različnih barv.

Računalniško ozadje

Operaciji, ki določa rumenost, pravimo *ekskluzivni ali* (*exclusive or*, *xor*) in predstavlja eno osnovnih logičnih operacij, ki jih uporabljamo tako pri programiranju kot pri razvoju logičnih vezij in čipov.

Razpored sob

Bobrova družina – oče Janez, mati Ana in otroci Jože, Peter in Iva – se seli v novo podzemno bivališče. Sobe so še prazne. Vsako od njih lahko uredijo v prehodno sobo ali spalnico. Prehodna soba lahko vodi v nove prehodne sobe in spalnice. **Spalnica ne vodi nikamor**: če soba postane spalnica, bodo sobe pod njo ostale neuporabljene.



Bobri hodijo spat večkrat dnevno.

- Janez in Ana gresta spat dvakrat na dan,
- Jože gre spat trikrat,
- Peter gre spat petkrat,
- Iva gre spat šestkrat na dan.

Vsak bober potrebuje svojo spalnico. Razporedili se bodo tako, da se **ne bo zgodilo, da bi kdo, ki hodi spat večkrat, spal nižje od nekoga, ki spi manjkrat** na dan.

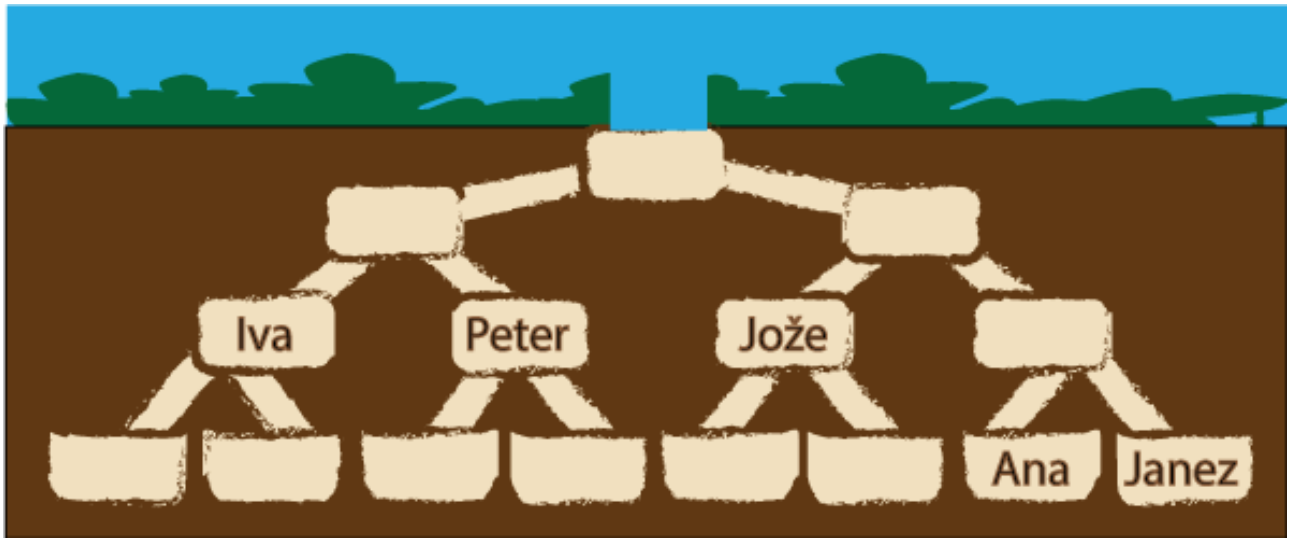
Koliko nadstropij pod zemljo (**prvo je najvišje**) bo Petrova soba?

Rešitev

V prvem nadstropju ne more spati nihče, saj bi s tem zaprl dostop v nižja nadstropja.

Če postavimo Petra v drugo nadstropje, recimo na levo stran, bo zaprl polovico sob in vsi drugi bodo spali v drugi polovici, torej na desni. Tam ne bo smel nihče spati v drugem nadstropju, saj bi sicer zaprl pot do ostalih spalnic. Prišli bi torej v situacijo, ko bi Peter spal v drugem nadstropju, Iva pa v tretjem ali četrtem. To pa ni prav, saj Iva hodi spat večkrat kot Peter.

Bo Peter lahko v tretjem nadstropju? Da, to gre.



Računalniško ozadje

Recimo, da bi želeli zapisati, kdo je hodil spat. Namesto, da zapisujemo imena, lahko povemo kar, kam je šel. Tako bomo za zapisovanje lahko uporabljali le dva znaka: če napišemo, na primer, DL, je šlo za Jožeta, saj le ta gre najprej desno in potem levo. Opišemo lahko tudi celo zaporedje odhodov na spanje, na primer DL DDD LL DL – to pomeni Jožeta, Janeza, Ivo in spet Jožeta.

Za računalnikarje je tudi presledek znak, torej smo v gornjem zaporedju uporabljali tri različne znake – D, L in presledek. V resnici lahko presledek tudi izpustimo: DLDDDLLDL. Tudi to se da prebrati: beremo po vrsti in vedno, ko pridemo do spalnice, začnemo od začetka. Kako vemo, da bo to delovalo in ni potrebno iti naprej, do spalnice v kakem nižjem nadstropju? Preprosto: vemo – ker smo se tako dogovorili – da pod spalnico ni nobenih spalnic več.

Dodatno pravilo (v resnici uporabljamo nekoliko bolj zapletenega, a v takšni nalogi vam pač ne moremo razkriti vse računalniške znanosti) skrbi za to, da bi bila zaporedja čim krajša. Tiste poti, ki se pojavljajo večkrat (na primer Ivina) so opisane z manjšim številom znakov kot tiste, ki niso tako pogoste.

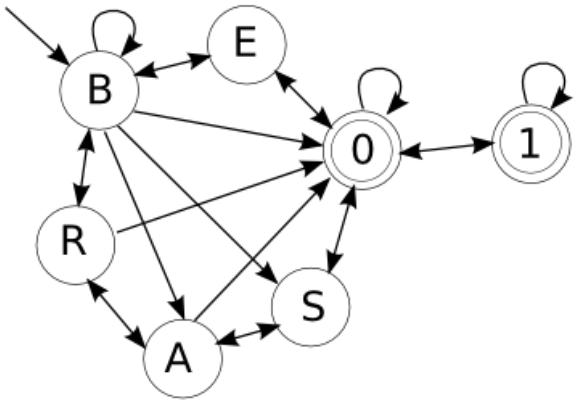
V številnih nalogah s tega tekmovanja si bobri izmislijo določen način zapisovanja znakov. Primer je, recimo, Bobrova koda z letošnjega tekmovanja. Vse te naloge imajo enako ozadje: čim krajši zapis besedil.

Več o tem lahko izveš na Vidrini aktivnosti Stiskanje besedil.

Registrske tablice

6. – 7. razred (državno)

Gornji splav je imeniten, vendar ima neveljavno registrsko tablico. Prave tablice so sestavljene po spodnji shemi.



Začnemo pri črki B in sledimo puščicam. Posamezen znak lahko izberemo večkrat. Zadnji znak mora biti 0 ali 1. Registrska tablica je sestavljena iz znakov, ki jih naberemo na poti. Primer veljavne registracije je, na primer, BR0EBS00.

Katere od spodnjih registracij niso veljavne?

BARAS0E0 BEBARA010 BE0SARB BARBARA01 BABE001 BARABA01

Rešitev

Neveljavne so registracija BE0SARB, ki se ne konča z 0 ali 1, ter registraciji BABE001 in BARABA01, pri katerih črka B sledi črki A, kar ni dovoljeno.

Računalniško ozadje

Shemi, s kakršno so določene pravilne registrske tablice, pravimo končni avtomat. Računalniki ga uporabljajo pri branju besedil, prevajanju programov in še marsikje.

Čarobni napoj

6. – 9. razred

Bober Nejc je odkril pet čarobnih napojev z naslednjimi učinki:

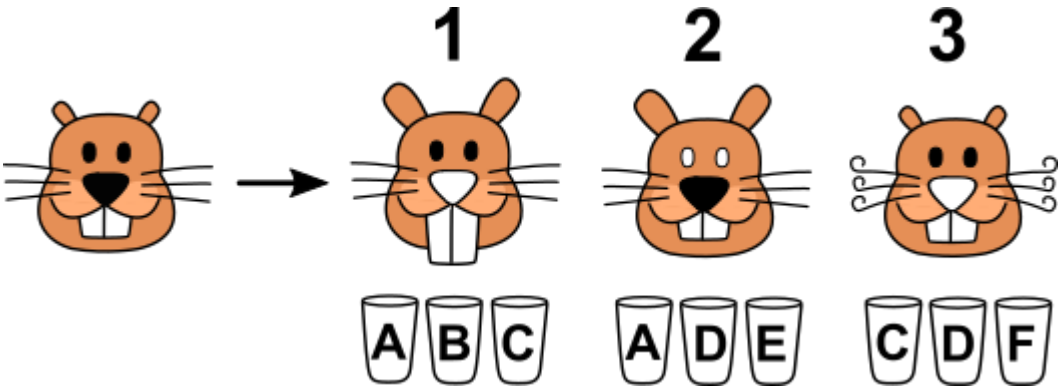
- po enem do napojev se bobru povečajo ušesa,
- po enem mu zrastejo zobje,
- eden od napojev mu zaviha brke,
- eden od njih obarva bobrov nos belo,
- eden mu pobeli oči.

Bober Nejc natoči napoje v kozarce, doda pa še kozarec z vodo. Da bi vedel, kaj je kje, je kozarce označil s črkami – vendar je pozabil zapisati, v kateri kozarec je vlil kateri napoj. Zato naredi niz poskusov.

Poskus 1: če bober spije napoje iz kozarcev A, B in C, je učinek prikazan na sliki 1.

Poskus 2: če bober spije napoje iz kozarcev A, D in E, je učinek prikazan na sliki 2.

Poskus 3: če bober spije napoje iz kozarcev C, D in F, je učinek prikazan na sliki 3.



V katerem kozarcu je samo voda?

Rešitev

Kozarec D.

Pri poskusu 1 noben od kozarcev A, B in C ni voda, saj so se pri bobru prikazali trije učinki.

Pri poskusu 2 je eden od kozarcev D ali E voda, drugi pa napoj, kjer se bobru nos obarva belo. Kozarec A ni voda, kar smo ugotovili že iz poskusa 1.

Pri poskusu 3 je eden od kozarcev D ali F voda, drugi pa napoj, kjer se bobru zavihajo brki. Kozarec C ni voda, kar smo ugotovili že iz poskusa 1.

Torej je voda v kozarcu D.

Računalniško ozadje

V tej nalogi je predstavljena zbirka dejstev, na podlagi katerih moramo uvideti nova dejstva. Pomagamo si z logičnim sklepanjem.

Ravno tako pa lahko to nalogo rešujemo s pomočjo množic, ko opazujemo, kateri elementi se pojavljajo v skupnih množicah in kateri ne. Na tak način sklepamo, kaj je skupnega množicam s skupnimi elementi in kaj ne.

Dieta

6. – 9. razred

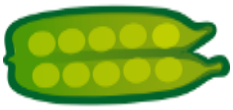
Bober David je na šestdnevni dieti. Vsak dan lahko je le eno vrsto hrane s seznama: krompir, korenje, ribe in grah.

Vsak dan se bober David odloči, kaj bo jedel tisti dan, glede na sledeča navodila:

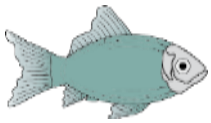
- Dva dni zapored ne sme jesti iste hrane.
- Korenje mora jesti natančno vsak tretji dan; začne s tretjim dnevom diete.
- Ribe lahko je le, če ni prejšnji dan jedel krompirja.

Prejšnje štiri dni je jedel:

1. dan: grah



2. dan: ribe



3. dan: korenje

4. dan: krompir

Kaj mora jesti na peti in šesti dan diete?

A. 5. dan: korenje, 6. dan: grah
B. 5. dan: krompir, 6. dan: ribe
C. 5. dan: ribe, 6. dan: korenje
D. 5. dan: grah, 6. dan: korenje

Rešitev

D

Glede na 2. pravilo mora bober David 6. dan jesti korenje.

Glede na 1. pravilo 5. dan ne more jesti niti krompirja niti korenja, ker bi potem dva dni zapored jedel isto vrsto hrane. Torej mu za 5. dan preostanejo ribe ali grah.

5. dan ne more jesti rib, ker bi potemtakem na 4. dan jedel krompir in zatem ribe na 5. dan, s čimer bi kršil 3. pravilo. Torej preostane le še odgovor D.

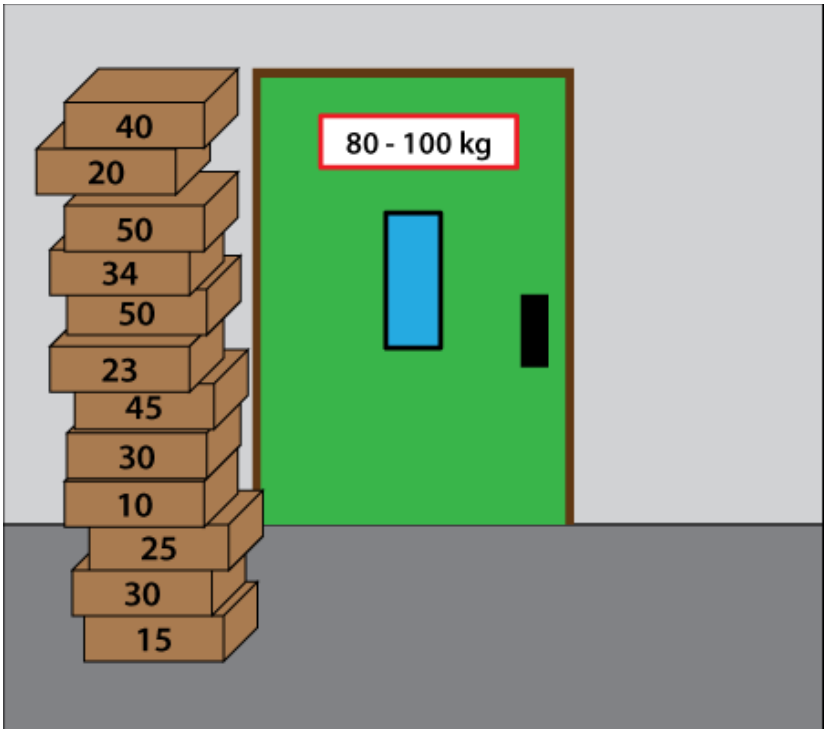
	1. dan	2. dan	3. dan	4. dan	5. dan	6. dan	kršeno pravilo
A.							2
B.							1
C.							3
D.							

Računalniško ozadje

V računalništvu se s situacijami, kjer so vnaprej določena stanja in veljajo pravila za prehajanje na naslednje, ukvarja cela znanost. Takšnim sistemom pravijo končni avtomati. Npr. v našem problemu iz »stanja« krompir ni možno nadaljevati na stanje riba ali krompir.

Dvigalo

6 - 9. razred



V dvigalo lahko naložimo od 80 do 100 kg. Levo od dvigala je kup različno težkih škatel. V dvigalo najprej postavimo 40-kilogramsko škatlo, nato 20-kilogramsko. 50-kilogramska bi presegla dovoljeno težo, zato jo odložimo desno od dvigala. Ko v dvigalo dodamo še 34-kilogramsko škatlo, je v dvigalu 94 kg, zato odpeljemo tovor.

Ko se vrnemo nadaljujemo tako, da nalagamo škatle z levega kupa (ne desnega!) in jih nalagamo v dvigalo, dokler teža v njem ne doseže 80 kg, ko ga lahko odpeljemo. Pretežke škatle odlagamo na desno.

Če se levi kup izprazni, začnemo jemati škatle z desnega, pretežke škatle pa odlagamo na levo. To počnemo, dokler ne izpraznimo desnega kupa; nadaljujemo z levim in tako naprej, dokler ne odpeljemo vseh škatel.

Kaj od spodnjega je res?

A. Škatel na ta način ni mogoče prepeljati v manj kot petih vožnjah.
B. V dvigalo nikoli ne bomo naložili natančno 100 kg.
C. Nikoli ne bo potrebno preložiti škatle z desnega na levi kup.
D. Postopka ni mogoče izpeljati.

Rešitev

Odgovor C.

Takole gre:

odpeljemo	levi kup	desni kup
	40 20 50 34 50 23 45 30 10 25 30 15	
40 20 34	50 23 45 30 10 25 30 15	50
50 23 10	25 30 15	30 45 50
25 30 15 30		45 50
45 50		

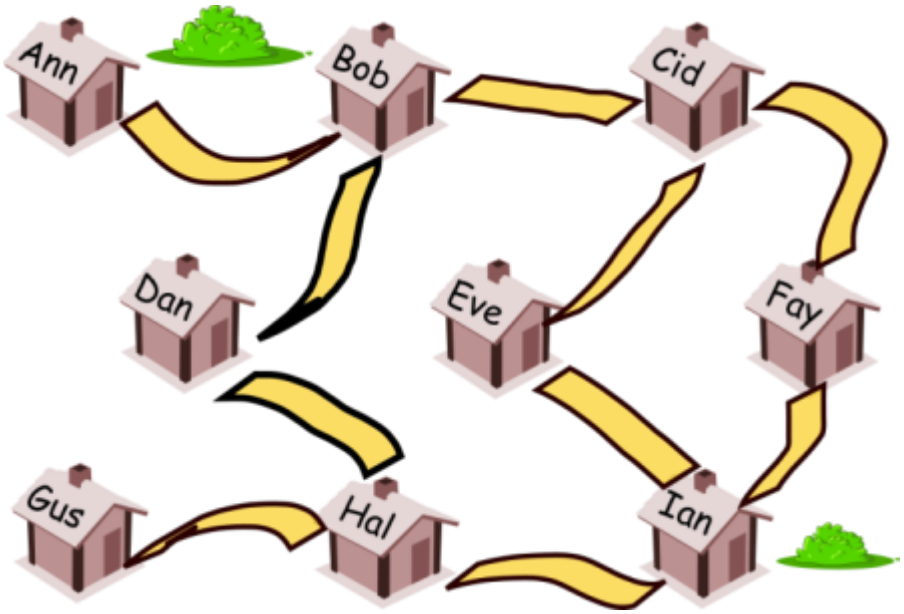
Računalniško ozadje

Računalniki uporabljajo dve obliki vrst. Prvi pravimo kar vrsta: kdor prej pride, prej gre. Druga je sklad: kdor zadnji pride, gre prvi. Druga zveni »krivično«. V resničnem svetu morda res, v računalništvu pa jo srečamo za vsakim vogalom. Popolnoma preprost – in vsakdanji primer – je tipka »nazaj« v brskalniku, ki ti najprej pokaže tisto stran, ki si jo obiskal nazadnje.

Gasilci

6. – 9. razred

Župan Bobermesta išče prostovoljne gasilce. Zemljevid predstavlja domove bobrov in ceste med njihovimi domovi.



Župan želi, da je vsaka hiša ali dom prostovoljnega gasilca ali pa je od doma prostovoljnega gasilca oddaljena le eno cesto.

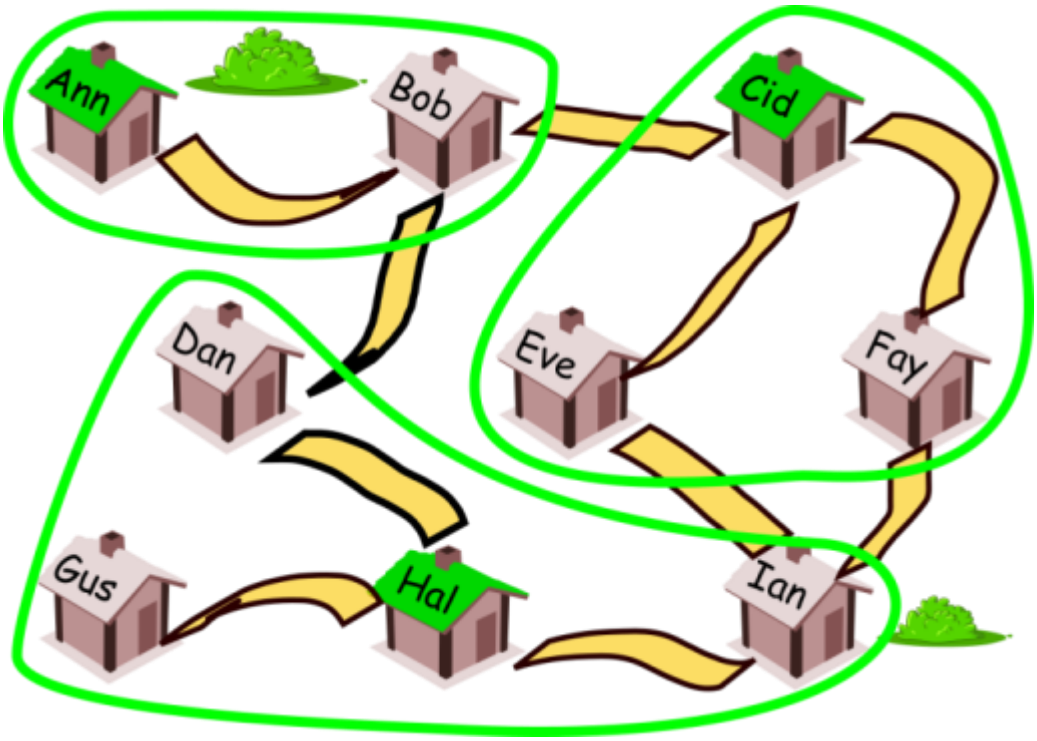
Kolikšno je najmanjše število prostovoljnih gasilcev, ki jih župan potrebuje?

Rešitev

Župan potrebuje 3 prostovoljne gasilce.

Ker Ann in Gus živita v hišah, do katerih vodi le ena cesta, mora zagotovo biti gasilec ali Bob ali Ann ter ali Gus ali Hal. Če sta gasilca tako Bob kot Hal, s tem zavarujemo več hiš, vendar še vedno nista zavarovani hiši Eve in Fay. Zato mora biti gasilec še ali Cid ali Ian.

Obstaja več možnih rešitev. Ena je spodaj.



Si lahko zamisliš še kakšno?

Računalniško ozadje

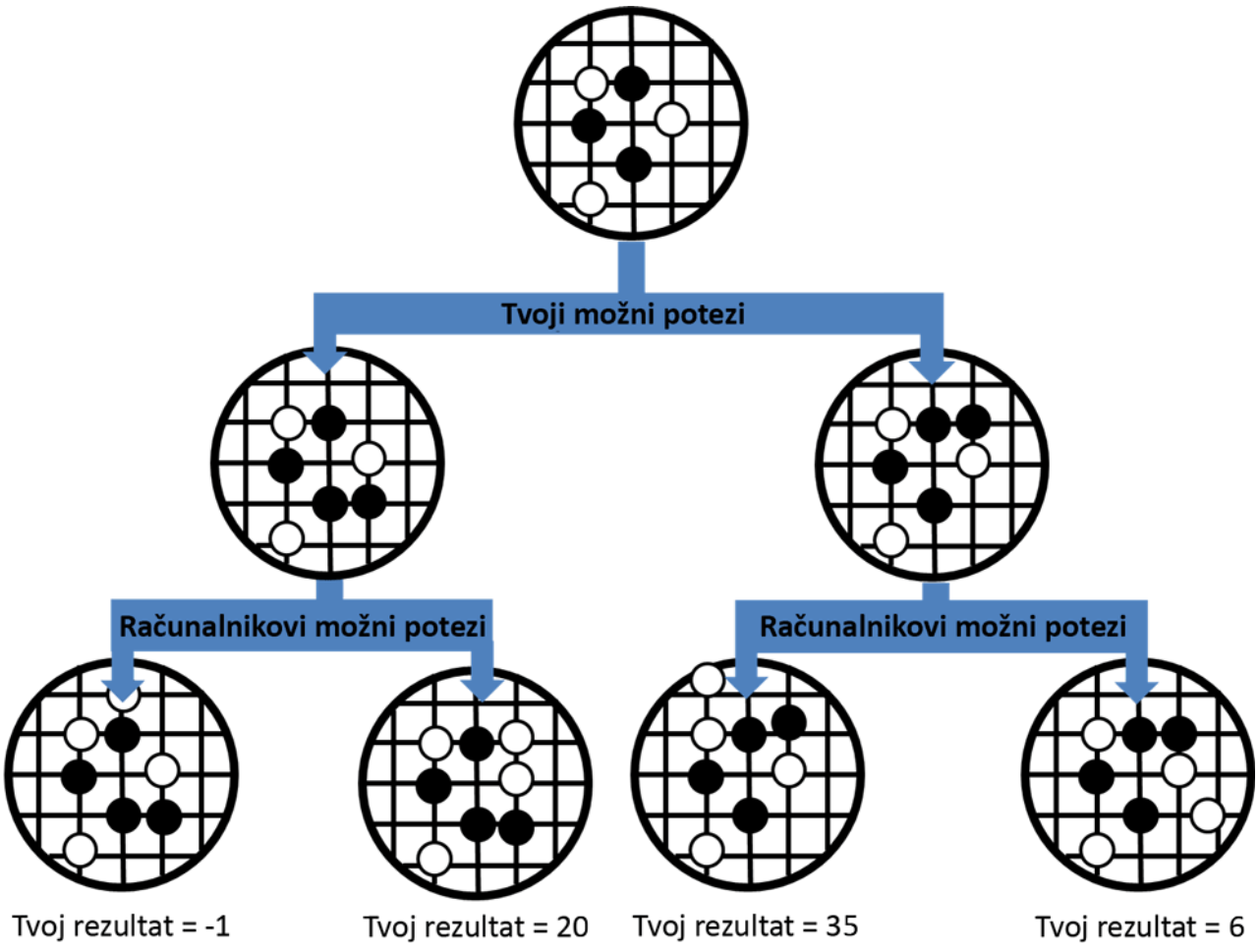
Hiše in ceste predstavljajo vozlišča in povezave v grafu. Predstavljeni problem je minimalno pokritje vozlišč, ki je pomemben del računalništva. V vsakdanjem življenju na take probleme naletimo, ko je potrebno določiti, kam postaviti varnostne kamere ali pa kam postaviti lekarne, da so na voljo določenemu okraju in da jih ni preveč ali premalo.

Go

6 - 9. razred

Go je zapletena igra s pravili, ki jih razumejo samo Bobri. V tej nalogi je nimamo časa razlagati.

V neki igri proti računalniku si se znašel v položaju, ki jo kaže zgornji krog na sliki. Na voljo imaš dve potezi – črni žeton lahko položiš na polje zgoraj ali spodaj desno (kroga v drugi vrsti). Po vsaki od teh potez bo imel tudi računalnik dve možni potezi in igra se bo končala z enim od štirih položajev v zadnji vrsti. Prvi vsakem od njih je napisan tudi rezultat.



Kakšen je najvišji rezultat, ki ga lahko dosežeš? Upoštevaj, da je (tudi) računalnik zelo dober igralec?

Rešitev

6

Na prvi pogled je videti, da lahko dobiš rezultat 35. Vendar moramo upoštevati, da je računalnik pameten in igra tako, da je slabše zate: če res odigraš desno potezo, bo računalnik odigral svojo desno in rezultat bo 6, ne 35. Leva poteza se ti tudi ne izplača, saj bi v tem primeru računalnik prav tako odigral levo potezo in rezultat bi bil -1.

Računalniško ozadje

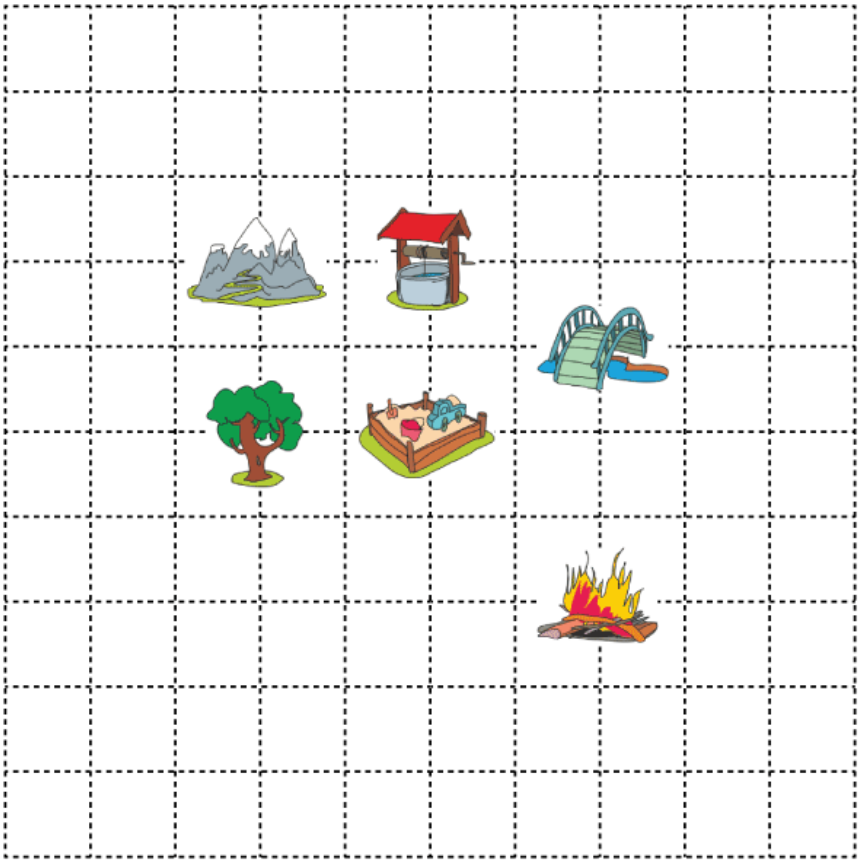
Računalniki igrajo šah in podobne igre tako, da si sestavijo »drevo«, kot ga vidimo na sliki. Ker ne morejo preiskati vseh možnih pozicij (že pri tako »preprosti« igri, kot je šah, jih je namreč preveč), sestavijo drevo za, na primer, naslednjih petnajst potez in ocenijo, kako dobri ali slabi (zanje) so položaji na koncu. Nato opravijo podoben razmislek, kot si ga moral pri reševanju naloge ti: v vsakem koraku predpostavi, da se bo on odločil za najboljšo potezo zase, ti pa za najslabšo zanj.

Takšnim drevesom - v katerih se izmenjuje najboljše za enega in najslabše za drugega igralca - učeno rečemo Minimax drevesa.

Kje je zaklad?

6 - 9. razred

Ognjišče je pri (3|3) in vodnjak pri (7|5). Zaklad je na (7|7). Kje je skrit? Pri hribu, drevesu, igrišču ali mostu?



Rešitev

Pri hribu.

Odkriti moramo, da prva številka pomeni razdaljo od spodnjega roba, druga pa od desnega.

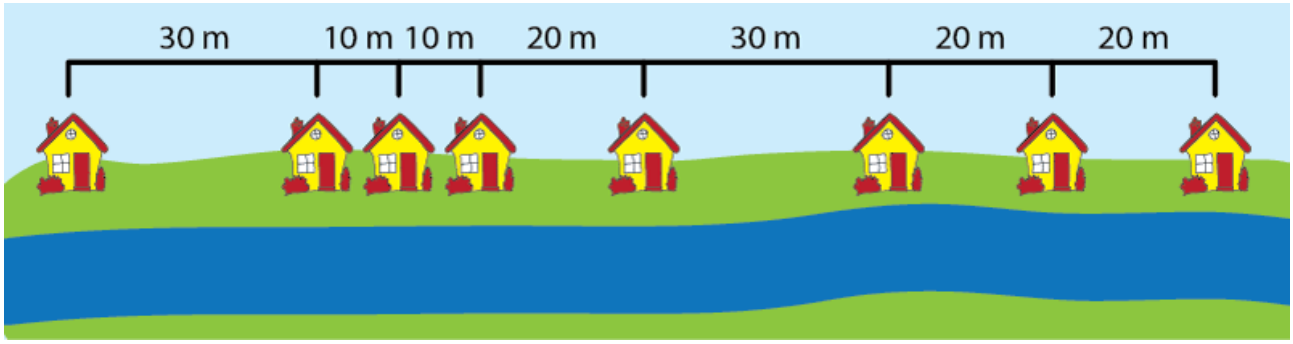
Računalniško ozadje

V računalniški grafiki uporabljamo različne koordinatne sisteme – zapise, s katerimi povemo, kje na sliki se nahaja določen objekt. V resnici pogosto uporabljamo ravno obraten zapis, v katerem prva številka pove razdaljo od levega, druga pa od zgornjega roba.

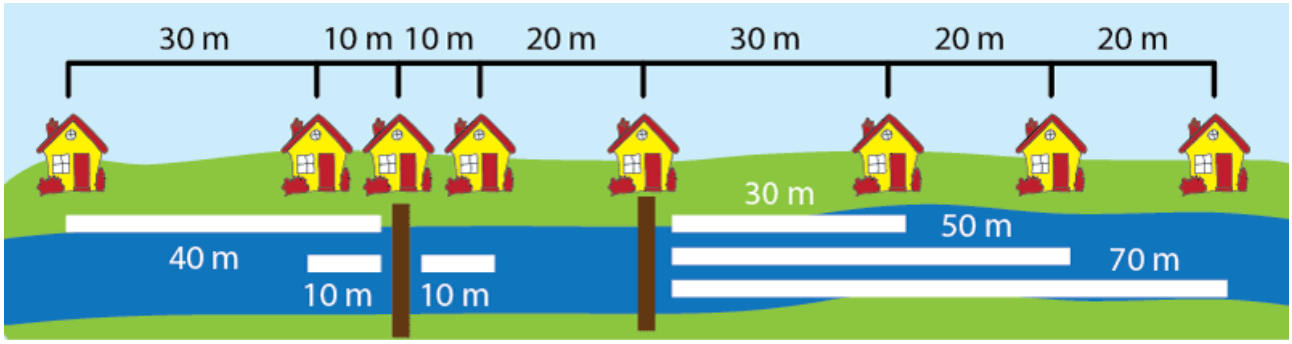
Mosta

6 - 9. razred

Bobri živijo ob reki. Razdalja med bivališčema prvih dveh je 30 metrov, med bivališčema drugih dveh 10 metrov, do četrtega je še 10 metrov ... in tako naprej.

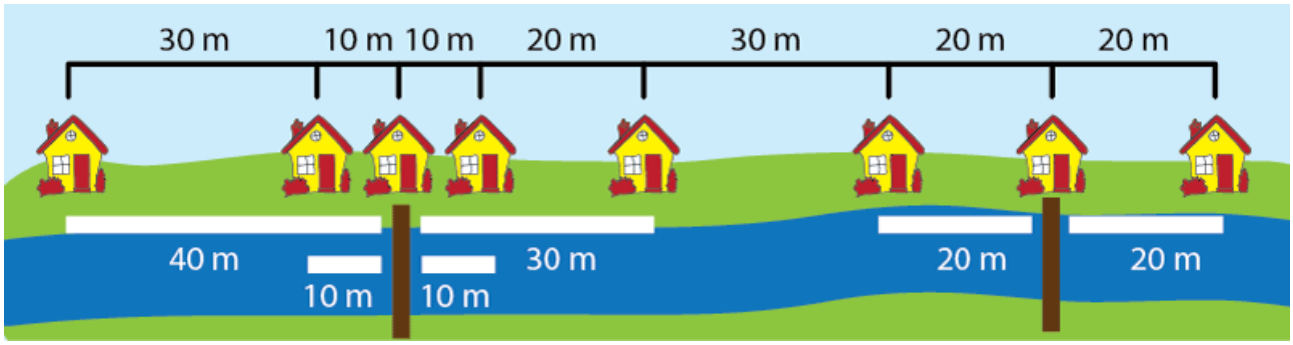


Nekdo je predlagal, da postavijo dva mosta, enega pred tretjo hišo in enega pred peto. Vendar so kmalu uvideli, da to ni dobra ideja: če bo vsak hodil čez najbližji most, bo vsota poti, ki jih bodo morali opraviti 40 + 10 + 0 + 10 + 0 + 30 + 50 + 70 = 210 metrov.



Nekateri so menili, da je potrebno prestaviti srednji most bolj desno in levega še bolj levo. Drugi so imeli drugačne predloge. Kaj pa praviš ti? Pred katere hiše postaviti mosta, da bo vsota poti od vsake hiše do najbližjega mostu najkrajša? Kolikšna je v tem primeru najmanjša vsota poti v metrih?

Rešitev



40 + 10 + 0 + 10 + 30 + 20 + 0 + 20 = 130 metrov.

Računalniško ozadje

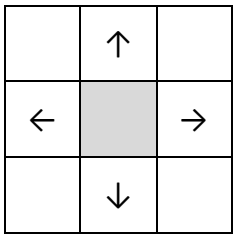
Takšnim problemom pravimo optimizacijski problemi. V okviru omejitev, ki jih imamo (postaviti želimo dva mosta, stati pa morata pred dvema hišama in ne kje vmes) moramo poiskati rešitev, ki je najboljša glede na določen kriterij (skupna razdalja od vsake hiše do najbližjega mostu mora biti najmanjša).

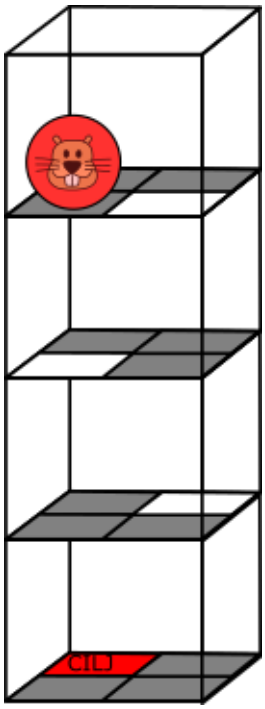
Z optimizacijskimi problemi se ljudje ukvarjajo, že odkar so začeli graditi piramide (ali pa še dlje). Matematiki so odkrili veliko načinov za reševanje tovrstnih problemov. Najbolj zapletene med njimi pa danes rešujemo s pomočjo računalnikov.

Robot

6. – 9. razred

Robota kroglo usmerjaš z igralnim ploščkom. Na voljo so štirje gumbi, označeni s puščicami, s katerimi premakneš robota na naslednje polje.





Če se robot premakne na belo polje, pade na polje pod njim.

Štirje otroci so sestavili zaporedja ukazov, ki naj bi pripeljala robota do rdečega polja. Eno od zaporedij **ne deluje pravilno**. Katero?

A. → ← → ↑ ←

B. → ← ↑ → ↓ ← ↑

C. ↑ → ← ↑ → ↓ → ↑ ←

D. ↑ → ↓ ↑ ← ↓ ↑ → ←

Rešitev

C.

Računalniško ozadje

Računalniški program je zaporedje danih ukazov. Ta naloga od reševalca zahteva, da prebere program v zelo preprostem jeziku z le štirimi ukazi (smerne puščice) in najde napako. Temu v računalništvu rečemo razhroščevanje (angl. debugging).

Trak

6 - 9. razred

Na rojstnodnevni zabavi si želel okrasiti prostor s pisanim trakom, ki je videti takole:

Nagajiva sestra je odrezala in ukradla kos traku, ki je na sliki označen s tremi pikami. Vrnila ti ga bo, če ji poveš, kako dolg je.

Ocenjuješ, da je daljši kot 30, a krajši kot 35 koščkov. Kako dolg je?

Rešitev

31.

Trak je sestavljen iz vzorca dolžine 4, ki se ponavlja: MRRZ. Trak, ki ga ima sestra, se začenja z Z. Nato sledi neznano število koškov MRRZ. Njen del se konča z MR, saj se desni del na gornji sliki nadaljuje z RZ.

Njen del je torej dolg 1 kos (Z) + del neznane dolžine, ki je večkratnik 4 + 2 kosa. Če dolžino traku delimo s 4, moramo dobiti ostanek 3. Možne dolžine so 31, 32, 33 in 34. Ostanek po deljenju 31 s 4 je 3, torej je to iskana dolžina.

Računalniško ozadje

V računalništvu pogosto iščemo ponavljajoče se vzorce in jih poskusimo zakodirati na krajši način. Na podoben način iščemo vzorce v DNK, le da so tam vzorci, ki jih iščemo, bolj zapleteni.

Vrni se

6. – 9. razred

Robotska čebela ima na hrbtu štiri gumbe s puščicami. Čebela se premika po tleh, tlakovanih s kvadratnimi ploščicami. Pri tem upošteva program, ki ga sestavimo z zaporedjem pritiskov na gumbe:

 Premakni se naprej na ploščico spredaj.

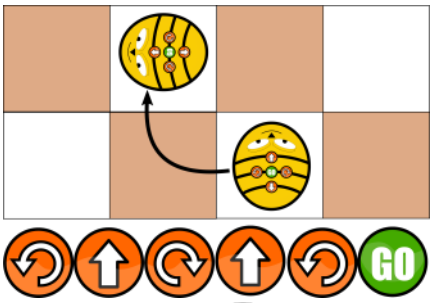
 Ostani na tej ploščici in se obrni za 90° levo.

 Ostani na tej ploščici in se obrni za 90° desno.

 Vzvratno se premakni na ploščico zadaj.

Zaporedje ukazov se začne izvajati s pritiskom na gumb  .

Primer zaporedja pritisnjenih gumbov in njegova izvedba je prikazan na sliki.



Čebela si zapomni zadnje vneseno zaporedje ukazov, zato ponovni pritisk na gumb  povzroči ponovno izvedbo vnesenega zaporedja. Pri nekaterih programih se čebela, če dovoljkrat pritisnemo  vrne ravno na začetni položaj. Kateri od spodnjih programov nikoli ne pripelje čebele na začetno polje z začetno usmeritvijo, ne glede na to, kolikokrat pritisnemo gumb  ?

A. 

B. 

C. 

D. 

Rešitev

A. 

Računalniško ozadje

V nalogi smo opazovali programe v preprostem programskem jeziku. Pritiskanje na gumb  predstavlja »zanko«, ki omogoča večkratno ponavljanje določenih ukazov.

Vrstni red kuhanja

6. – 9. razred

Rok pripravlja večerjo za prijatelje. Grabi ga panika, saj njegovi prijatelji pridejo kmalu.

Rok ima dva štedilnika. Skuhati mora krompir, juho, pripraviti omako ter meso. Recept zahteva, da meso pripravimo v omaki, in zato je nujno, da je omaka pripravljena pred mesom.

Krompir se kuha 30 minut, za juho potrebuje 80 minut, omaka je pripravljena v 10 minutah, meso pa se mora pripravljati v omaki 60 minut.

Gostje pridejo čez 90 minut. Kakšen vrstni red kuhanja naj uporabi, da bodo jedi pripravljene pred njihovim prihodom?

A.	1. korak	2. korak
Štedilnik 1	krompir	juha
Štedilnik 2	omaka	meso

B.	1. korak	2. korak
Štedilnik 1	omaka	juha
Štedilnik 2	krompir	meso

C.	1. korak	2. korak
Štedilnik 1	meso	juha
Štedilnik 2	krompir	omaka

D.	1. korak	2. korak
Štedilnik 1	meso	krompir
Štedilnik 2	juha	omaka

Rešitev

Odgovor B.

Kuhanje juhe in krompirja na istem štedilniku traja 110 minut in priprava juhe in mesa na istem štedilniku traja 140 minut, kar je v obeh primerih predolgo. Torej morata biti juha in omaka pripravljena na istem štedilniku. S tem izločimo odgovora A in C.

Če najprej skuhamo juho, potem moramo počakati, da se skuha juha preden začnemo s pripravo mesa. To traja skupno 120 minut (80 min za juho, 10 min za omako in 30 min za meso). Ker moramo najprej skuhati omako (pred mesom), izločimo odgovor D.

Edini možni odgovor je tako B. Omaka je pripravljena v 10 minutah, zatem na štedilniku kuhamo juho 80 minut. Na drugem štedilniku začnemo s kuhanjem krompirja (30 minut), nato pa meso pripravljamo v že ta čas narejeni omaki (60 minut).

Računalniško ozadje

Računalniki lahko izvajajo več programov istočasno. Predstavljaj si, da sta štedilnika dva procesorja, ki opravljata svoje delo, medtem ko je Rok programer. En štedilnik lahko kuha le eno jed naenkrat. Tudi procesor opravlja eno nalogo naenkrat. Če imamo le en štedilnik oz. procesor, moramo kuhati jedi zaporedoma oz. ukaze izvrševati enega za drugim. Če pa imamo na voljo več štedilnikov oz. procesorjev, lahko istočasno (vzporedno) kuhamo več jedi oz. izvršujemo več ukazov, kar skrajša čas izvajanja.

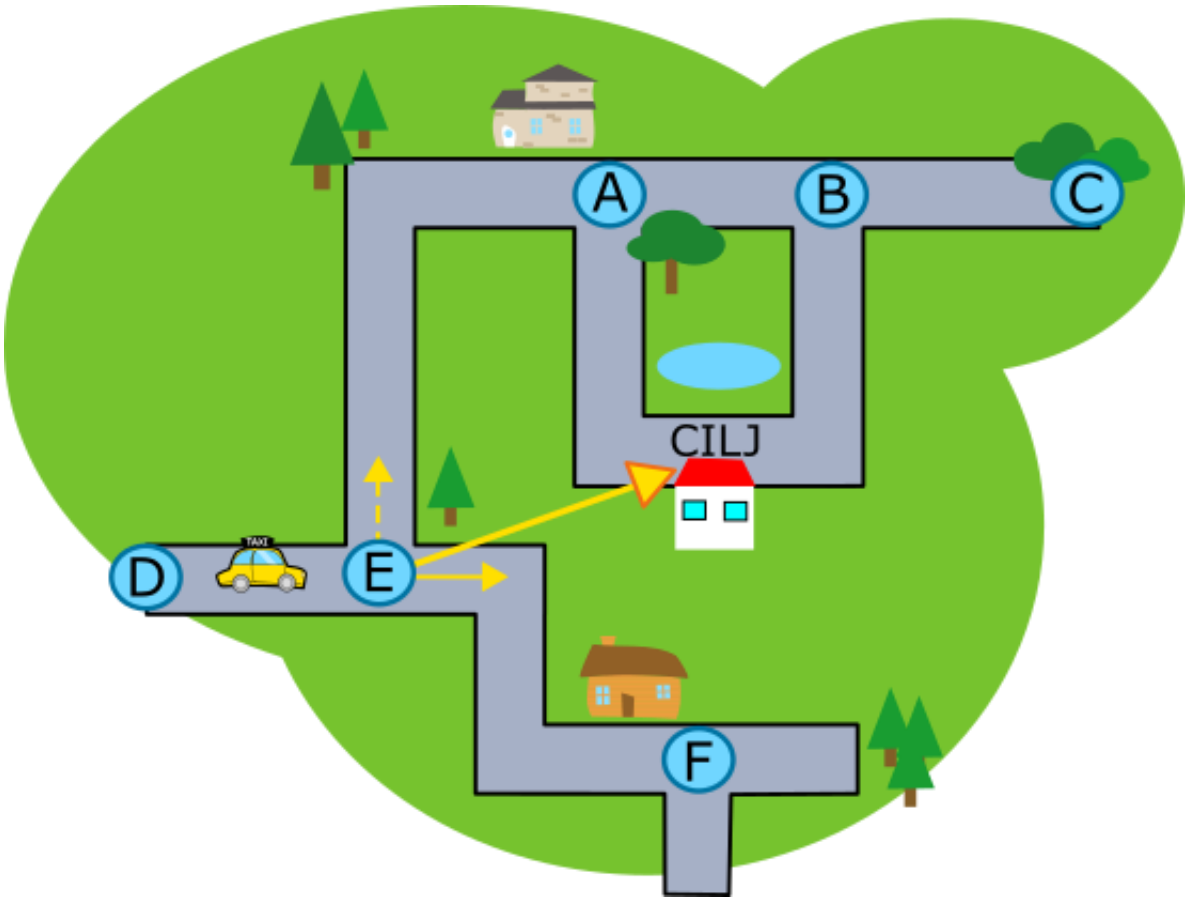
Avtonomni taksi

6. - 9. razred (državno)

Avtonomni taksi se po cesti samodejno premika proti svojemu cilju. Pri tem upošteva naslednja pravila:

1. Sledi cesti.
2. Na križišču (na sliki so označena s črkami) izberi cesto, katere smer se najmanj razlikuje od tiste, v kateri leži cilj. To ne velja za zavijanje v nasprotno smer (180°).
3. Na koncu slepe ulice se obrni v nasprotno smer (180°).

Cilj je bela hiša. Na križišču E pelje taksi naravnost, saj je smer puščice do cilja bolj vodoravna kot navpična.

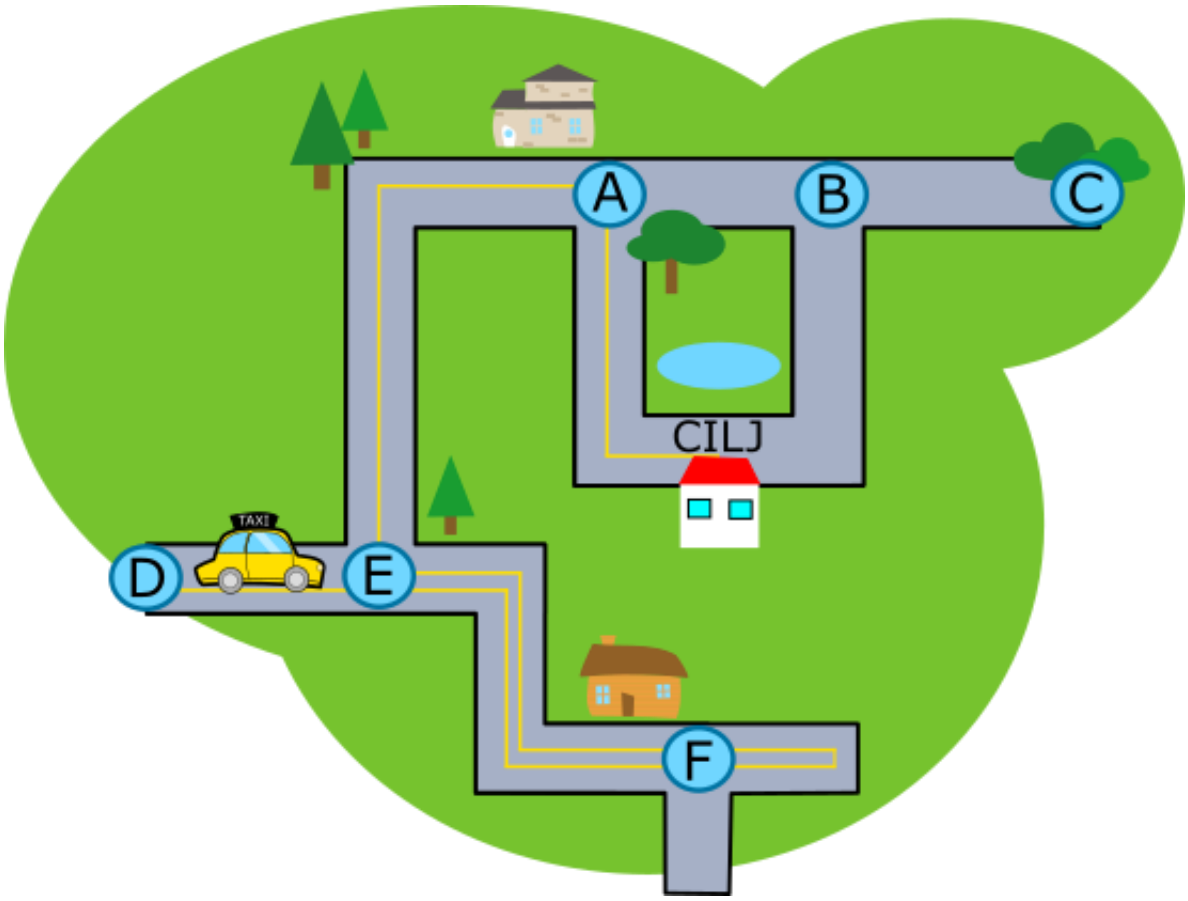


V kakšnem vrstnem redu bo taksi prevozil križišča, da bo prispel na cilj?

A. EFEAB
B. EFFEAB
C. EFFEA
D. EFFFEA
E. EFFFEAB

Rešitev

Taksi pelje skozi križišča EFFEA.



Računalniško ozadje

Naloga kaže, kako delovale navigacijske naprave, če bi nas vedno vodile le v smeri proti cilju. Kot si videl, to ne deluje dobro, zato takšne naprave v resnici uporabljajo veliko naprednejše algoritme, ki upoštevajo tako potek cest, kot kako hitro vožnjo omogočajo in celo, koliko gneče je v tem trenutku na njih.

Lahko najdeš zemljevid, kjer taksi z upoštevanjem pravil iz naloge ne bi nikoli prišel na svoj cilj?

Besedna sestavljanka

6. - 9. razred (državno)

Sandra je v angleškem časopisu našla besedno sestavljanko. Sestavljena je iz 9 črk, ki so zapisane v tabeli velikosti 3x3. Za rešitev uganke moraš iz teh črk sestaviti čim več angleških besed, ki:

- vsebujejo vsako črko iz tabele največ enkrat,
- vsebujejo vse označene črke in
- so dolge vsaj 4 črke.

Reševala je različne sestavljanke. Rešitev ene od njih je:

angel, challenge, clan, clean, eagle, glance, hall, heal, lane, lean.

Katera sestavljanka je to?

A)

h	l	n
e	c	g
l	e	a

B)

h	l	e
n	c	g
l	e	a

C)

h	l	n
b	c	g
a	e	a

D)

h	l	n
e	c	g
l	e	a

Rešitev

Pravilen odgovor je D, saj vse besede vsebujejo črki a in l.

Odgovor A ni pravilen, ker besede angel, eagle itd. ne vsebujejo črke c.

Odgovor B ni pravilen, ker besede eagle, heal in hall ne vsebujejo črke n.

Odgovor C ni pravilen, ker v sestavljanki ni dovolj črk e in l, da bi sestavili besedo challenge.

Računalniško ozadje

Računalniški programi so, čenekoliko poenostavimo, sestavljeni iz ukazov in pogojev, ki določajo, ali se ti ukazi izvedejo ali ne. Branje in razumevanje pogojev je torej ena od osnovnih veščin računalnikarjev.

Igra s frnikolami

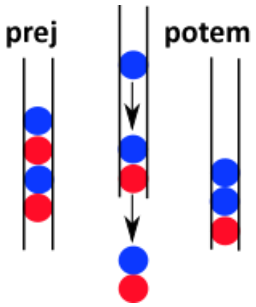
6. – 9. razred (državno)

Ernest preizkuša novo igro na svojem pametnem telefonu. Igra se začne z najmanj tremi barvnimi frnikolami v stolpcu, ki ga pred začetkom igre pripravi Ernest. Vsaka frnikola je ali rdeče ali modre barve.

Po pritisku na gumb GO se stanje stolpca frnikol spremeni tako, da spodnji dve frnikoli izpadeta, na vrh stolpca pa se dodajo nove frnikole glede na naslednji dve pravili:

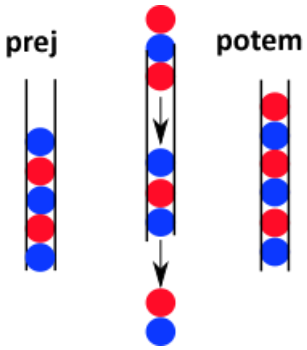
Če je prva izpadla frnikola **rdeča**,

se na vrhu stolpca pojavi ena modra frnikola.



Če je prva izpadla frnikola **modra**,

se na vrhu stolpca pojavijo tri nove frnikole: ena rdeča, ena modra in še ena rdeča.



Če so v stolpcu vsaj tri frnikole, lahko igro nadaljujemo s pritiskom na gumb GO. Igra se konča, če oz. ko sta v stolpcu le dve frnikoli ali manj.

Če Ernest pripravi začetni stolpec frnikol, ki ga prikazuje slika na desni, koliko klikov na gumb GO je potrebnih, da se igra konča?

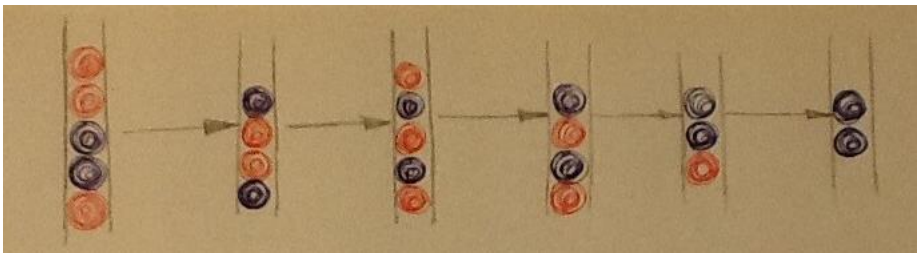
Rešitev

Pravilni odgovor je 5.

Rešitev lahko enostavno ugotovimo tako, da simuliramo potek igre. Po petih klikih na gumb GO nam v stolpcu ostaneta le dve modri frnikoli, zato se igra zaključi. Če barve frnikol v stolpcu zapišemo od zgoraj navzdol, dobimo naslednji potek igre:

RRMMR → MRRM → RMRMR → MRMR → MMR → MM

Ta potek igre prikazuje tudi spodnja slika.



Računalniško ozadje

Naloga predstavlja *Postov produkcijski sistem* (angleško *Post production system*), model računanja na nizih simbolov, ki ga je razvil Emil Leon Post v dvajsetih letih prejšnjega stoletja, prvič pa je bil objavljen leta 1943. Emil Leon Post (1897-1954) je bil na Poljskem rojeni ameriški matematik in logik, ki je veliko prispeval k razvoju teorije izračunljivosti.

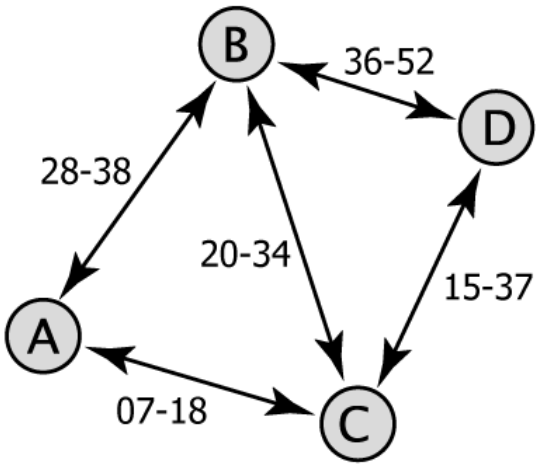
Model računanja temelji na prepisovanju nizov, kjer s pomočjo različnih formul (pravil) zamenjujemo podnize v nizih z drugimi nizi in tako dobimo nove nize. Na model lahko gledamo tudi kot na (končni) sistem objektov z (končno) množico relacij, ki definirajo možne manipulacije in transformacije podanih objektov.

V teoretičnem računalništvu take sisteme imenujemo *kontekstno neodvisne gramatike* (angleško *context-free grammars* ali *CFG*). Tako lahko na primer sistem množenj in seštevanj definiramo kot enostavno kontekstno neodvisno gramatiko z le nekaj pravili. Drug znan in zelo uporaben primer pa je uporaba primerne kontekstno neodvisne gramatike za točno in popolno definicijo programskega jezika, ki ga lahko uporabimo tako za namene izobraževanja kot tudi za njegovo implementacijo v računalnikih.

Načrtovanje poti

6. - 9. razred (državno)

Slika prikazuje železniške povezave med štirimi kraji A, B, C in D. Števili, ki sta zapisani na povezavi, povesta, koliko minut po polni uri vlaka na tej povezavi pripeljeta oz. odpeljeta (en v vsaki smeri). Urnik se vsako uro ponovi. Tako vlak iz smeri kraja A proti kraju B odpelje iz kraja A ob 8.28, 9.28, 10.28 itd. in prispe v kraj B po desetih minutah ob 8.38, 9.38, 10.38 itd. Tudi vlaki iz kraja B v kraj A odpeljejo iz B ob 8.28, 9.28, 10.28 itd. in prispejo v A deset minut kasneje.



Ana je ob 8:45 v kraju A in želi priti v kraj D. Ob kateri uri lahko najhitreje prispe?

Rešitev

Pravilen odgovor je 9.52.

Ob 9.07 gre na vlak, ki potuje proti kraju C in tja prispe ob 9.18. Nato ob 9.20 skoči na vlak, ki gre proti kraju B in tja prispe ob 9.34. Na koncu pa potuje z vlakom v smeri kraja D in tja prispe ob 9.52.

Računalniško ozadje

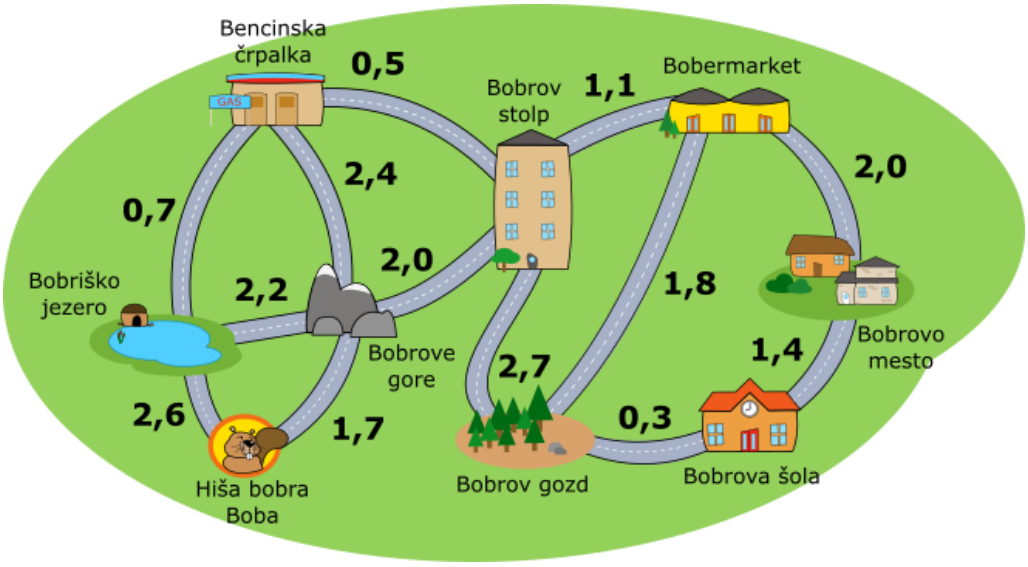
Je bilo reševanje naloge zapleteno? Najbrž ne preveč. Pa si predstavljaš, kako zapleteno bi postalo, če ne bi imel le štirih krajev, temveč bi jih bilo, na primer, sto? Za takšne primere smo si izmislili postopke, s katerimi sistematično poiščemo najhitrejšo pot.

Pot na večerjo

6. - 9. razred (državno)

Bober Bob želi obiskati prijatelje v Bobrovem mestu, ker so ga ob 19:00 povabili na večerjo.

- Bob želi pred tem kupiti darila za prijatelje v Bobermarketu.
- Preden gre v Bobrovo mesto, mora v Bobrovi šoli pobrati še sina Robija.
- V avtomobilskem rezervoarju dovolj goriva, da se lahko z njim vozi še 4 ure, zato se mora po poti ustaviti tudi na bencinski črpalki, da dotoči gorivo.



S polnim rezervoarjem lahko vozi 9 ur. Čas, ki ga potrebuje za vožnjo po posamezni cesti, je označen v urah. Trenutno je ura 10:00 in Bob mora pohiteti. Po kateri poti naj gre?

A. hiša → jezero → črpalka → stolp → gozd → šola → gozd → Bobermarket → mesto
B. hiša → gore → stolp → Bobermarket → gozd → šola → mesto
C. hiša → jezero → črpalka → stolp → Bobermarket → gozd → šola → mesto
D. hiša → gore → črpalka → stolp → gozd → Bobrova šola → mesto

Rešitev

Za vožnjo po poti C potrebuje 8,4 ure in prispe ob 18:24. Pot A traja 10,9, kar je preveč. B ne pelje do bencinske črpalke, zato Bobu na poti zmanjka goriva. Po poti D bi prispel do bencinske črpalke šele po 4,1 urah, ko bi mu že zmanjkalo goriva.

Računalniško ozadje

Iskanje optimalne poti ali optimalnega zaporedja opravil je tipična naloga za računalništvo. Običajno za te namene uporabljamo posebne algoritme, kot je Dijkstrin algoritem za iskanje najboljše poti.

Predor

Beni se je z družino odpravil v hribe in na poti našel tunel. Ker je predor zelo ozek in temen, lahko iz varnostnih razlogov istočasno skozenj potujeta le ena ali dve osebi, ki morata obvezno imeti s seboj svetilko. Družina ima s seboj le eno svetilko. Družinski člani skozi predor hodijo različno hitro: Beni potrebuje 5 minut, njegova sestra Ana dvakrat toliko, mama potrebuje 20 minut, oče pa 25 minut. Družina želi priti na drugo stran predora v eni uri.

Kako hitro je lahko cela družina na drugi strani predora?

A. 35 minut
B. 45 minut
C. 60 minut
D. Nemogoče je, da bi cela družina prečkala predor v eni uri.

Rešitev

Družina potrebuje šestdeset minut. Mama in oče morata predor prečkati hkrati, ne da bi se bilo treba komurkoli od njiju vračati. Tako morata najprej skozi predor Beni in Ana (10 minut), eden od njiju se vrne nazaj (5 ali 10 minut), nato prečkata mama in oče (25 minut), drugi od otrok se vrne (10 ali 5 minut) in še enkrat skupaj prečkata predor (10 minut).

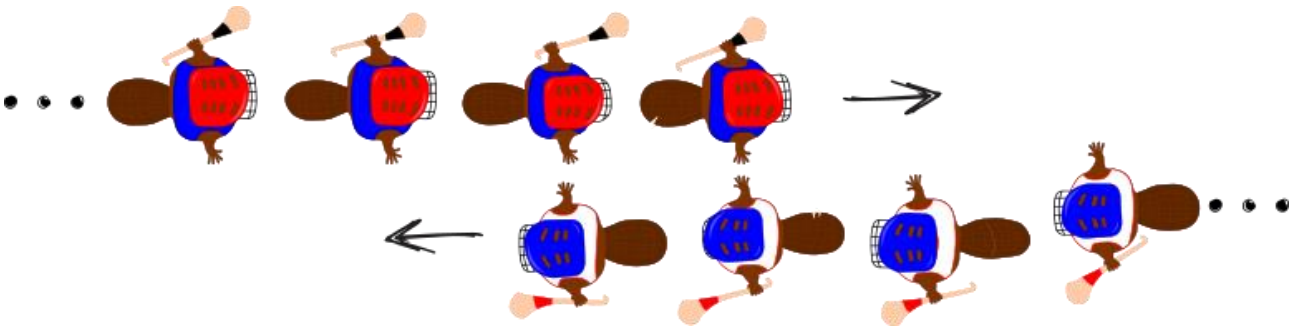
Računalniško ozadje

Računalničarji pogosto iščejo rešitve z znanimi omejitvami, pri čemer morajo problem rešiti, kolikor hitro in preprosto je to mogoče. Temu rečemo optimizacija.

Rokovanje igralcev hurlinga 6. - 9. razred (državno)

Bobri so zelo dobri v irski igri po imenu hurling. Ko se igra hurlinga konča, se igralci vsake ekipe eden za drugim postavijo v svojo vrsto in se sprehodijo mimo igralcev druge ekipe. Ko grejo mimo igralca nasprotne ekipe, se z njim rokujejo in rečejo »Hvala za igro!«

Na začetku se rokujeta le prva igralca obeh ekip. Nato si istočasno sežejo v roke prvi igralec ene ekipe in drugi igralec druge ekipe ter drugi igralec prve ekipe in prvi igralec druge ekipe (kot je vidno na sliki). To se nadaljuje, dokler se vsak igralec iz ene ekipe ne rokuje z vsemi igralci nasprotne ekipe.



Pri hurlingu je v vsaki ekipi 15 igralcev. Če vsak igralec potrebuje 1 sekundo, da se rokuje in premakne k naslednjemu igralcu, koliko sekund rokovanja bo po igri?

Rešitev

Pravilen odgovor je devetindvajset. Število rokovanj je identično seštevku dolžin obeh vrst minus ena.

Računalniško ozadje

V računalništvu ves čas poskušamo pisati hitrejše in hitrejše programe. Ko računamo, koliko časa bo program potreboval za določeno opravilo, razmišljamo na podoben način, kot si moral razmišljati ob reševanju te naloge.

Štirje opravki

6. - 9. razred (državno)

Aleksandra želi med odmorom (12:00 – 13:00) oditi po naslednjih opravkih:

- kupiti knjigo v knjigarni,
- kupiti steklenico mleka v trgovini,
- poslati pravkar kupljeno knjigo na pošti in
- spiti skodelico kave v kavarni.

Kraj	Trajanje	Gneča
Knjigarna	15 min	12:40 – 13:00
Trgovina	10 min	12:00 – 12:40
Pošta	15 min	12:00 – 12:30
Kavarna	20 min	12:30 – 12:50

Aleksandra je ocenila, koliko časa bo potrebovala za posamezen opravek, vendar ti časi veljajo le, kadar ni gneče, zato se želi izogniti gneči.

Pomagaj Aleksandri razvrstiti njena opravila tako, da se bo zagotovo izognila gneči. Kakšen je pravilni vrstni red opravkov, da bo vse opravke zaključila med odmorom?

Rešitev

Kavarna, knjigarna, pošta, trgovina.

Naloga vsebuje številne omejitve. Aleksandra mora knjigarno obiskati pred 12:40, trgovino po 12:40, na pošto gre lahko šele za tem, ko je obiskala knjigarno. Pošto mora obiskati po 12:30. Kavarno lahko obišče pred 12:30, ker po 12:50 ne bo imela več dovolj časa v svojem odmoru.

	Trajanje	12:00 – 12:05	12:05 – 12:10	12:10 – 12:15	12:15 – 12:20	12:20 – 12:25	12:25 – 12:30	12:30 – 12:35	12:35 – 12:40	12:40 – 12:45	12:45 – 12:50	12:50 – 12:55	12:55 – 13:00
Knjigarna	15 min					X	X	X					
Trgovina	10 min											X	X
Pošta	15 min								X	X	X		
Kavarna	20 min	X	X	X	X								

Ob upoštevanju vseh omejitev ugotovimo, da mora Aleksandra kavarno obiskati med 12:00 – 12:20, knjigarno 12:20 – 12:35, pošto 12:35 – 12:50 in trgovino 12:50 – 13:00.

Računalniško ozadje

Razvrščanje procesov je ena ključnih nalog operacijskega sistema, saj brez tega uporabniki ne bi mogli uporabljati več programov hkrati. Delu sistema, ki določa, kateri procesi se bodo izvajali kdaj in koliko dolgo, rečemo razvrščevalnik (ang. scheduler).

Večerja

Bobrovka Sara želi organizirati večerjo. Zato mora govoriti s petimi prijatelji: z Anico, Borutom, Cenetom, Darinko in Erikom.

1. Z Erikom lahko govori takoj.
2. Preden govori z Darinko, mora govoriti z Anico.
3. Preden govori z Borutom, mora govoriti z Erikom.
4. Preden govori s Cenetom, mora govoriti z Borutom in Darinko.
5. Preden govori z Anico, mora govoriti z Borutom in Erikom.

V kakšnem vrstnem redu naj pokliče prijatelje?

Rešitev

Pravilen odgovor je Erik – Borut – Anica – Darinka - Cene.

Pri nalogi moramo biti pozorni na pogoje, ki so podani.

- Erik ni odvisen od drugih in zato ga kliče prvega.
- Borut je odvisen le od Erika in zato je klican kot drugi.
- Anica je odvisna od Boruta ter Erika in je zato klicana kot tretja.
- Darinka je odvisna od Anice in je zato klicana kot naslednja.
- Cene pa je odvisen tako od Anice kot od Darinke in je zato klican zadnji.

Računalniško ozadje

Naloga spominja na urejanje, le da ne gre za urejanje po abecedi, velikosti ali čem podobnem. Naloga podaja le, kdo mora biti pred kom. Takšne naloge - na katere v resnici pogosto naletimo tako v računalništvu kot v vsakdanjem življenju - nimajo nujno le ene same možne rešitve: zgodi se lahko, da je s pogoji skladnih več možnih zaporedij.

Učeno ime za takšno urejanje je *topološko urejanje*.

Stanko je navdušen zbiratelj skodelic z imeni mest z vseh koncev sveta. Ker je vikend in ima čas, se je odločil, da ponovno pregleda svojo bogato kolekcijo in skodelice razvrsti glede na barvo in celino, s katere prihaja skodelica.

Po napornem preštevanju in razvrščanju skodelic v kategorije si je vzel premor. Zapiske je pustil na mizi in odšel na zaslužen počitek. Tačas pa je zapise našla njegova nagajiva sestrica Anja in spremenila natančno eno vrednost v tabeli, ki je prikazana spodaj.

	rdeča	rumena	zelena	modra	rjava	skupno
Azija	2	1	0	2	2	7
Evropa	0	1	1	2	2	6
Severna Amerika	1	2	3	0	1	6
Južna Amerika	0	1	2	1	0	4
Afrika	1	0	0	0	0	1
Oceanija	0	2	1	1	0	4
skupno	4	7	6	6	5	28

Ali lahko najdeš številko, ki jo je spremenila Anja, in jo popraviš na prvotni zapis?

A. Nikakor ne moremo najti spremenjene vrednosti. Stanko mora ponovno prešteti in razvrstiti skodelice.
B. Število rumenih skodelic iz Severne Amerike bi moralo biti 1.
C. Število rumenih skodelic iz Evrope bi moralo biti 0.
D. Število zelenih skodelic iz Severne Amerike bi moralo biti 2.

Rešitev

Pravilen odgovor je D. Število zelenih skodelic iz Severne Amerike bi moralo biti 2.

	rdeča	rumena	zelena	modra	rjava	skupno
Azija	2	1	0	2	2	7
Evropa	0	1	1	2	2	6
Severna Amerika	1	2	3	0	1	6
Južna Amerika	0	1	2	1	0	4
Afrika	1	0	0	0	0	1
Oceanija	0	2	1	1	0	4
skupno	4	7	6	6	5	28

Če seštejemo vrednosti za vsak kategorijo, ugotovimo, da je vsota pri stolpcu »zelena« in vrstici »Severna Amerika« napačna.

zelena: 0+1+3+2+0+1=7

Severna Amerika: 1+2+3+0+1=7

Pri primerjavi vsot omenjenega stolpca in vrstice opazimo, da je vsota za 1 manjša kot bi morala biti. Torej je potrebno vrednost pri zelenih skodelicah iz Severne Amerike zmanjšati za 1, torej tam zapisati 2, ker 3-1=2.

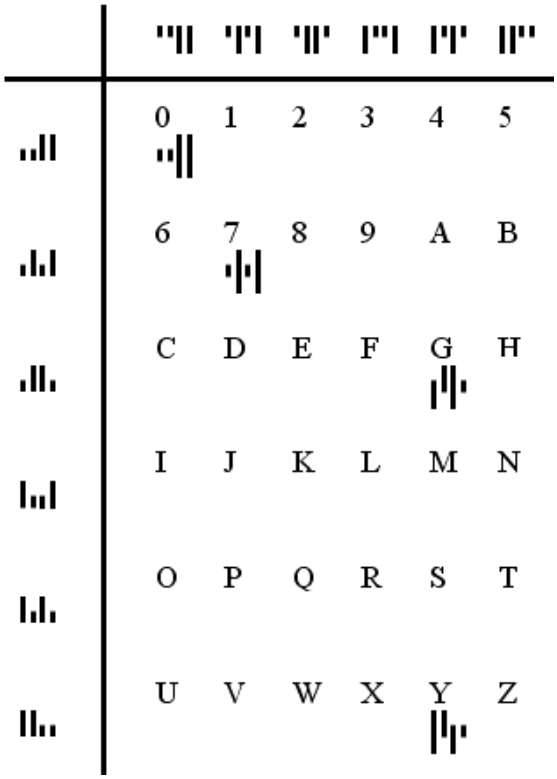
Računalniško ozadje

Tako preverjanje je pogosto uporabljeno pri shranjevanju ali prenašanju podatkov. Tako računalniki ugotovijo, ali je pri prenosu oz. shranjevanju prišlo do napake. Če računalnik ugotovi napako, zahteva ponovno izvršitev.

KIX koda

Poštni urad je izumil KIX kodo, ki olajša branje poštnih oznak z računalniki.

V oznakah se pojavljajo številke in znaki angleške abecede. Zložili so jih v tabelo. Vsak znak v KIX kodi je predstavljen s štirimi črtami. Njihovi gornji deli povedo, v kateri vrstici se nahaja znak. Spodnji deli povedo stolpec.



Oznako G7Y0 zapišejo kot  .

Katero kodo pa predstavljajo črte  ?

Rešitev

BC16

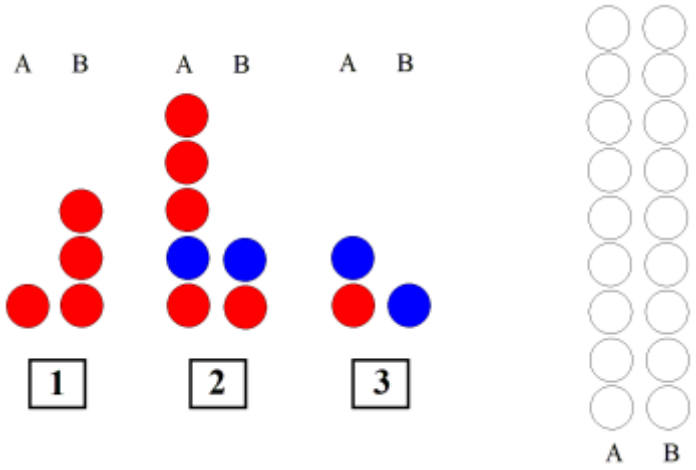
Računalniško ozadje

Ta naloga sploh ni izmišljena! Natančno takšne kode v resnici uporablja nizozemska pošta. Le malenkost drugačne kode uporabljajo tudi v Angliji, Singapurju, Kanadi, Avstraliji, ZDA ...

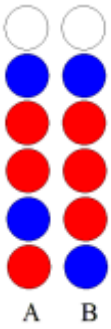
Padajoče frnikole

6. – 9. razred, srednja šola (državno)

Imamo tri oštevilčene pare stolpcev barvnih frnikol. Kadar kliknemo na gumb s številko, pade v stolpca A in B na desni ustrezno število frnikol, vedno v vrstnem redu, ki je prikazan na sliki. Na začetku sta stolpca prazna.



Če na primer kliknemo najprej gumb 3, nato gumb 1 in na koncu ponovno gumb 3, bomo v obeh stolpcih na desni dobili frnikole, ki jih prikazuje spodnja slika.



Kako dolgo je najkrajše zaporedje klikov na gumbe, da v obeh stolpcih na desni dobimo povsem enako neprazno zaporedje barvnih frnikol?

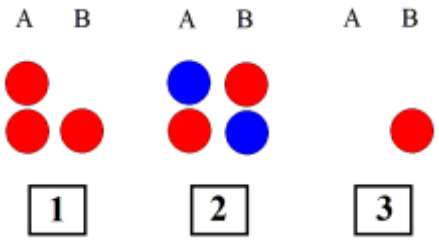
Rešitev

Pravilni odgovor je 4. Zaporedje klikov na gumbe je 2, 1, 1 in 3.

Iskana rešitev se mora začeti s klikom na gumb 2 (gumb 3 da frnikoli različnih barv, gumb 1 pa ustvari zaporedje preveč rdečih frnikol v stolpcu B). Temu lahko sledi le gumb 1, ki ga pritisnemo dvakrat (gumba 2 ali 3 bi dodala modre frnikole, zato nista primerna). Na koncu s klikom na gumb 3 zaključimo oba stolpca frnikol.

Zaporedje klikov na gumbe (2, 1, 1, 3) je najkrajša rešitev podanega problema, vendar obstajajo tudi druge rešitve, ki vključujejo ponovitve tega zaporedja, kot je na primer (2, 1, 1, 3, 2, 1, 1, 3). Če torej najdemo rešitev, lahko poiščemo tudi neskončno ponavljajočih se rešitev (a le ena rešitev je najkrajša).

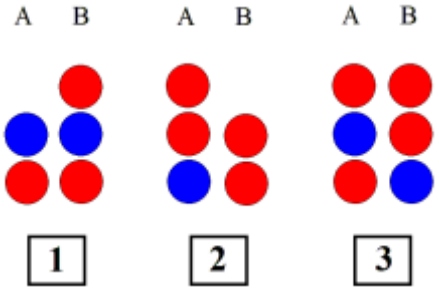
Poglejmo si še drugo različico tega problema, pri kateri pa imamo dejansko neskončno rešitev, ki se ne ponavljajo. Slika prikazuje tri gumbe v novi različici:



V tej različici gumb 3 spusti frnikolo le v stolpec B. V tem primeru je rešitev vsako zaporedje oblike (3, *n*-krat 2, 1), poleg tega pa rešitev predstavljajo tudi njegove ponovitve. Vendar sta najkrajši rešitvi le dve: (3, 1) in (1, 3).

V splošnem lahko rečemo, da zaporedje dveh rešitev da novo rešitev.

Za konec pa si poglejmo še različico problema, kjer rešitev ne obstaja:



Računalniško ozadje

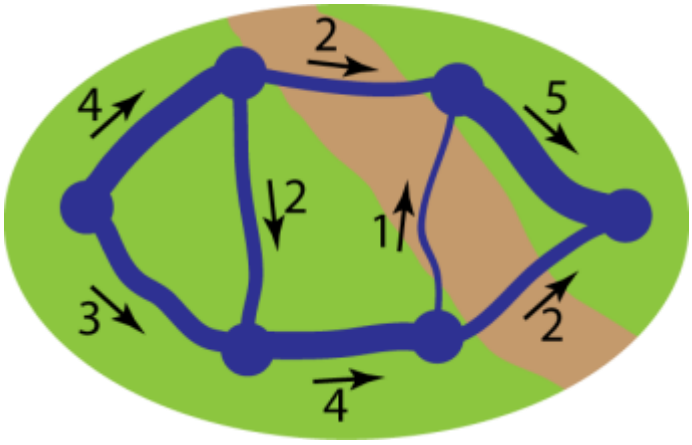
V nalogi opisan problem je rešljiva različica *Postovega korespondenčnega problema* (angleško *Post correspodence problem*), ki ga je leta 1946 predstavil Emil Leon Post. Postov korespondenčni problem spada med tako imenovane neodločljive probleme. Izkaže se kot izjemno uporaben pri dokazovanju neodločljivosti ter pri določanju izračunljivosti v teoriji formalnih jezikov.

Problem, ki je opisan v nalogi, je neodločljiv problem, ker v splošnem pri podani poljubni množici parov stolpcev računalnik ne more izračunati oziroma odločiti, ali lahko sestavimo dva enaka stolpca frnikol ali ne.

Emil Leon Post (1897-1954) je bil na Poljskem rojeni ameriški matematik in logik, ki je veliko prispeval k razvoju teorije izračunljivosti.

Prevoz hlodov

Po sistemu kanalov transportiramo hlode. Ob vsakem kanalu je označeno, koliko hlodov na minuto lahko spravimo po njem in v katero smer. Koliko hlodov na minuto je mogoče spraviti s skrajne leve na desno točko?



Sistem kanalov prečka neko suho področje. Ne oziraj se nanj. (Ali pač?)

Rešitev

Pet hlodov. Ozko grlo je ravno suho področje. Očitno lahko prek njega spravimo največ pet hlodov na minuto – a še to le, če lahko do gornje točke pred tem področjem spravimo vsaj dva hloda (lahko) in do spodnje vsaj tri hlode (lahko). Tudi desno od tega področja se vse lepo izide.

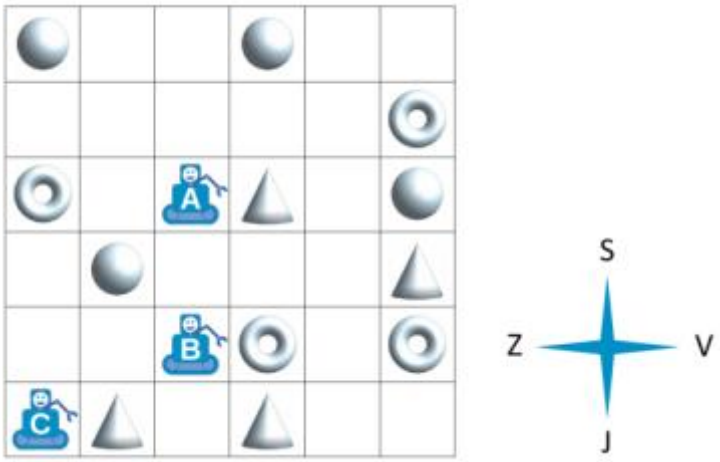
Računalniško ozadje

V nalogi si reševal enega najbolj znanih računalniških problemov: problem iskanja največjega pretoka. Izrek, ki je enako slaven kot problem, pravi, da je največji možni pretok enak najmanjšemu prerezu. Najmanjši prerez je tak »prerez« grafa, ki najbolj omeji pretok. Če ga poiščemo (in tule je bil »slučajno« že označen), vemo tudi, kakšen je največji možni pretok.

Sočasni premiki

V skladišču roboti vedno delajo v skupini. Ko skupina dobi navodilo za premik v neko smer (to je lahko S, J, V ali Z), se vsi roboti v skupini sočasno premaknejo za en kvadrat v zahtevani smeri. Ko roboti izpolnijo zaporedje podanih premikov, vsak od njih pobere tisto stvar, ki se nahaja v kvadratu robota.

Tako bi na primer v spodnjem primeru skupine treh robotov, če ji podamo zaporedje ukazov S, S, J, J, V, robot A pobral stožec, robot B obroč, robot C pa tudi stožec.



Kakšno zaporedje ukazov moramo podati skupini robotov, da bo skupina pobrala natanko eno kroglo, en stožec in en obroč?

A. S, V, V, V
B. S, V, V, J, V
C. S, S, J, V, S
D. S, V, V, J, Z

Rešitev

Skupini moramo podati ukaze S, V, V, J, V.
Poglejmo, kaj poberejo roboti, če izvedejo podane ukaze.

A. Če damo skupini zaporedje premikov S, V, V, V, potem bo robot A pobral obroč, robot B stožec, robot C pa bo tudi pobral obroč. Noben od njih ne bo pobral krogle, zato ta odgovor ni pravilen.
B. Če damo skupini zaporedje premikov S, V, V, J, V, potem bo robot A pobral kroglo, robot B obroč, robot C pa bo pobral stožec. Torej bo skupina pobrala natanko vsakega od treh objektov in odgovor je pravilen.
C. Če damo skupini zaporedje premikov S, S, J, V, S, potem bo robot A pobral kroglo, robot B stožec, robot C pa bo tudi pobral kroglo. Noben od njih ne bo pobral obroča, zato ta odgovor ni pravilen.
D. Če damo skupini zaporedje premikov S, V, V, J, Z, potem bo robot A pobral stožec, robot B obroč, robot C pa bo tudi pobral stožec. Noben od njih ne bo pobral krogle, zato tudi ta odgovor ni pravilen.

Računalniško ozadje

Kadar roboti ali pa računalniki delajo skupaj sočasno, rečemo, da delajo vzporedno.

Če imamo manjše število robotov ali računalnikov, ki delajo vzporedno, bi lahko vsakemu od njih dali drugačna navodila za delo. Vendar pri večjem številu računalnikov, če imamo tisoč ali milijon računalnikov, ki delujejo skupaj, pa pisanje navodil za vsakega posebej ne bi bilo praktično. Zato v takih primerih damo večji skupini računalnikov enaka navodila za delo. Tako ima na primer superračunalnik Tianhe-2 3 milijone ločenih računskih jeder, ki lahko delajo skupaj za rešitev enega bolj kompleksnega problema.

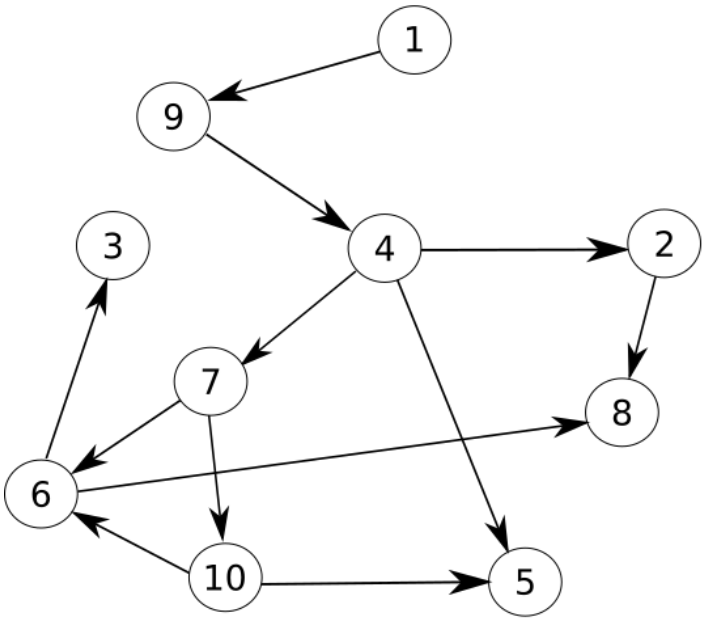
V naši nalogi bi lahko naleteli še na dodatno težavo. Ker roboti delajo v istem prostoru, moramo pri pripravi navodil paziti tudi na to, da se ne zaletijo med seboj ali pa kako drugače onemogočijo delovanje drugega robota. Zato je priprava vzporednih navodil v takem okolju izjemno zahtevna. V računalništvu se s tem ukvarja posebno področje, imenovano vzporedno programiranje.

Pravo jutranje zaporedje

8. - 9. razred, srednja šola

Rok je prespal budilko in se zbudil kasneje kot običajno, zato se mu je že mudilo v šolo. In to ravno danes, ko gredo s šolo v gledališče in mora izgledati karseda imenitno!

Preden se v celoti obleče, mora Rok narediti 10 opravkov. Nekatere opravke lahko naredi sočasno in morda nadoknadi nekaj prespanega časa ter še pravočasno prispe do šole. Po drugi strani pa so nekateri opravki taki, da mora z njimi počakati, dokler se drugi opravki ne zaključijo. Primer takega opravka je obuvanje čevljev, ki ga lahko naredi šele takrat, ko že ima oblečene hlače in obute nogavice.



Obstaja več načinov, v kakšnem zaporedju se Rok obleče za šolo. Katero od navedenih zaporedij opravkov **ni** pravi način za uspešno oblačenje za šolo?

A. 1, 9, 4, 2, 7, 10, 6, 8, 3, 5
B. 1, 9, 4, 7, 10, 6, 2, 5, 8, 3
C. 1, 9, 4, 7, 6, 5, 2, 10, 8, 3
D. 1, 9, 4, 7, 2, 10, 6, 3, 5, 8

Rešitev

Zaporedje pod C je nepravilno. Opravilo 10 se mora zaključiti, preden se lahko prične opravilo 6. Vsa ostala zaporedja opravil so pravilna.

Računalniško ozadje

Pri nalogah je zelo pogosto, da imajo neke predpogoje (naloge, ki morajo biti opravljene, preden lahko začnemo opravljati dano nalogo). Običajen tak primer je, kadar se moramo obleči: spodnje perilo moramo obleči, preden lahko oblečemo hlače, pa tudi majico moramo obleči, preden nataknemo še jakno. Po drugi strani pa ni važno, ali obujemo nogavice preden oblečemo hlače ali po tem. Kadar so le nekatere naloge odvisne od drugih, tvorijo tako imenovano delno urejenost, ki jo pogosto predstavimo z usmerjenim acikličnim grafom, v katerem je vsaka naloga eno vozlišče, vsak predpogoj pa je ena povezava. Če bi imel tak graf cikle, potem nalog ne bi bilo mogoče opraviti. Zaporedje nalog, ki zadošča vsem predpogojem, lahko poiščemo s pomočjo algoritma, ki se imenuje topološko urejanje. Algoritem hrani seznam vseh vozlišč, katera imajo izpolnjene vse predpogoje, izbere enega izmed njih in na seznam doda vsa vozlišča, ki so v tem koraku izpolnila vse predpogoje. To je primer požrešnega algoritma, ki v vsakem koraku lahko enostavno določi svojo izbiro.

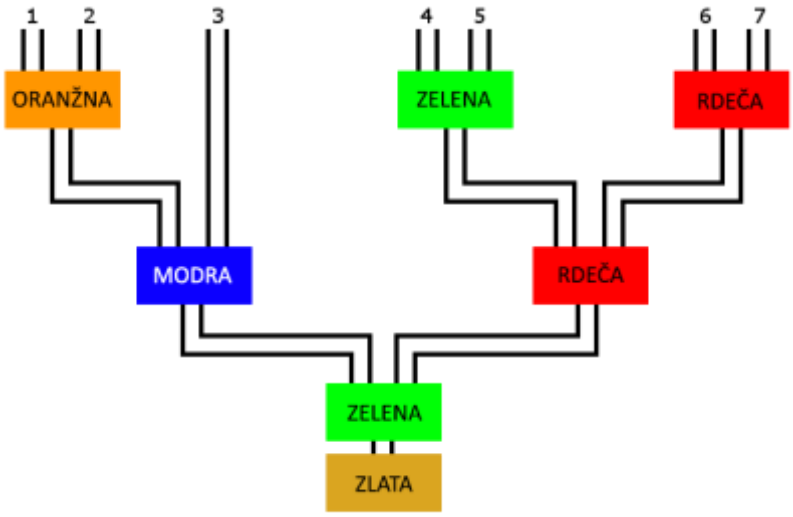
Zlata frnikola

8. - 9. razred, srednja šola

Bor se zelo rad igra s frnikolami. Ima le črne in bele frnikole, a v bližnjem nakupovalnem središču imajo avtomat, v katerem lahko dobi tudi posebno zlato frnikolo. Da bi lahko dobil to zlato frnikolo, mora v sedem vhodnih odprtin avtomata vstaviti črne in bele frnikole. V avtomatu je več vrat različnih barv: modra, rdeča, oranžna in zelena; vsaka vrata na vhod sprejmejo dve frnikoli in prepustijo skozi le eno frnikolo. Če do zlatih vrat na dnu avtomata pride črna frnikola, se na izhodu avtomata pojavi zlata frnikola.

Štiri vrata v avtomatu delujejo po naslednjih pravilih:

- Modra vrata: če sta obe frnikoli na vhodu črni, da na izhod črno frnikolo. Če sta obe beli, na izhod ne pride nobena frnikola. Če pa sta frnikoli na vhodu različnih barv, pride na izhod bela frnikola.
- Rdeča vrata: prepustijo na izhod črno frnikolo le v primeru, če sta obe frnikoli na vhodu črni. V vseh ostalih primerih pride na izhod bela frnikola.
- Zelena vrata: prepustijo na izhod belo frnikolo le v primeru, če sta obe frnikoli na vhodu beli. V vseh ostalih primerih pride na izhod črna frnikola.
- Oranžna vrata: prepustijo na izhod belo frnikolo, če sta obe frnikoli na vhodu beli. Če sta obe črni, na izhod ne pride nobena frnikola. Če sta frnikoli na vhodu različnih barv, pride na izhod črna frnikola.



Kakšno kombinacijo frnikol moramo vstaviti v vhod avtomata, da dobimo zlato frnikolo?

A. 1: bela, 2: bela, 3: črna, 4: črna, 5: bela, 6: črna, 7: črna
B. 1: bela, 2: bela, 3: črna, 4: bela, 5: bela, 6: črna, 7: črna
C. 1: bela, 2: bela, 3: bela, 4: črna, 5: bela, 6: bela, 7: črna
D. 1: črna, 2: bela, 3: bela, 4: črna, 5: bela, 6: črna, 7: bela

Rešitev

1: bela, 2: bela, 3: črna, 4: črna, 5: bela, 6: črna, 7: črna (odgovor A).

Poglejmo si vse možne odgovore.

V primeru A oranžna vrata dobijo dve beli frnikoli in na izhod pride bela frnikola. Ta gre na vhod modrih vrat skupaj s črno frnikolo, tako da je izhod teh vrat bela frnikola. Na desni pa prva zelena vrata dobijo belo in črno frnikolo in izhod je črna frnikola. Prva rdeča vrata dobijo dve črni frnikoli, izhod pa je tudi črna frnikola. Izhod zelenih vrat (črna) in rdečih vrat (črna) je vhod drugih rdečih vrat, zato na izhodu teh dobimo črno frnikolo. Zadnja zelena vrata torej dobijo na vhod dve črni frnikoli in na izhod dajo tudi črno frnikolo. Ta pride na vhod zlatih vrat in na izhod dobimo zlato frnikolo.

V primeru B oranžna vrata dobijo dve beli frnikoli in na izhod pride bela frnikola. Ta gre na vhod modrih vrat skupaj s črno frnikolo, tako da je izhod teh vrat bela frnikola. Na desni pa prva zelena vrata dobijo dve beli frnikoli in izhod je bela frnikola. Prva rdeča vrata dobijo dve črni frnikoli, izhod pa je tudi črna frnikola. Izhod zelenih vrat (bela) in rdečih vrat (črna) je vhod drugih rdečih vrat, zato na izhodu teh dobimo belo frnikolo. Zadnja zelena vrata torej dobijo na vhod dve beli frnikoli in na izhod dajo tudi belo frnikolo. Ta pride na vhod zlatih vrat, a ker ni črne barve, ne dobimo zlate frnikole.

V primeru C oranžna vrata dobijo dve beli frnikoli in na izhod pride bela frnikola. Ta gre na vhod modrih vrat skupaj z belo frnikolo, tako da na izhod teh vrat ne pride nobena frnikola. Tako imajo zadnja zelena vrata le eno frnikolo na vhodu in torej ne morejo dati nobenega izhoda. Do zlatih vrat tako ne pride nobena frnikola.

V primeru D oranžna vrata dobijo belo in črno frnikolo in na izhod pride črna frnikola. Ta gre na vhod modrih vrat skupaj z belo frnikolo, tako da je izhod teh vrat bela frnikola. Na desni pa prva zelena vrata dobijo črno in belo frnikolo in izhod je črna frnikola. Prva rdeča vrata dobijo črno in belo frnikolo, izhod pa je bela frnikola. Izhod zelenih vrat (črna) in rdečih vrat (bela) je vhod drugih rdečih vrat, zato na izhodu teh dobimo belo frnikolo. Zadnja zelena vrata torej dobijo na vhod dve beli frnikoli in na izhod dajo tudi belo frnikolo. Ta pride na vhod zlatih vrat, a ker ni črne barve, ne dobimo zlate frnikole.

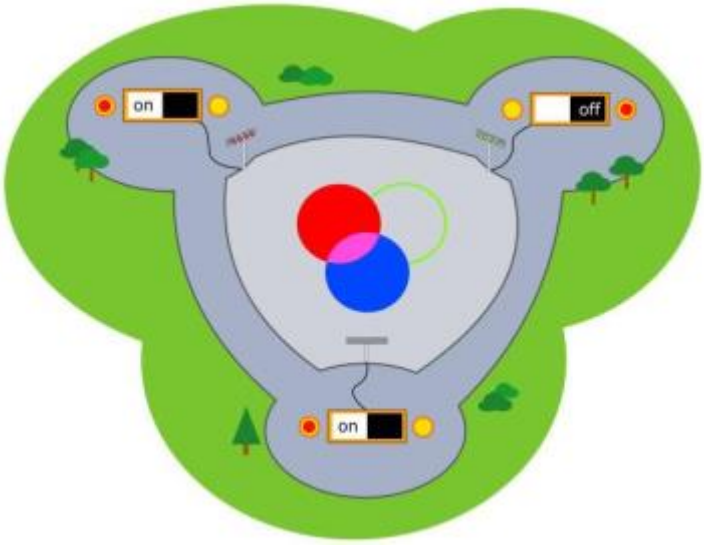
Računalniško ozadje

Barvna vrata v nalogi delujejo kot logična vrata, ki so osnovni gradniki vsakega procesorja v vsakem računalniku.

Frankova stikala

državno, 8. – 9. razred, srednja šola

Frank (ki začuda ni bil bober) je razsvetlil najvišjo stolpnico v mestu z lučmi, katerih barvo lahko spreminja s tremi stikali. Vsaka kombinacija stikal da drugo barvo. Ko so vsa stikala ugasnjena, je stavba v temi. Nekoliko ponesrečeno pa je stikala postavil tako, da so med seboj oddaljena po en kilometer.



Na začetku stoji pri prvem stikalu. Vsa stikala so ugasnjena. Koliko kilometrov bo moral prepotovati, da bo preskusil vse možne kombinacije?

Rešitev

	off off off
	off off **on**
1 km	off **on** on
2 km	off on **off**
3 km	**on** on off
4 km	on on **on**
5 km	on **off** on
6 km	on off **off**

Različnih kombinacij je osem. Če bi jih spreminjal tako, kot, recimo, običajno štejemo po dvojiško in bi stanju off-on-on sledil on-off-off, bi za preklop med njima prepotoval dva kilometra (prestavi stikalo, pri katerem trenutno stoji, nato pa mora potovati še do ostalih dveh).

Iščemo rešitev, pri kateri naštejemo vseh osem kombinacij v takšnem vrstnem redu, da se dve zaporedni kombinaciji razlikujeta le v enem stikalu. Takšno rešitev kaže tabela na levi. Ker mu za prvi preklop ni potrebno potovati nikamor, preskus vseh kombinacij zahteva šest kilometrov.

Računalniško ozadje

Naloga torej v bistvu sprašuje, ali je mogoče našteti vsa števila od 0 do 7 v dvojiškem zapisu tako, da se zaporedni števili razlikujeta le v enem bitu.

Frank, ki ni bil bober temveč ameriški fizik, se je pisal Gray in takšnemu zaporedju pravimo Grayevo kodiranje. Srečamo ga na mnogih popolnoma nepričakovanih mestih.

Takšno zaporedje obstaja za poljubno število bitov, ne le za tri, kot v tej nalogi. Ker je to prav zanimivo, si oglejmo, zakaj. Tule so zaporedja za 1 – 5 bitov.

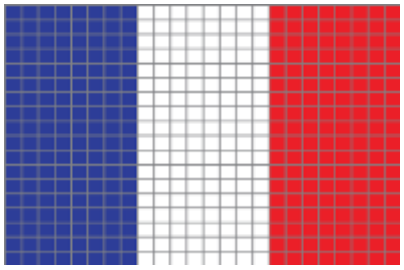
1 bit	0 1
2 bita	00 01 11 10
3 biti	000 001 011 010 110 111 101 100
4 biti	0000 0001 0011 0010 1010 1011 1001 1000
5 bitov	00000 00001 00011 00010 01010 01011 11000 11001 11011 11010 10010 10011 01001 01000 10001 10000

O prvi vrstici se ni kaj pogovarjati. Vsako naslednjo vrstico pa dobimo iz vrstice pred njo tako, da jo dvakrat ponovimo: prvič od leve proti desni, na začetek pa postavimo 0 (iz, recimo, 000 001 011 010 tako dobimo 0000 0001 0011 0010), drugič pa od desne proti levi, na začetek pa prilepimo 1 (000 001 011 010 obrnemo v 010 011 001 000 in prilepimo 1, s čimer dobimo 1010 1011 1001 1000). Zakaj se bodo zaporedni elementi vedno razlikovali le v enem bitu? No, kje pa bi lahko prišlo do razlike v dveh bitih?! Preverimo le zaporedje v četrti vrstici, pa bo hitro jasno, kaj se dogaja. Predpostavimo, da je zaporedje v tretji vrstici pravilno in se elementi razlikujejo le v enem bitu. Tedaj je pravilen tudi prvi del zaporedja v četrti: vsem smo prilepili ničlo, torej se spreminja le bit, ki se je spreminjal že v prvi vrstici. Pravilen je tudi drugi del zaporedja, kjer smo povsod prilepili enico. Do napake bi lahko prišlo le v sredini, na prelomu med polovicama, saj smo enemu prilepili ničlo, drugemu enico. Prav tam pa se na obeh straneh pojavi zadnje število iz prejšnje vrstice (010), torej se je spremenil samo dodatno prilepljeni bit.

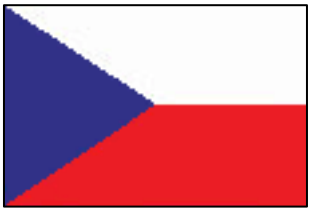
Na ta način lahko iz vsake vrstice pridobimo naslednjo. Frank bi lahko uporabil poljubno število stikal in še vedno poiskal zaporedje, ki preskusi vse kombinacije tako, da v vsaki spremeni le en bit.

Opisi zastav

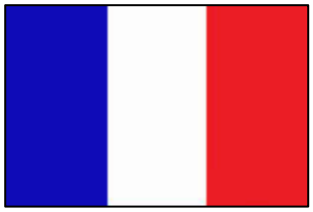
Zastave bomo opisovali tako, da bomo prek njih narisali mrežo z 18 vrsticami in 24 stolpci ter zapisali barvo vsakega polja. Opis bomo skrajšali tako, da bomo za zaporedna polja iste barve zapisali prve tri črke barve in število zaporednih polj te barve. Prva vrstica francoske zastave je tako opisana z [mod, 8][bel, 8][rde, 8]. Ker je francoska zastava kar preprosta, so vse naslednje vrstice enake tej. Celoten opis je torej sestavljen iz osemnajstih ponovitev gornjega opisa vrstice.



Razvrsti spodnje zastave glede na dolžino njihovih opisov od najkrajšega do najdaljšega.

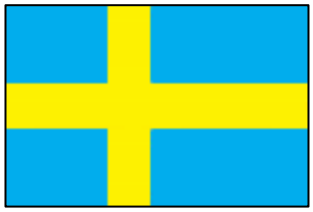


Češka



Francija

Nemčija



Švedska

Rešitev

Nemčija, Češka, Švedska, Francija. Pri nemški zastavi je vsaka vrstica sestavljena le iz ene barve, pri češki iz dveh, pri švedski iz dveh ali treh, pri francoski pa iz treh.

Računalniško ozadje

Čim krajši zapis slike je pomemben, da lahko na kartico fotoaparata ali v pomnilnik telefona shranimo čim več fotografij in video posnetkov ter da lahko slike in filme čim hitreje prenašamo prek interneta. Postopkov stiskanja je veliko. Nekateri pri tem sliko nekoliko pokvarijo, a zato zelo dobro stisnejo (takšen je format JPG oz. JPEG in vsi praktično uporabljani zapisi za video), pri drugih ostane slika enake kvalitete, datoteka s sliko pa je zato večja.

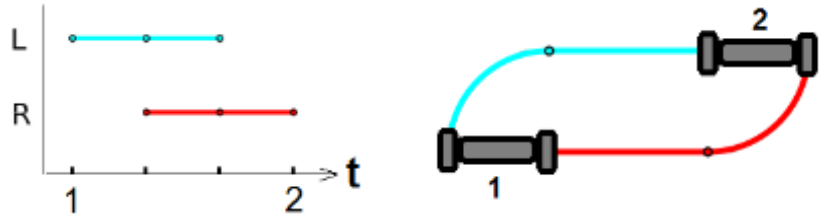
Oblika, ki jo predstavlja naloga, se imenuje »kodiranje z dolžino čet« (*run-length encoding*, RLE). To obliko zapisa so uporabljali telefaksi (ki jih danes lahko najdemo predvsem v muzejih), poleg tega pa se uporablja v nekaterih oblikah datotek PDF.

Segway

državno, 8. – 9. razred, srednja šola

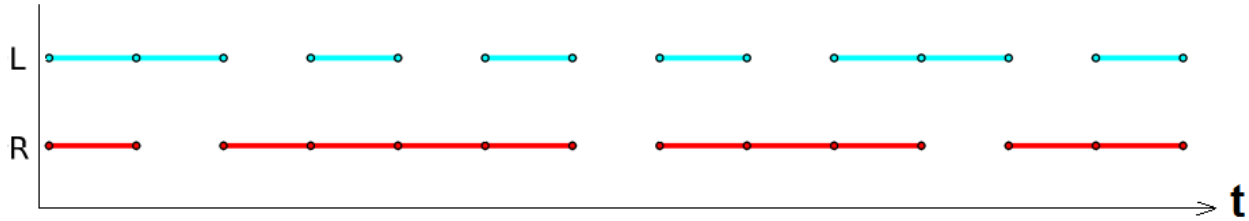
Jan se vozi z vozilom, podobnim Segwayu. Ta ima dve tipki. Če pritisne obe tipki, se vrtita obe kolesi in Segway se pelje naravnost naprej. Če pritisne le modro tipko na levi, se vrti levo kolo, tako da se Segway obrača na desno. Če pritisne rdečo tipko na desni, se vrti desno kolo in Segway se obrača na levo.

Slika kaže, kaj se zgodi, če pritisnemo najprej levo tipko (L), nato držimo obe, in nato nekaj časa le desno (R).



Segway se najprej obrne na desno, nato gre naravnost in na koncu obrne levo.

Zdaj pa recimo, da je Segway obrnjen proti severu. Upravljamo ga, kot kaže spodnja slika. Kam je obrnjen na koncu?



Rešitev

Proti jugu. Kadar sta pritisnjeni obe tipki ali pa nobena, se Segway ne obrača. Samo leva je pritisnjena dvakrat, samo rdeča pa štirikrat. Segway se bo torej dvakrat obrnil na desno in štirikrat na levo. Obrnjen bo natančno v nasprotno smer kot na začetku, torej proti jugu.

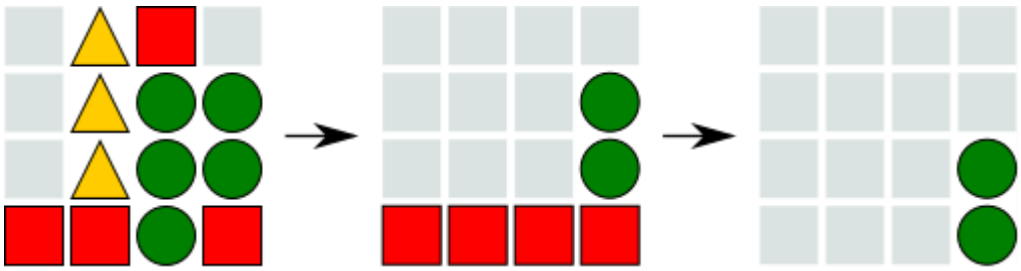
Računalniško ozadje

Programiranje robotov – ali kakih drugih motorjev – je ena bolj zabavnih reči, ki se jih lahko lotiš.

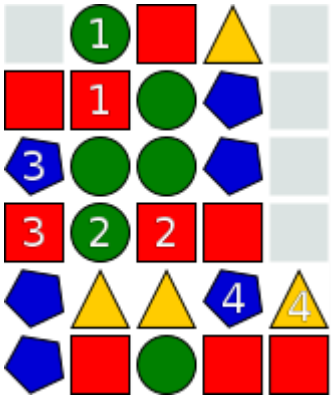
Tri v vrsto

Tri v vrsto je priljubljena računalniška igrica, kjer igralci zamenjujejo like. Če dobimo tri ali več enakih likov skupaj v isti vrstici ali stolpcu, ti liki izginejo, vsi ostali liki pa ustrezno padejo na njihova mesta. Potem se postopek izginjanja likov ponovi, dokler imamo po tri ali več likov skupaj v neki vrstici ali stolpcu. Cilj igre je, da vsi liki izginejo.

Primer izginjanja likov prikazuje naslednja slika:



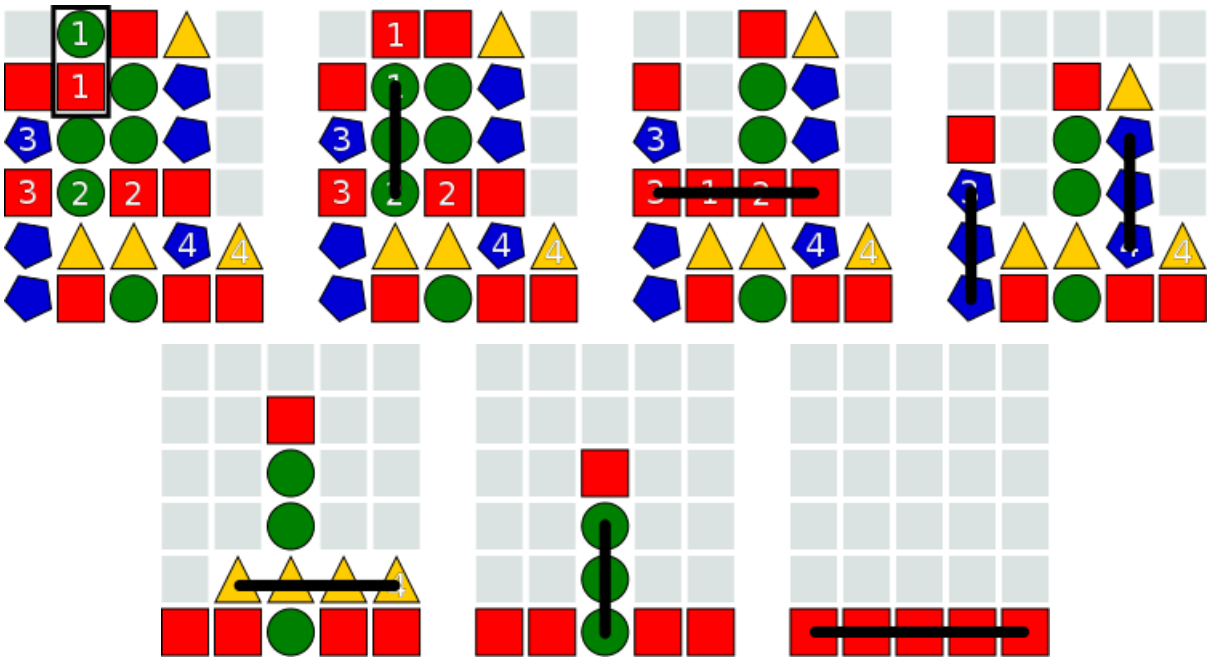
Katera lika naj igralec zamenja, do bodo izginili vsi liki v igri na spodnji sliki?



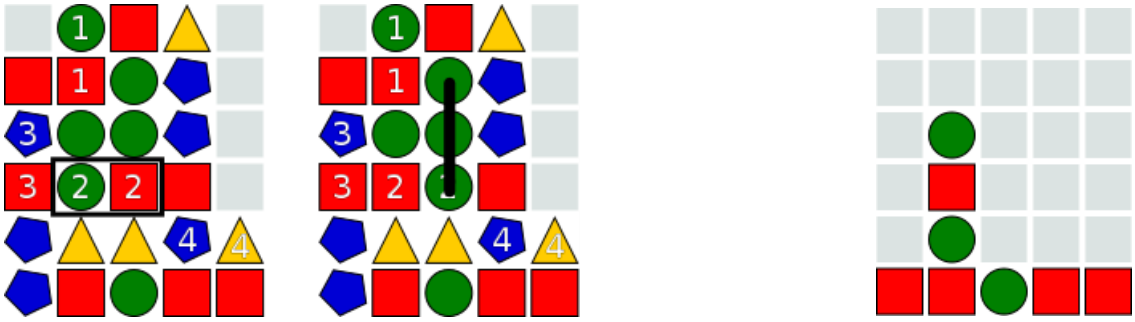
Rešitev

Zamenjati mora lika z oznako 1.

Igro lahko opišemo v naslednjih 6 korakih (v vsakem koraku s črto označimo like, ki izginejo v tem koraku):

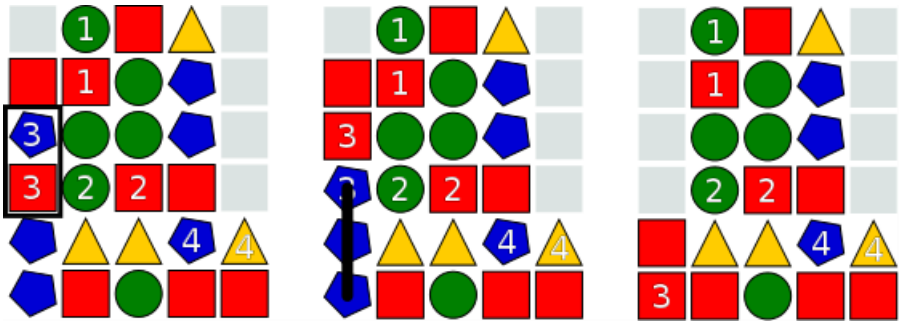


Zamenjava likov z oznako 2 bi spremenil število zelenih krogov v drugem stolpcu z leve na 2 in tako zelena kroga ne bi nikoli uspela izginiti (glej srednjo sliko). Slika na desni prikazuje končno stanje igre.

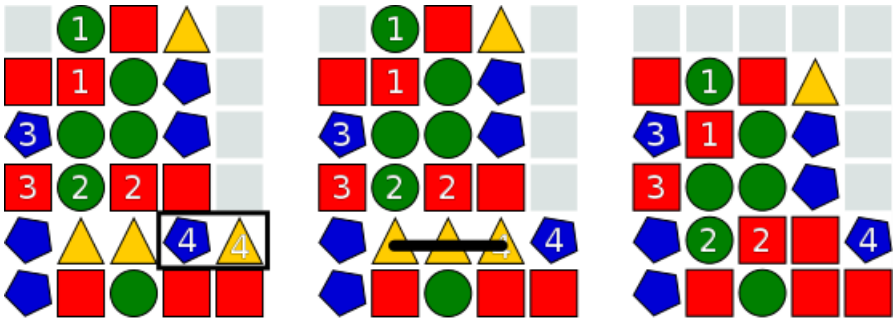


Zamenjava likov 3 in 4 povzročita konec igre že po prvem koraku, saj se po izginotju modrih petkotnikov (oziroma rumenih trikotnikov) ne ustvari nobena vrsta s tremi liki.

Potek igre pri zamenjava likov z oznako 3:



Potek igre pri zamenjava likov z oznako 4:



Računalniško ozadje

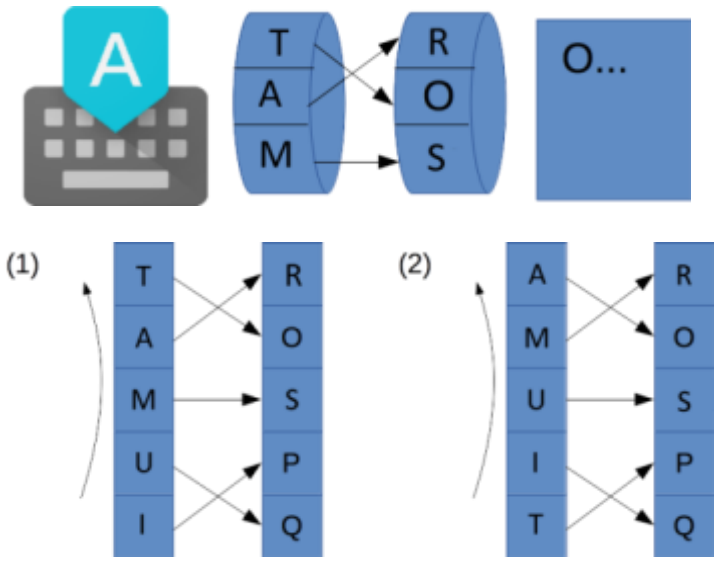
Programerji morajo pri pisanju programov razumeti, kaj bo njihov program naredil, ne da bi ga dejansko pognali. Tudi pri tej uganki imamo podobno situacijo: poiskati smo morali zmagovalno kombinacijo, ne da bi dejansko izvedli igro do konca.

Prazgodovinska šifra

srednja šola

Že v prazgodovini je rivalstvo med različnimi plemeni, predvsem za hrano, privedlo do izuma različnih pripomočkov, s katerimi si je posamezno pleme pomagalo v boju za preživetje. Tako je nastala tudi prazgodovinska polavtomatska kodirna naprava, s katero so si praljudje pošiljali skrivna sporočila.

Kodirna naprava deluje tako, kot prikazuje naslednja slika.



Pri vsaki vpisani črki, ki jo želimo zakodirati (npr. črko T), levo kolo določa ustrezno kodo na desnem kolesu (npr. O v prvem koraku pod (1)). Ko črko zakodiramo, se levo kolo zavrti za eno mesto v smeri puščice ter pristane v položaju, kot ga prikazuje (2). Desno kolo se nikoli ne premakne, pa tudi povezave med obema kolesoma (na sliki so prikazane s puščicami) se ne spremenijo.

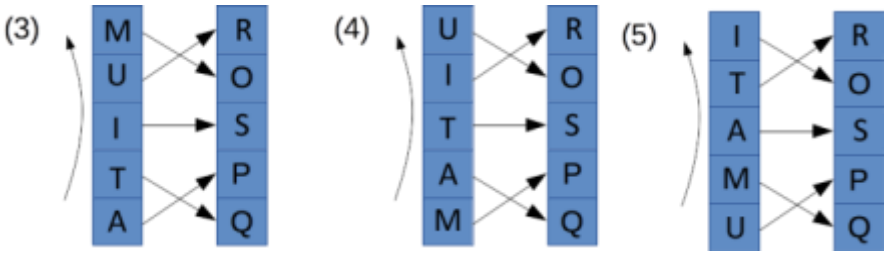
Izvidnik plemena Tam-tam je opazil čredo živali in želi starešini plemena poslati sporočilo MAMUTI, vendar mora paziti, da sporočila ne bi prestregli v rivalskem plemenu Tu-tu. Zato bo sporočilo zakodiral s prazgodovinsko polavtomatsko kodirno napravo.

Kakšno zakodirano sporočilo mora poslati izvidnik, če kodiranje začne v položaju (1)?

A. SOSAEB
B. SOSQOP
C. SOOORP
D. SOOSPQ

Rešitev

Pravilen odgovor je C.
V položaju (1) se črka M zakodira v črko S in v položaju (2) se črka A zakodira v črko O, kot prikazuje zgornja slika. Naslednje pretvorbe za 3. do 5. črko prikazujejo naslednje slike:



Za zadnjo, šesto črko uporabimo spet položaj (1).
V položaju (3) se črka M zakodira v črko O, torej odgovora A in B ne moreta biti pravilna.
V položaju (4) se črka U zakodira v črko O, torej je tudi odgovor D nepravilen.

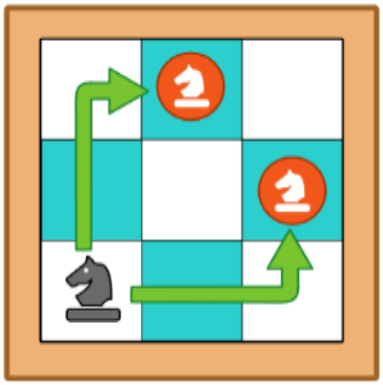
Računalniško ozadje

Opisan stroj za kodiranje je poenostavljena različica kodirnega stroja Enigma, ki so ga za kodiranje sporočil uporabljale nemške oborožene sile med drugo svetovno vojno. Britanska obveščevalna služba se je celo obdobje vojne trudila razbiti šifre Enigme. Delo na razbitju kode je vodilo tudi do razvoja prvih računalnikov. Predstavitev Enigme je tudi najpogostejši uvod na predavanjih o kriptografiji.

Poteze skakača

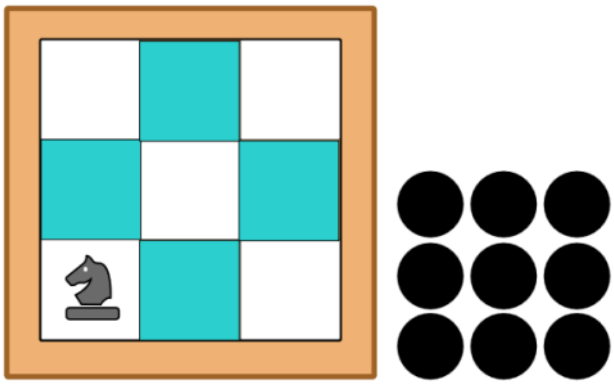
srednja šola

Prijatelja Jan in Žan igrata igro na kvadratni igralni plošči z devetimi polji. Žan ima figuro šahovskega konja, ki se po igralni plošči premika kot pri šahu: konj se lahko premakne za dva koraka v eno smer, se obrne za 90 stopinj in naredi še en korak. Slika prikazuje dva možna premika konja, ki se lahko prestavi na polji, ki sta označeni z oranžnim krogom.



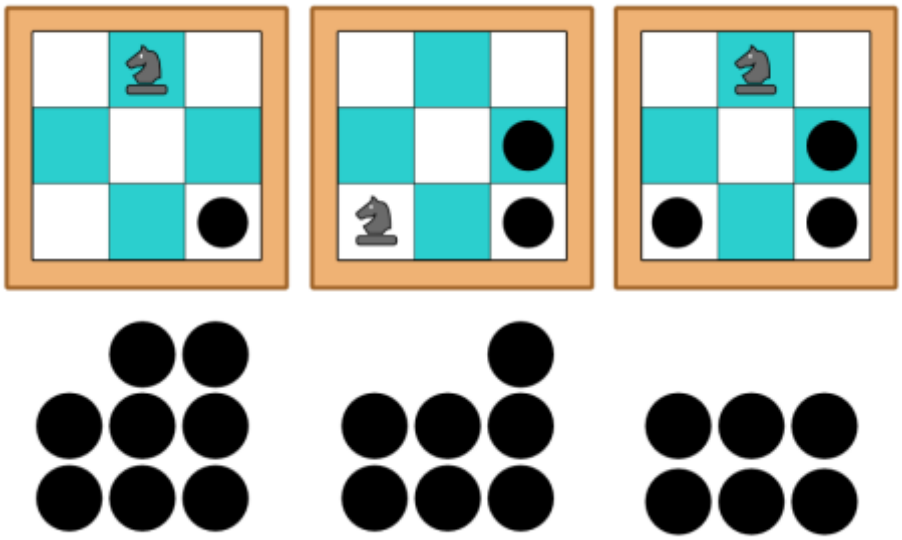
Žanov nasprotnik Jan pa pri igri uporablja figure dame, s katerimi zapre polja na igralni plošči, na katera bi lahko skočil konj pri posameznem premiku.

Začetni položaj igre je prikazan na spodnji sliki. Prvi se mora premakniti Žan, poteze pa prijatelja vlečeta izmenično. Jan poskuša čimprej onemogočiti Žanov premik, saj to vodi v zmago pri igri. Kakšno je največje število premikov, ki jih lahko naredi Žan, preden Jan onemogoči vse možne premike konja?



Rešitev

Tri.
V tej igri se konj nikoli ne more premakniti na polje v sredini. Z vseh ostalih polj pa se konj lahko premakne na dve različni polji. Ko sta obe polji zaprti, se igra konča. Torej je Janova strategija igranja lahko naslednja: v prvem koraku bo prisilil Žana, da se premakne nazaj na začetno pozicijo. V drugem koraku bo prisilil Žana, da ponovi svoj prvi premik. V tretjem koraku pa bo lahko v celoti onemogočil vse Žanove premike.



Računalniško ozadje

Pri igranju igre je pomembno, da znamo vnaprej predvideti možne poteze. To lahko dosežemo tako, da ustvarimo drevo igre, ki predstavlja vse možne premike in odgovore nanje. Da lažje predvidimo naslednje korake igre, moramo znati prešteti vse možnosti v vseh različnih situacijah igre ter skrčiti odvečne informacije v takem drevesu.

Rekurzivna umetnost

srednja šola

Rekurzijo dobimo, kadar se algoritem znotraj algoritma sklicuje na samega sebe. Primer rekurzije iz realnega sveta bi bil odsev v ogledalu, kjer ogledalo držite nasproti drugemu ogledalu, tako da se slika rekurzivno zrcali.

Primer rekurzivnega algoritma je tudi naslednji.

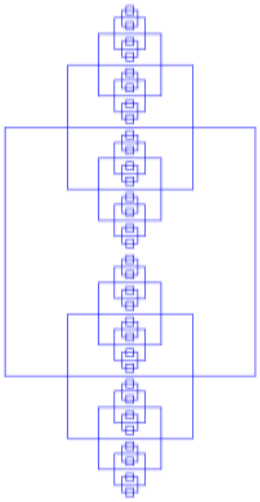
Ukaz NarišiKvadrat(x, y, s) ukaže računalniku, naj izvrši naslednje korake:

- Izriši kvadrat s stranico s in središčem v točki (x, y).
- Če je stranica kvadrata večja od 2 pikslov, potem
  - NarišiKvadrat(x+s/2, y, s/2) [nariši manjši kvadrat na desni] in
  - NarišiKvadrat(x-s/2, y, s/2) [nariši manjši kvadrat na levi]

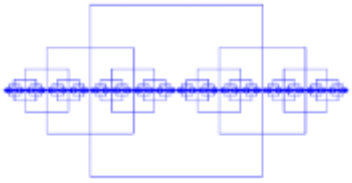
Katerega od spodnjih vzorcev bi lahko narisali z ukazom NarišiKvadrat?

A.

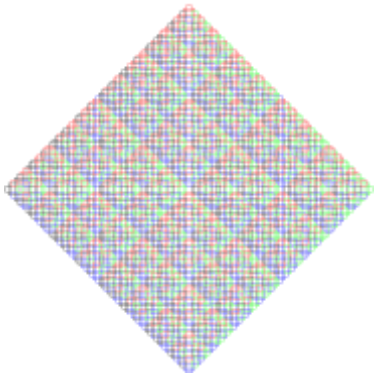
C.



B.



D.



Rešitev

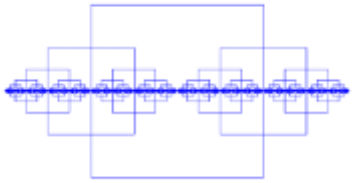
Odgovor B.
Poglejmo si razlago za vsako sliko.

A.

Nepravilen.

Kvadrati so sicer izrisani rekurzivno, a manjkajo kvadrati levo oz. desno od vsakega kvadrata.

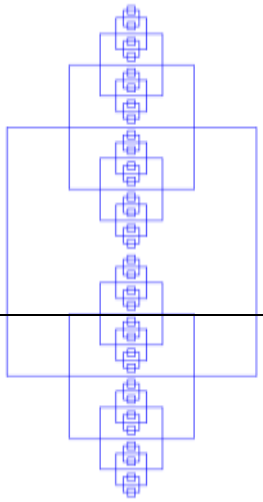
B.



Pravilen.

Vzorec enega velikega kvadrata z dvema manjšima kvadratoma na levi in na desni se ponovi v vsakem kvadratu.

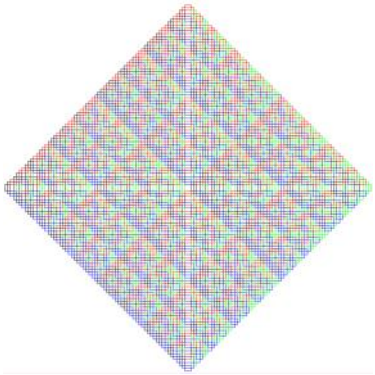
C.



Nepravilen.

Vzorec velikega kvadrata in dveh manjših kvadratov se sicer ponovi v vsakem kvadratu, a so manjši kvadrati nad in pod središčem kvadrata, namesto levo in desno.

D.



Nepravilen.

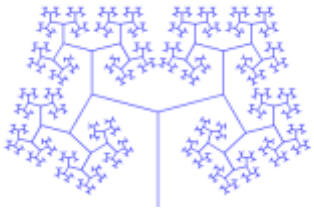
Vzorec velikega kvadrata in dveh manjših kvadratov se ponovi levo in desno, a tudi zgoraj in spodaj.

Računalniško ozadje

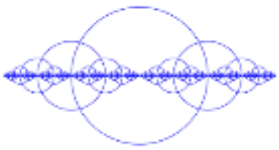
Večina programskih jezikov podpira rekurzijo. To omogoča, da procedure kličejo same sebe za reševanje manjših podproblemov. Tak pristop je zelo uporaben, kadar delamo s kompleksnimi podatkovnimi strukturami, kot je na primer ugnezden seznam.

Rekurzivni fraktali so oblika računalniškega razmišljanja, ki lahko ustvari zanimive kompleksne vzorce z le nekaj vrsticami kode. Slike, ki so narejene s pomočjo algoritma, imenujemo algoritmična umetnost. Nekaj primerov algoritmičnih umetnin, ustvarjenih s pomočjo rekurzivnih fraktalov, prikazujejo naslednje slike.

Kochova krivulja



Fraktalna drevesa



Rekurzivni krogi

V računalniški grafiki se algoritmično umetnost ali fraktalno geometrijo uporablja za ustvarjanje digitalnih slik, ki posnemajo naravo. Za razliko od klasične geometrije ustvari fraktalna geometrija nepravilne oblike, ki so podobne tistim v naravi, kot so na primer snežinke, drevesa in listi. Na tak način lahko ustvarimo dinamične in realistične podobe v računalniških simulacijah.

Kopičenje napak

Anja obožuje staro računalništvo strojno opremo. Zadnjič je kupila starejši računalnik, ki pri izračunih omogoča uporabo le ene decimalke, vse ostale decimalke pa enostavno odreže.

Tako se na primer rezultat izračuna 7/5 na Anjinem računalniku shrani kot 1,4 (kar je povsem pravilno), pri izračunu 7/4 pa se rezultat shrani kot 1,7 (7/4 = 1,75, vendar se druga decimalka, to je 5, odreže). Tako dobimo pri tem izračunu na Anjinem računalniku napako 0,05. Napaka je razlika med pravilno vrednostjo in v računalniku shranjenim rezultatom.

Tako rezanje decimalk se izvede pri vsakem izračunu. Če na primer želimo na Anjinem računalniku izračunati, koliko je (7/4)/2, se najprej izračuna 7/4 in dobimo rezultat 1,7, nato pa še 1,7/2 in dobimo rezultat 0,8. Tako je skupna napaka še večja in znaša 0,075.

Kakšna bo napaka, če na Anjinem računalniku izračunamo rezultat ((10/3)*(10/3))*9?

Rešitev

Napaka je 2,8.
Rezultat 10/3 se shrani kot 3,3. Potem je rezultat izračuna (10/3)*(10/3) = 3,3*3,3 = 10,89 shranjen kot 10,8. Če ta rezultat pomnožimo še z 9, dobimo 10,8*9 = 97,2, tako da se shrani rezultat 97,2. Če bi izračun delali brez napak na decimalnih mestih, bi dobili rezultat ((10/3)*(10/3))*9 = (100/9)*9 = 100. Napaka je torej 100-97,2 = 2,8.

Računalniško ozadje

Podobno le na določeno število decimalk računajo vsi računalniki. Računalniki imajo namreč omejeno količino pomnilnika in lahko shranijo le omejeno količino podatkov. Tudi srce računalnika, centralna procesna enota (CPE), ima le omejeno število bitov za shranjevanje števil. Tako pri shranjevanju ulomkov ne moremo shraniti natančnega decimalnega števila, ampak ga moramo na nekem decimalnem mestu zaokrožiti. Tako se pojavijo zaokrožitvene napake, ki se pri računanju kopičijo, posledično pa so daljši izračuni lahko bolj nenatančni.

Računalničarji se trudijo zmanjšati te napake v nekaterih algoritmih ter tako omogočiti izračune, ki so bližje pravi vrednosti.

Hitro potenciranje

Izračunati moraš rezultat 2^{37}, a tipka za eksponent na tvojem kalkulatorju ne deluje. Prijatelj ti poda naslednja dva nasveta, kako priti do rezultata:

- Če je eksponent sodo število, izračunaj 2 na stopnjo potence, ki je enaka polovici eksponenta, ter rezultat pomnoži s samim seboj. Primer: $2^{10} = 2^5 \times 2^5$.
- Če je eksponent liho število, izračunaj 2 na stopnjo potence, ki je za ena manjša od eksponenta (tako da je stopnja soda), in nato rezultat pomnoži z 2. Primer: $2^{11} = 2^{10} \times 2$.

Če uporabiš prijateljeva nasveta, koliko množenj moraš opraviti, da lahko izračunaš rezultat 2^{37}?

Rešitev

Sedem.
Če upoštevamo oba nasveta, lahko rezultat potenciranja izračunamo z uporabo množenj na naslednji način:

$2^{37} = 2^{36} \times 2$ (eksponent je liho število, zato uporabimo drugi nasvet)

Sedaj moramo izračunati 2^{36}, kar je že nekoliko manjši problem:

$2^{36} = 2^{18} \times 2^{18}$ (eksponent je sodo število, zato uporabimo prvi nasvet)

Sedaj moramo za rešitev izračunati rezultat 2^{18}, kar je še nekoliko manjši problem, uporabimo pa spet oba nasveta prijatelja. Naslednji koraki so:

$2^{18} = 2^9 \times 2^9$

$2^9 = 2^8 \times 2$

$2^8 = 2^4 \times 2^4$

$2^4 = 2^2 \times 2^2$

$2^2 = 2 \times 2$

Zadnji korak pa je enostavno izračunati, saj ne zahteva potenciranja. Končen rezultat dobimo tako, da gremo od spodaj, od zadnjih izračunov navzgor ter sproti s pomočjo množenja računamo delne rezultate potenciranj vedno višjih eksponentov. Vidimo, da imamo skupaj sedem množenj.

Računalniško ozadje

Prijateljeva nasveta vodita v izjemno učinkovit algoritem glede na število potrebnih množenj. Uporabili smo le sedem množenj; primerjajmo to s številom množenj, ki bi jih uporabili sicer:

$$2^{37} = 2 \times 2$$

Ta razlika je še bolj očitna pri večjih vrednostih eksponenta. Natančneje, če ima neko število E v dvojiški predstavitvi k števk, potem potrebujemo le k množenj, da lahko izračunamo število 2^E (ali katerokoli drugo število na potenco E). Z drugimi besedami, število množenj je enako dvojiškemu logaritmu eksponenta E. V računalništvu temu rečemo, da ima kompleksnost $O(\log_2(E))$.

Oba prijateljeva nasveta izhajata iz naslednje lastnosti potenc: $b^n \times b^m = b^{n+m}$. Če je eksponent naravno število, se postopek vedno zaključi, ker velja:

- v vsakem koraku (ko uporabimo enega od nasvetov), se eksponent zmanjša najmanj za 1,
- ko eksponent doseže 1, s postopkom končamo, saj ne potrebujemo več nobenih izračunov,
- eksponent bo vedno liho ali sodo število, zato lahko vedno uporabimo enega od nasvetov.

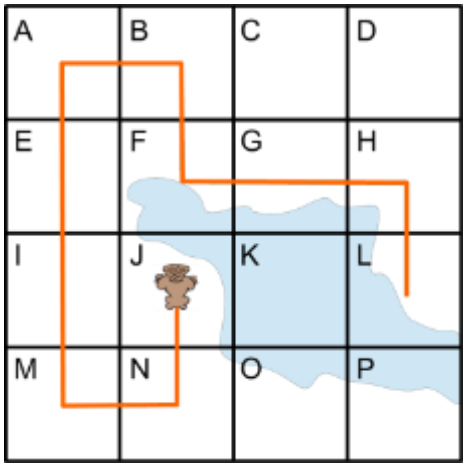
Opisan pristop se imenuje tudi *kvadriraj in pomnoži* (angl. *square-and-multiply*) in je poznan že več kot 2200 let. Postopek je rekurziven, saj se potenca računa tako, da najprej izračunamo potence z manjšimi eksponenti in nato zmnožimo rezultate.

Obstajajo pa tudi hitrejši načini za računanje potenc, a ti navadno zahtevajo vodenje evidence predhodno izračunanih rezultatov ali pa uporabljajo vnaprej izračunana števila.

Načrt opravil

srednja šola

Bobri zelo radi natančno opravljajo svoja opravila. V tem so bobri tako dobri, da si zabeležijo navodila, da lahko opravilo ponovijo. Vsako navodilo temelji na zemljevidu, ki je razdeljen na kvadrate.



Bober Luka je za zapis navodila uporabil štiri simbole:

x	skoči en kvadrat naprej
+	poberi poleno
-	odloži poleno
o	obrni se na levo (90 stopinj v nasprotni smeri urinega kazalca)

Luka želi pobrati nekaj polen, zato je vzel eno od shranjenih navodil:

xoooox++++oooxxxoooox--oooxoxx++++oooox

To navodilo vodi Luko od njegovega doma (v kvadratu J) na drugo stran jezera (kvadrat L) in mu omogoči pobrati 6 polen (od doma starta z 0 poleni).

Luka je ravno prišel na drugo stran jezera v kvadrat L, ko se je spomnil, da je pravzaprav na dieti in ne potrebuje toliko polen. Zato potrebuje novo navodilo, ki ga bo vodilo od kvadrata L nazaj do doma v kvadratu J, tako da bo sledil svoji prejšnji poti. Po poti pa bo v vsako skrivališče odložil natančno toliko polen, kot jih je skrivališče imelo, preden se je Luka odpravil na to pot.

Napišite ustrezno zaporedje navodil na Luko (sestavljeno je lahko le iz znakov **x**, **+**, **-** in **o**).

Rešitev

oox----oxxoooox++oxoxxx----oxox

Če želimo obrniti prvotno navodilo, moramo zamenjati vse ukaze »odloži poleno« (**-**) in »poberi poleno« (**+**). Poleg tega moramo ukaz za »obrni se na levo« (**o**) zamenjati z ukazom »obrni se na desno« (**ooo**) in obratno. Pravzaprav lahko ustvarimo neskončno veliko pravilnih navodil, če vanje vstavimo ukaze, ki nimajo učinka, kot so »obrni se za 360 stopinj« ali pa »poberi in odloži poleno«. Podana rešitev je najkrajša.

Računalniško ozadje

Lukova navodila so kot računalniški program. Na program lahko gledamo tudi kot na matematično funkcijo, ki preslika eno stanje sveta v drugo. Včasih imamo take preslikave, ki različnih elementov nikoli ne preslikajo v isti element, zato lahko tako preslikavo tudi obrnemo in poiščemo obratno funkcijo, ki naredi nasprotno preslikavo. V računalništvu obravnavamo program kot stroj, ki spreminja svet iz začetnega v končno stanje, pri tem pa je vsak korak izračuna sam zase majhen program, ki spremeni svoje začetno stanje (stanje pred korakom) v stanje po koraku. Obrnjen program je torej sestavljen iz posameznih obrnjenih korakov. Podobno smo imeli tudi v naši nalogi: »odloži poleno« in »poberi poleno« sta medsebojno obrnjena ukaza, obrnjen ukaz za »obrni se levo« pa so trije zaporedni »obrni se levo«, ukaz »skoči en kvadrat naprej« pa je sam sebi obrnjen ukaz (spremeni le stanja pred in po).

Aljošin računalnik obdeluje podatke na poseben način, saj uporablja le naslednje štiri operacije:

- (max x_1 x_2 … x_n) poišče največjo vrednost izmed x_1, x_2, … x_n
- (min x_1 x_2 … x_n) poišče najmanjšo vrednost izmed x_1, x_2, … x_n
- (+ x_1 x_2 … x_n) izračuna $x_1 + x_2 + \ldots + x_n$
- (· x_1 x_2 … x_n) izračuna $x_1 \cdot x_2 \cdot \ldots \cdot x_n$

Vse te operacije lahko tudi gnezdimo. Tako nam na primer izračun (+ (· 2 3) (+ 1 2)) da rezultat 9.

Kakšen rezultat nam da naslednji izračun?

```
(+ (max (min 3 9 2) (· (max 0 4) (min 0 4))) (min (max 3 6) (max 5 7 2)))
```

Rešitev

Rezultat je 8.

Računamo ga od znotraj navzven; najprej izračunamo rezultate najbolj notranjih operacij. V vsakem koraku izračunamo rezultate podčrtanih delov, ki jih nato uporabimo pri operacijah v naslednjem koraku, ki je zapisan v novi vrstici:

```
(+ (max (min 3 9 2) (· (max 0 4) (min 0 4))) (min (max 3 6) (max 5 7 2)))
                  ⇒ (+ (max 2 (· 4 0)) (min 6 7))
                        ⇒ (+ (max 2 0) 6)
                              ⇒ (+ 2 6)
                                    8
```

Računalniško ozadje

Naloga obravnava funkcijsko programiranje in gnezdenje izrazov. Idejo gnezdenja uporabljamo pri preglednicah: podatke pridobivamo z neko funkcijo (na primer vsoto števil), ki ji sledi druga funkcija, (na primer največja vrednost vseh izračunanih vsot števil), tej sledi tretja...

Na tem vzorcu računanja temeljijo nekateri programski jeziki, med katerimi so najbolj znani Lisp, Scheme in Racket. Nekaj podobnih idej najdemo tudi v popularnejših jezikih, kot sta C# in Python.

Takšnemu zapisu izrazov, v katerem je operator zapisan pred operandi, pravimo prefiksna (ali poljska) notacija.

Štirje prijatelji si izmenjujejo sporočila. Ana in Beti uporabljata le N-Mail. Ciril in Dejan včasih uporabita N-Mail in včasih P-Mail. Pri posredovanju sporočil drugi osebi N-Mail vedno doda nov del sporočila nad izvorno besedilo, P-Mail pa nov del sporočila doda pod izvorno besedilo.

Predpostavimo, da je Ana poslala prvo sporočilo Cirilu. Ciril je uporabil P-Mail in ga preposlal Beti. Na koncu je Beti preposlala sporočilo Dejanu. Sporočilo, ki ga je prejel Dejan, zgleda tako:

Naslednja slika prikazuje drugo izmenjavo sporočil. Ker uporabljajo različna programa, ne vemo, kdo je poslal prvo sporočilo.

Kdo zagotovo **ni** poslal prvega sporočila?

Rešitev

Ana. Če bi ga poslala Ana, Betijino sporočilo ne bi smelo biti pod njenim. Vsi ostali so lahko poslali prvo sporočilo.

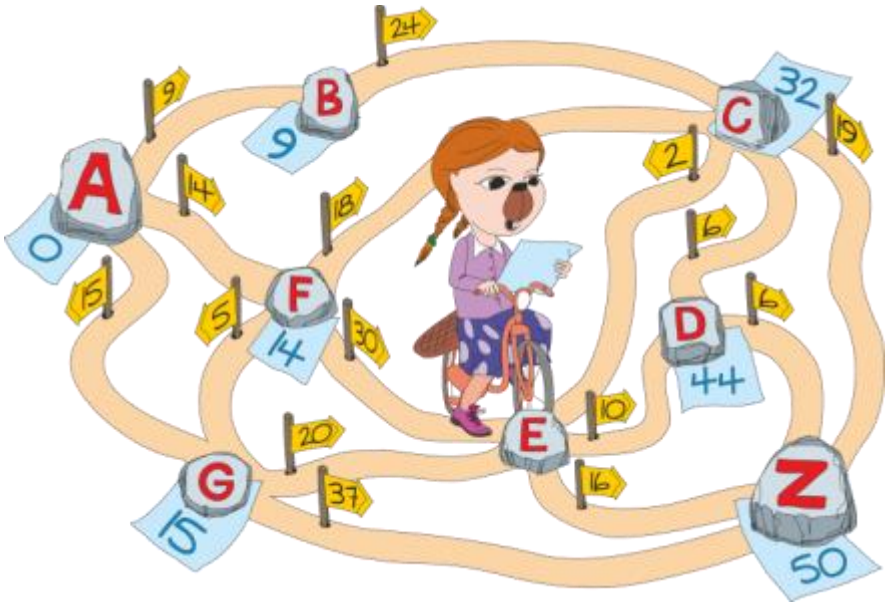
Ozadje naloge

Pri odgovarjanju na e-poštna sporočila in objave na forumih se uporabljajo trije načini objavljanja izvornega sporočila: objavljanje po vrsticah, objavljanje spodaj in objavljanje zgoraj. Če uporabimo dva programa za e-pošto, ki uporabljata različna načina objavljanja izvornega sporočila, lahko pride do situacije, ki je predstavljena v nalogi.

Raziskovanje poti

srednja šola

Cleveria s kolesom raziskuje enosmerne poti skozi vasi, označene od A do Z. Na vseh poteh so rumene zastavice z oznakami za dolžino poti in dovoljeno smer vožnje.



Cleveria vasi obišče večkrat in po poti v vsaki vasi pusti pod kamnom moder listek s številko. Kaj pomenijo te številke?

A. Dolžino najkrajše poti od A do določene vasi, ki gre skozi najmanj drugih vasi.
B. Dolžino najkrajše poti od A do določene vasi.
C. Dolžino poti od A do določene vasi, na kateri, kadar je to mogoče, na križiščih zavijemo levo.
D. Dolžina poti od A do določene vasi, na kateri, kadar je to mogoče, na križiščih zavijemo desno.

Rešitev

Pravilni odgovor je b. Cleveria išče najkrajše poti od A do ostalih vasi.

- Odgovor a je napačen, ker bi sicer imeli D = 45 in Z = 52.
- Odgovor c je napačen, ker bi sicer imeli C = 33, D = 45 in Z = 52.
- Odgovor d je napačen, ker bi sicer imeli C = 51, D = 45 in Z = 52.

Ozadje naloge

Na takšnem označevanju temelji Dijkstrin algoritem, ki ga običajno uporabljamo za iskanje poti.

Vezalke

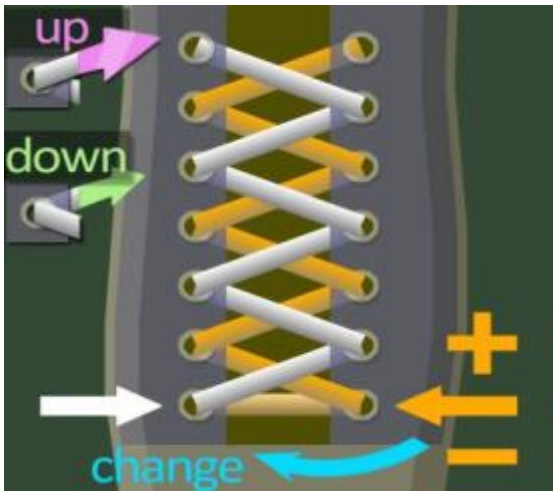
srednja šola

Maja obožuje nenavadno zavezane vezalke na čevljih. Želi si imeti robota, ki bo zavezoval vezalke namesto nje, zato si je izmislila nov programski jezik, s katerim lahko robotu pove, kako naj ji zaveže vezalke.

V programu je predpostavila, da je del vezalke, ki se začne na desni strani čevlja, vedno oranžen in del vezalke, ki se začne na levi strani, vedno bel. Jezik sestavljajo naslednji ukazi:

`{…}`	Vse med oklepajema bo ponovljeno, kolikorkrat je to mogoče
`n{…}`	Vse med tema oklepajema bo ponovljeno n-krat, pri čemer je n neko podano število
`oranžna:`	Naslednji ukazi veljajo samo za oranžno vezalko
`bela:`	Naslednji ukazi veljajo le za belo vezalko
`gor`	Obrni vezalko navzgor in jo vodi skozi luknjo, ki jo kaže oranžna ali bela puščica
`dol`	Obrni vezalko navzdol in jo vodi skozi luknjo, ki jo kaže oranžna ali bela puščica
`+`	Premakni oranžno ali belo puščico na naslednjo luknjo zgoraj
`-`	Premakni oranžno ali belo puščico na prejšnjo luknjo spodaj
`zamenjaj`	Oranžna in bela puščica naj glede na trenutni položaj zamenjata stran čevlja

Primer:



Spodnja koda opisuje, kako je zavezan čevelj na levi sliki.

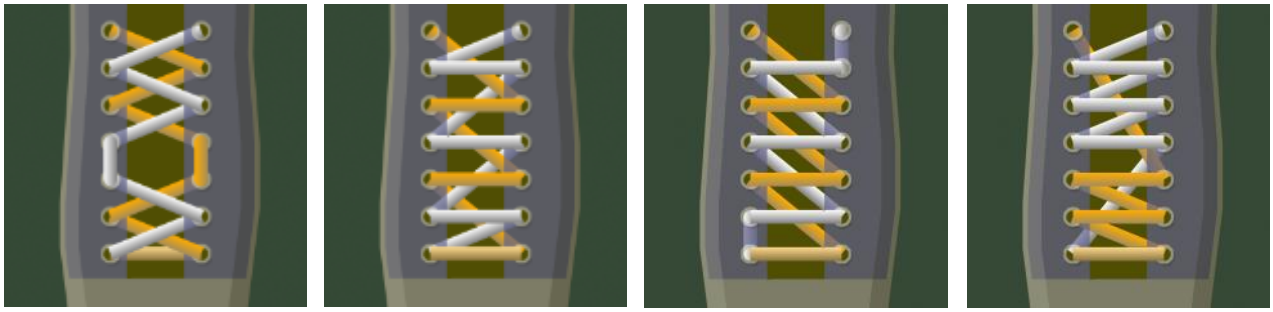
```
oranžna: gor
bela: gor
{

 oranžna: + zamenjaj gor
 bela: + zamenjaj gor

}
```

Kako izgledajo zavezane vezalke, če robot izvede naslednjo kodo?

```
oranžna: gor
bela: gor
2{
      oranžna: + zamenjaj gor
      bela: + zamenjaj gor
}
oranžna: + dol
bela: + dol
{
      oranžna: + zamenjaj gor
      bela: + zamenjaj  gor
}
```

Namig: osredotoči se na eno izmed vezalk.

A B C D



Rešitev

Pravilen odgovor je A.

Ozadje naloge

Ta preprost programski jezik je v nekaterih pogledih podoben običajnemu programskemu jeziku. V njem najdemo tako spremenljivke kot tudi zanke.

Vrnitev na začetek

srednja šola

Robotska čebela ima na hrbtu štiri gumbe s puščicami. Čebela se premika po tleh, tlakovanih s kvadratnimi ploščicami. Pri tem upošteva program, ki se ga napiše s kombinacijo pritiskov na gumbe:

Premakni se naprej na ploščico spredaj

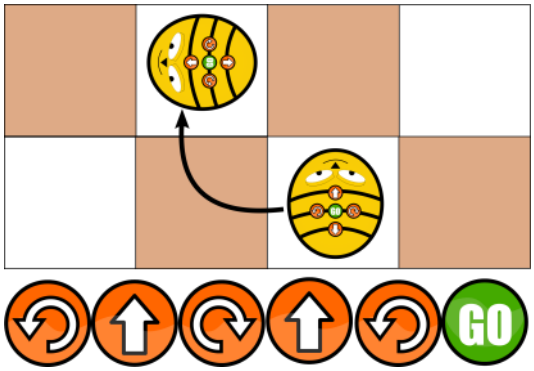
Ostani na tej ploščici in se obrni za 90° levo

Ostani na tej ploščici in se obrni za 90° desno

Vzvratno se premakni na ploščico zadaj

Zaporedje ukazov se začne izvajati s pritiskom na gumb GO.

Primer zaporedja pritisnjenih gumbov in njegova izvedba sta prikazana na sliki.



Čebela si zapomni zadnje vneseno zaporedje, zato ponovni pritisk na gumb GO povzroči ponovno izvedbo vnesenih korakov. Opazite lahko, da se lahko čebela po večkratnih zaporednih pritiskih na gumb GO bodisi vrne na začetni položaj bodisi se tja ne vrne nikoli.

Recimo, da čebela izvaja neko zaporedje, po katerem se - po določenem številu pritiskov na GO vrne na začetno polje in je obrnjena enako kot v začetku. Največ koliko pritiskov bo potrebnih za to?

Rešitev

4.

Obstajajo 3 različne možnosti, da se čebela lahko vrne domov:

1. Ena izvedba zaporedja ukazov ne spremeni orientacije čebele. V tem primeru se čebela lahko vrne v prvotno pozicijo le, če se vrne po prvi izvedbi zaporedja ukazov.
2. Ena izvedba zaporedja ukazov spremeni smer za 180°. Naslednja izvedba zaporedja ukazov bo čebelo obrnila nazaj v prvotno smer. V prvotno pozicijo se mora v tem primeru vrniti po 2 izvedbah.
3. Ena izvedba zaporedja ukazov obrne čebelo za 90° levo ali desno. Po treh dodatnih pritiskih na gumb GO, se mora čebela vrniti v začetno pozicijo, sicer se ne vrne nikoli.

Ozadje naloge

Programi so zaporedja ukazov, ki se izvajajo zaporedno kot pri programiranju robotske čebele. Če želimo zaporedje istih ukazov izvesti večkrat zapored, te ukaze zapišemo v zanko, ki se izvaja, dokler ne doseže izhodnega pogoja.

Čitalca slik

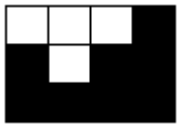
srednja šola (državno)

Imamo dva čitalca slik, ki sliko zakodirata tako, da piksle na sliki pretvorita v posebno kodo. Koda najprej navede število vseh zaporednih pikslov iste barve (bele ali črne), temu sledi število vseh zaporednih pikslov druge barve in tako naprej do konca slike, začenši v zgornjem levem kotu in premikajoč se od leve proti desni, vrstico za vrstico.

Čitalca uporabljata različne načine obravnave konca vrstice:

- Čitalec A obdeluje vse piksle po vrsticah in v vsaki novi vrstici začne s kodiranjem na novo (ponastavi števec zaporednih pikslov enake barve).
- Čitalec B obdeluje vse piksle po vrsticah, a v novi vrstici nadaljuje s kodiranjem iz predhodne vrstice (števca zaporednih pikslov enake barve ne ponastavi, ampak šteje dalje).

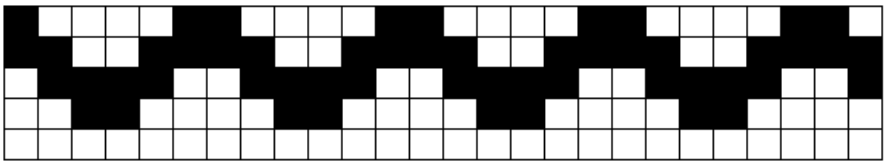
Tako bi na primer za sliko na desni oba čitalca sestavila različno kodo:



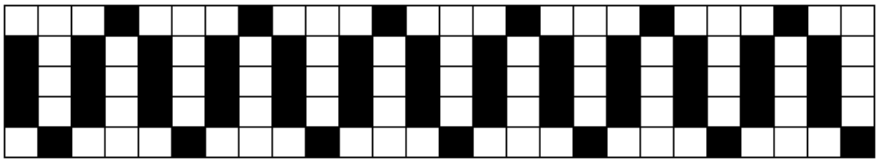
- čitalec A: 3, 1, 1, 1, 2, 4 (3 beli, 1 črn, 1 črn, 1 bel, 2 črna, 6 črnih)
- čitalec B: 3, 2, 1, 6 (3 beli, 2 črna, 1 bel, 6 črnih)

Pri kateri od spodnjih slik dobimo enako kodo, ne glede na to, katerega od čitalcev uporabimo?

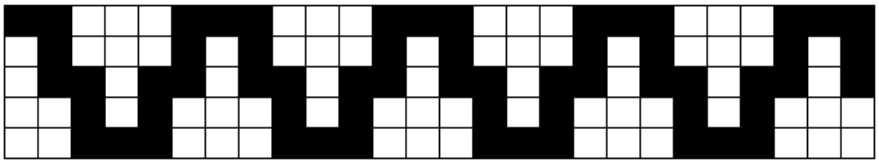
A.



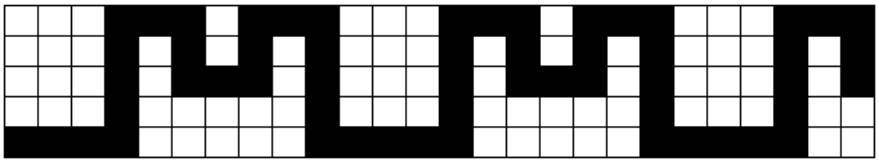
B.



C.



D.

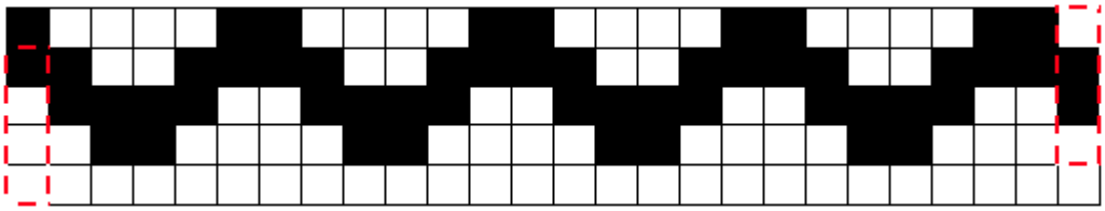


Rešitev

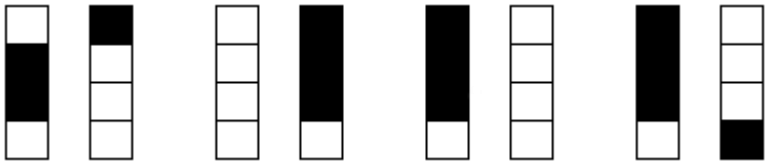
Pravilni odgovor je D.

Razlika pri zapisu kode pri obeh čitalcih je v tem, ali zadnji piksel v vrstici združimo s prvim pikslom v naslednji vrstici ali ne. Čitalec A ju ne združi, čitalec B pa ju združi le v primeru, če sta iste barve. Če torej ta dva piksla nista iste barve, je koda čitalca A in čitalca B enaka.

Poiskati moramo torej sliko, pri kateri zadnji piksel v vrstici in prvi piksel v naslednji vrstici nista iste barve. To mora veljati za vse vrstice.



Primerjati moramo le 4 zadnje piksle v vrsticah od 1 do 4 in ustrezne 4 prve piksle v vrsticah od 2 do 5 ter preveriti, ali so različne barve. Spodaj so prikazani omenjeni pari pikslov za vse štiri slike v podanih odgovorih.



Od vseh štirih slik lahko le na sliki D najdemo vse pare pikslov različnih barv.

Računalniško ozadje

Pri skeniranju slike se barva in svetlost vsakega delčka slike (piksla) izmerita s pomočjo senzorja ter zapišeta kot številčna vrednost. Temu procesu rečemo tudi digitalizacija slike.

Piksel (angleško *pixel*) je računalniški izraz, ki je nastal iz angleških besed *picture element*, kar pomeni slikovni element ali element slike. To je najmanjši element digitalne slike. Vsak piksel predstavlja delček originalne slike in več kot imamo teh delčkov, bolj natančno predstavitev originalne slike lahko dobimo.

Čitalec A v naši nalogi ponastavi kodiranje v vsaki novi vrstici, medtem ko čitalec B piksle bere in kodira, kot da so zapisani v eni sami nepretrgani vrstici. Vsak od opisanih pristopov ima svoje prednosti, ko ga uporabimo v praksi. Tako na primer čitalec B pri velikih slikah verjetno potrebuje manj številk za kodiranje slike, vendar mora zakodirati tudi dimenzije slike (vsaj širino slike). Zato tak pristop morda ni praktičen pri manjših slikah. Taki kompromisi so zelo pomembne odločitve, ki jih moramo nenehno sprejemati v računalništvu.

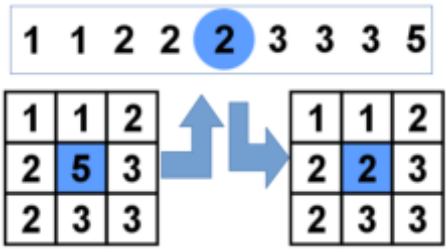
Filter

srednja šola (državno)

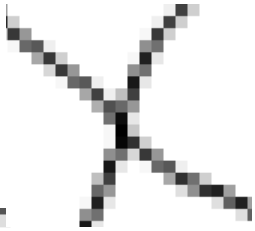
Sivinsko sliko imamo shranjeno kot tabelo števil, kjer vsak element tabele ustreza enemu pikslu slike. Če ima element tabele vrednost 1, to pomeni črn piksel; če ima vrednost 5, pomeni bel piksel. Ostala števila med 1 in 5 pa določajo različne odtenke sivine.

Na sliki uporabimo filter z mediano, ki sliko pretvori na naslednji način: za vsak piksel na sliki naraščajoče uredi vrednost tega piksla in vrednosti vseh njegovih sosednjih pikslov, nato pa uporabi vrednost na sredini (peto po vrsti) kot novo vrednost tega piksla. Filter pretvori vse piksle slike naenkrat.

Če pogledamo primer na spodnji sliki, se vrednost srednjega piksla spremeni s 5 na 2.

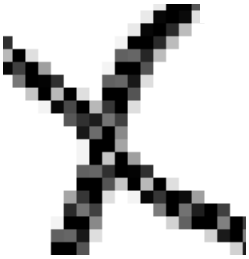


Kako bo izgledala desna slika, če na njej uporabimo opisan filter z mediano?



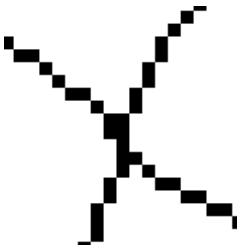
A.

C.



B.

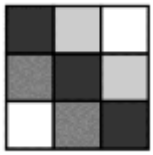
D.



Rešitev

Pravilni odgovor je A.

Poglejmo tipični element naše slike.



Če na tem elementu uredimo vse piksle po njihovi intenziteti, dobimo dva bela piksla na desni, tri črne piksle na levi ter štiri sive piksle na sredini. Tako filter spremeni sredinski (črn) piksel v sivega. Poleg tega bo črn piksel, ki je obkrožen s črnimi piksli, ostal črn; bel piksel, obkrožen z belimi piksli, pa bo ostal bel.

Računalniško ozadje

Procesiranje slik je pomembna funkcija del grafičnih urejevalnikov, pa tudi sistemov računalniškega vida. Grafični urejevalniki znajo popraviti slike, na primer zmanjšati šum na sliki, a uporabniki navadno ne poznajo algoritmov, na katerih temeljijo te izboljšave slike. Filter z mediano je eden takih algoritmov, katerega opis je tudi enostavno razumljiv.

Iskanje tatu

srednja šola (državno)

O, ne! Slavni modri diamant so danes ukradli iz muzeja! Tat ga je zamenjal za poceni imitacijo zelene barve.

Razstavo diamantov je danes obiskalo 2000 obiskovalcev. V sobo z diamantom so vstopali posamično. Detektiv Sraka mora odkriti tatu z zasliševanjem nekaterih današnjih obiskovalcev. Na voljo ima seznam vseh 2000 obiskovalcev v vrstnem redu, v katerem so vstopali v sobo. Vsakega obiskovalca bo vprašal isto vprašanje: *Je bil diamant, ki ste ga videli modre ali zelene barve?* Vsi bodo odgovorili po resnici, razen tatu, ki bo trdil, da je bil diamant že zelene barve.

Detektiv Sraka je prebrisan in bo uporabil strategijo, s katero bo izprašal najmanj ljudi in hkrati zagotovo odkril tatu. Kaj lahko obljubi?

A. Lahko zagotovim, da bom našel tatu in izprašal manj kot 20 ljudi.
B. Če izprašam 20 ljudi, to ne bo dovolj (razen, če bom imel srečo), ampak zagotovo lahko opravim nalogo, če izprašam manj kot 200 ljudi.
C. To bo težka naloga: izprašati bom moral vsaj 200 ljudi, ampak možno je, da jih bom moral izprašati 1999.
D. Ničesar ne morem obljubiti. Če ne bom imel sreče, bom morda moral izprašati vse obiskovalce.

Rešitev

Pravilen odgovor je A.

Morda presenetljivo, ampak detektiv Sraka mora izprašati manj kot 20 obiskovalcev. Inšpektor najprej vpraša 1000. obiskovalca po vrsti, nato pa se glede na njegov odgovor odloči, ali bo vprašal 500. ali 1500. V vsakem koraku se seznam osumljencev zmanjša za polovico in vsakič detektiv vpraša obiskovalca, ki je na sredini novega seznama. Če je na prvotnem seznamu 2000 obiskovalcev, je na drugem seznamu 1000 obiskovalcev, na tretjem 500, četrtem 250, petem 125, šestem 63, sedmem 32, osmem 16, devetem 8, desetem 4, enajstem 2. Ko pride do dveh osumljencev, mora vprašanje zastaviti prvemu po vrstnem redu, saj bo s tem vprašanjem izvedel, kdo je tat. Tako bo zaslišal le 11 oseb.

Računalniško ozadje

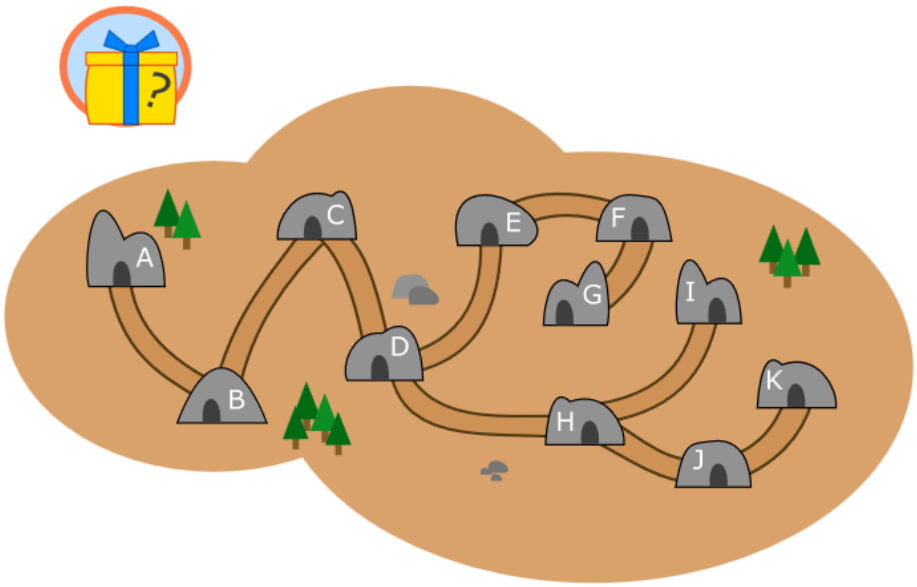
V ozadju naloge se skriva dvojiško (binarno) iskanje. To je optimalni algoritem za iskanje podatka v urejeni tabeli, ki v vsakem koraku področje iskanja v tabeli zmanjša za polovico.

Jame

srednja šola (državno)

Na kraškem predelu Slovenije je veliko jam, ki so jih za lažjo dostopnost vaščani povezali s potmi; med dvema jamama pa vodi vedno le ena sama pot (glej spodnjo sliko).

Aleš in Bor, ki živita v vasi blizu jam, sta si zamislila naslednjo igro. Aleš skrije darilo v eno izmed jam, Bor pa mora ugotoviti, v kateri jami je darilo, če ga želi dobiti. Pri tem si lahko pomaga z zemljevidom na sliki, Aleša pa sprašuje le z vprašanji oblike »Ali je darilo v jami X?«. Aleš mu odgovori z da, če je darilo res v jami X, oziroma mu pove ime sosednje jame od jame X, ki je na poti proti iskanemu darilu. Ko Bor ugotovi, v kateri jami je skrito darilo, je igre konec in Bor lahko odide do jame po zasluženo darilo.



Bor želi čim prej priti do darila, zato želi postaviti čim manj vprašanj. Koliko vprašanj mora postaviti v najslabšem primeru?

Rešitev

V najslabšem primeru mora zastaviti tri vprašanja.

Najprej pokažimo, da zadostujejo tri vprašanja ne glede na to, kje se nahaja darilo. Primer takih vprašanj je naslednji.

Prvo vprašanje: Ali je darilo v jami D?

Možni so štirje različni odgovori, pri vsakem od njih pa lahko najdemo darilo v najmanj dveh nadaljnjih korakih:

- Da: To pomeni, da smo našli jamo z darilom. Skupaj smo postavili 1 vprašanje.
- C: Vprašamo še za jamo B in že vemo odgovor. Skupaj smo postavili 2 vprašanji.
- E: Vprašamo še za jamo F in že vemo odgovor. Skupaj smo postavili 2 vprašanji.
- H: Vprašamo še za jamo H. Če je to prava jama ali pa če je odgovor I, smo ugotovili jamo z darilom. Skupaj smo postavili 2 vprašanji. Če pa je odgovor J, moramo postaviti še eno vprašanje, in sicer za jamo J, da lahko najdemo jamo z darilom. Skupaj smo postavili 3 vprašanja.

Sedaj pa pokažimo še, da v primeru manj kot treh vprašanj najdemo primere, ko po dveh vprašanjih ne moremo biti prepričani, kje se skriva darilo. Jasno je tudi, da Bor ne more kar uganiti z enim samim vprašanjem ali celo brez vprašanj. Recimo, da Bor postavi le dve vprašanji. Če prvo vprašanje preveri jamo D, je to najbolj primerno, saj v primeru, da se darilo skriva v jamah A, B, C, D, E, F ali G, lahko z drugim vprašanjem že ugotovimo pravo jamo. Vendar pa v primeru, če je odgovor na prvo vprašanje H, drugo vprašanje, ki nam še ostane, ni dovolj, da bi z gotovostjo lahko potrdili, da se darilo skriva v jamah H, I, J ali K.

Računalniško ozadje

Gre za vprašanje preiskovanja drevesa. V računalništvu so podatki velikokrat shranjeni v podatkovnih strukturah, ki so podobne drevesom. Zato sta razvoj in uporaba ustreznih algoritmov, s pomočjo katerih pridobimo podatke iz drevesnih struktur, pomemben del računalništva.

V našem primeru so jame vozlišča v drevesu. Še posebej so pomembna vozlišča, ki do katerih vodijo vsaj tri poti: ta vozlišča razdelijo drevo na veje in če vemo, kateri veji moramo slediti, da pridemo do iskanega podatka, lahko preiščemo le to vejo, ne pa tudi ostalega drevesa. Zato sta drevesna struktura in način shranjevanja podatkov v njej nadvse pomembna. S pravilnim organiziranjem podatkov lahko izberemo le eno vejo izmed več možnostmi in če je v drevesu več takih razvejitvenih točk, lahko pri iskanju ustreznih podatkov preiščemo le manjši del celega drevesa. V skrajnem primeru, ko imamo eno samo vejo, moramo preiskati celo vejo (celo pot), dokler ne najdemo iskanega podatka. V drugi skrajnosti, ko ima naše drevo obliko zvezde (ena jama na sredini, vse ostale jame pa so povezane s to jamo), pa moramo preiskati le to vozlišče (da najdemo jamo z darilom, zadostuje, če postavimo vprašanje le za to centralno jamo).

Ligra

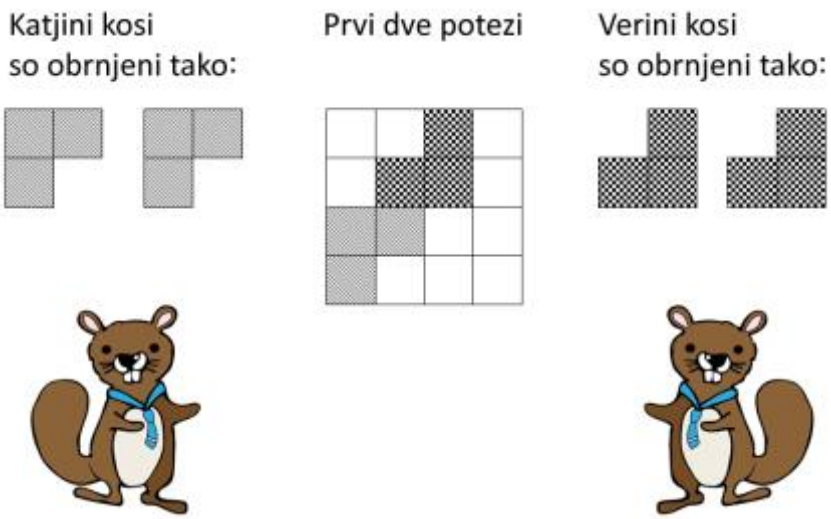
srednja šola (državno)

Ker smo na tekmovanju Bober, si poglejmo, kako se zabavajo mladi bobri. Bobrovki Katja in Vera najraje igrata namizno igro Ligra. Vsaka ima več kosov ploščic v obliki črke L, ki jih izmenično polagata na mrežo velikosti 4 x 4. Pri tem mora veljati:

- vsak kos, ki ga položi Katja, je obrnjen tako, kot prikazuje spodnja slika na levi;
- vsak kos, ki ga položi Vera, je obrnjen tako, kot prikazuje spodnja slika na desni;
- vsak kos mora biti v celoti položen na mrežo;
- nobena dva kosa se ne smeta prekrivati.

Ko je kos položen na mrežo, ga ne smemo več premikati. Igralec izgubi, ko je na potezi, a ne more postaviti svojega kosa na mrežo, ne da bi prekršil zgornja pravila.

Spodnja slika prikazuje primer, ko je igro začela Katja. Ko je drugič na potezi, lahko zmaga, če svoj kos postavi v spodnji desni kot.



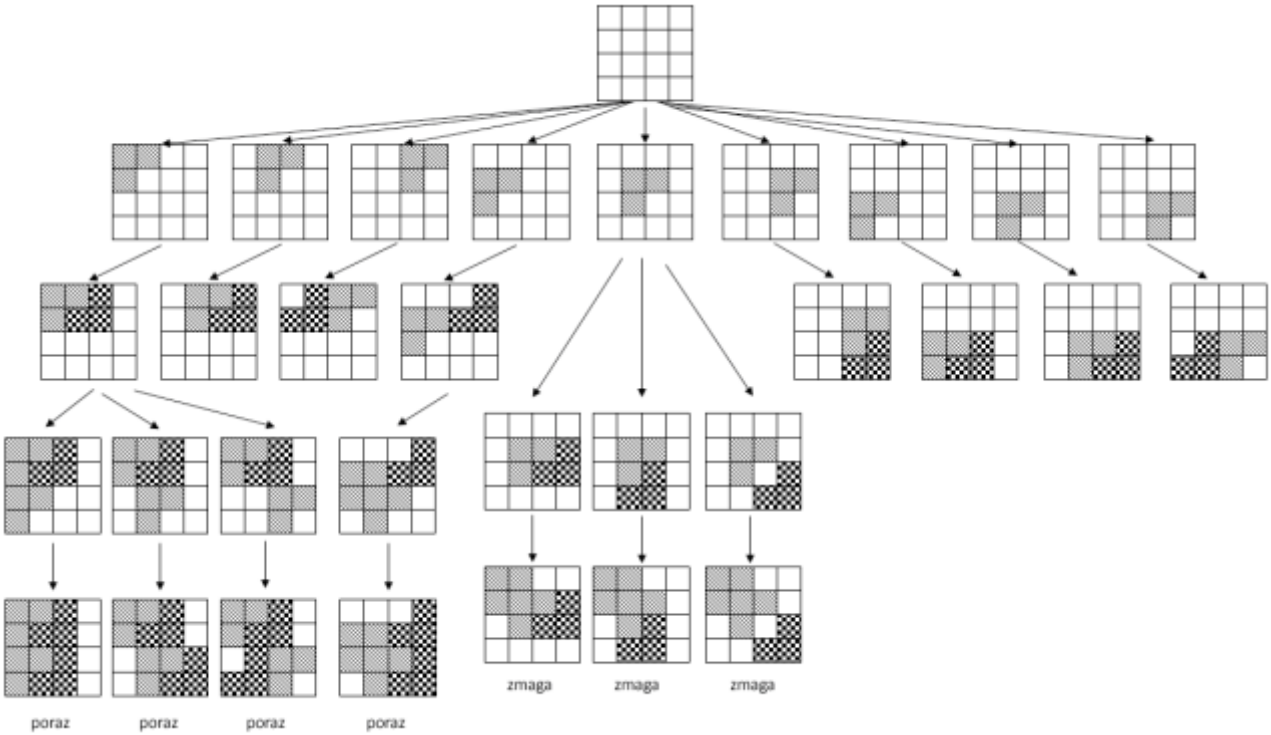
V koliko od devetih možnih začetnih potez ima Katja zagotovljeno zmago, ne glede na vse naslednje poteze?

Rešitev

Le ena začetna poteza Katji zagotovi zmago.

Če kos postavi v sredino mreže, ima Katja zagotovljeno zmago. Ne glede na to, kam Vera postavi svoj prvi kos, lahko Katja postavi drugi kos le v zgornji levi kot. Vera nato ne more več postaviti nobenega kosa, če se drži navedenih pravil igre.

Če Katja prvi kos postavi kamorkoli drugam, potem je vedno vsaj en možen potek igre, pri katerem Katja izgubi. Spodnji diagram prikazuje vseh devet različnih prvih postavitev (vse ostale postavitve lahko dobimo z upoštevanjem simetrije, torej če ustrezno obrnemo mrežo). Prikazan je tudi možen razvoj igre, ki vodi do Verine zmage (na levi). Podobno bi lahko razvili igro tudi pri postavitvah na desni strani drevesa.



Računalniško ozadje

V igrah navadno prikažemo vse možne kombinacije potez s pomočjo diagrama, kot je prikazan pri razlagi rešitve te naloge. Tako *drevo rešitev* ima v začetnem vozlišču začetno stanje igre (prazna mreža). Nato za vsako možno potezo pripravimo novo stanje igre, na katerega kaže puščica, ki to stanje povezuje s predhodnim stanjem. Če tako nadaljujemo do vseh možnih končnih stanj igre (stanj, pri katerih se igra konča), zgradimo celotno drevo. Drevo rešitev je posebna oblika *usmerjenega grafa*.

Drevo rešitev zgradimo in preiskujemo, kadar želimo odigrati ali preučiti igro. Pri tem včasih najprej raziščemo sosednja vozlišča, preden se usmerimo na vozlišča na naslednjem nivoju. Tako preiskovanje drevesa imenujemo *iskanje v širino* (angleško *breadth-first search* ali *BFS*). Včasih pa se nam bolj obnese, da najprej raziščemo kolikor lahko globoko po posamezni veji, preden se vrnemo in preiščemo še sosednja vozlišča. Tak način preiskovanja drevesa imenujemo *iskanje v globino* (angleško *depth-first search* ali *DFS*). Kakšen pristop bomo uporabili, je predvsem odvisno od vrste problema, ki ga rešujemo. Obe strategiji preiskovanja drevesa imata različne značilnosti in tudi zelo različno porabo pomnilnika.

Opica

srednja šola (državno)

Košato drevo  je obkroženo z dvema tankima drevesoma  in dvema palmama  , kot prikazuje spodnja slika.

Na drevesa postavimo pet vrst banan, poimenujmo jih kar s črkami O, P, R, S in T, in sicer na vsako drevo postavimo drugo vrsto banan. Opica skače z drevesa na drevo, poje banano in nadaljuje na naslednje drevo. Pri tem opica potrebuje:

- 3 sekunde, da skoči s košatega drevesa na katerokoli drugo drevo in poje banano na tem drevesu;
- 2 sekundi, da skoči s tankega drevesa na palmo (ali obratno) ter poje banano;
- 7 sekund, da skoči med dvema tankima drevesoma ali med dvema palmama in se pri tem izogne košatemu drevesu ter da na koncu še poje banano.

Opica skače po drevesih in poje po vrsti naslednje vrste banan: O, P, S, R, T, R in O.

Katere vrste banan so lahko na košatem drevesu, da je skupen čas, ki ga opica potrebuje za skakanje in hranjenje, čim manjši?

A. O ali P ali T.
B. O ali S ali T.
C. P ali S ali T.
D. P ali R ali S.

Rešitev

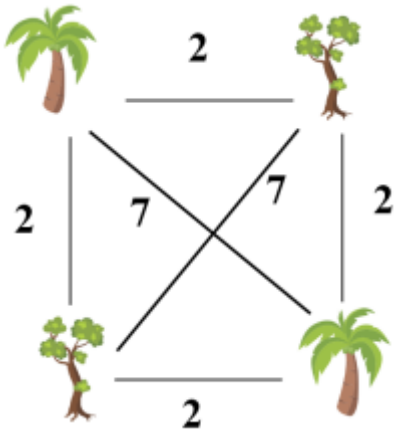
Pravilni odgovor je B: banane vrste O ali S ali T.

Podano zaporedje O, P, S, R, T, R, O vključuje vseh pet vrst banan, zato moramo imeti na vsakem drevesu drugo vrsto banan. Pri tem mora opica narediti šest skokov: O-P, P-S, S-R, R-T, T-R in R-O. Če želimo najti najmanjši skupni čas, morajo teh šest skokov sestavljati štirje 2-sekundni skoki (to so vsi možni najkrajši skoki) in

dva 3-sekundna skoka (preostala dva skoka sta druga najkrajša), torej vsi skoki skupaj zahtevajo 14 sekund.

Ker je tip banan R prisoten v štirih skokih (S-R, R-T, T-R in R-O) in ker morata skoka R-T in T-R trajati enako časa, so do/od R možni natanko trije skoki (to so do/od O, S in T) z morebitnimi različnimi časi. Posledično vrsta banan R ne more biti na košatem drevesu, saj bi v tem primeru trije skoki trajali 3 x 3 sekunde in skupen čas tako ne bi bil najmanjši. S tem smo izključili odgovor D, ki očitno ni pravilen. Je pa lahko na košatem drevesu vrsta banan T: v tem primeru skoka T-R in R-T zahtevata 3 sekunde, vsi ostali skoki pa 2 sekundi.

Preverimo še primer, ko so na košatem drevesu banane vrste P. V tem primeru trajata skoka O-P in P-S po 3 sekunde. Ker iščemo najkrajši skupni čas vseh skokov, morajo biti vsi skoki S-R, R-T, T-R in R-O dolgi največ 2 sekundi. To ni mogoče, kar lahko ugotovimo tudi, če poskusimo z O, R, S in T označiti drevesa na spodnji sliki.



Vrsta banan P torej ne more biti na košatem drevesu, zato za košato drevo ostanejo le še možnosti O, S in T (ustrezno razporeditev preverite sami).

Računalniško ozadje

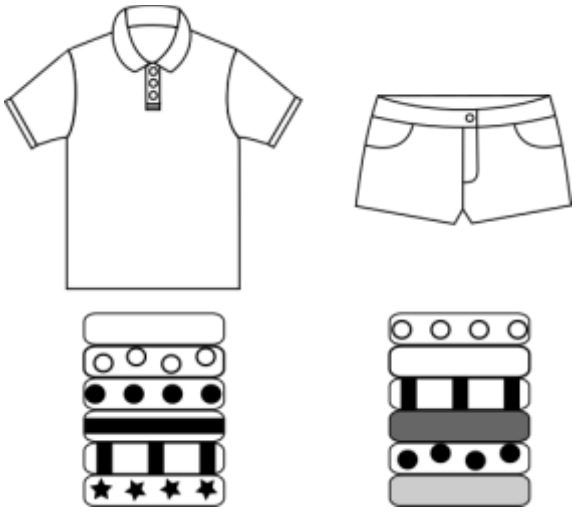
Ta problem opice vključuje iskanje najboljše (ali optimalne) rešitve podanega problema. Računalnike pogosto uporabljamo za iskanje največje ali najmanjše vrednosti nekih meritev.

Opisan primer se pogosto pojavlja v praksi. Če si zamislimo, da so drevesa okna različnih aplikacij na našem zaslonu na dotik, skoki opice pa so premiki uporabnikovega prsta od ene aplikacije k drugi, je naloga načrtovalca uporabniškega vmesnika, da poišče tako razporeditev aplikacij za pogosto uporabljana zaporedja uporabe, da za izvedbo uporabnik potrebuje kar najmanj časa.

Pranje oblačil

srednja šola (državno)

Borut se v ponedeljek zjutraj odpravlja na morje in bo s seboj vzel vso obleko, ki jo ima v omari v nedeljo zvečer (na sliki). Obleko bo na morju zložil v enakem vrstnem redu, kot jo ima v omari.



Borut vsako jutro z vrha vsakega kupa vzame in obleče čisto srajco in hlače. Njegova mama bo v torek in petek popoldne opala umazano obleko, ki se bo nato sušila še cel naslednji dan. Borut se bo cel dan potepal in se vrnil domov šele zvečer, zato mama obleke, ki jo bo nosil tisti dan, ne bo oprala isti dan.

Ko se bo obleka posušila, jo bo mama zložila v istem vrstnem redu, kot jo je Borut nosil.

V soboto bo Borut obiskal prijatelja Vilija. Kaj bo oblekel za obisk?

A) B) C) D) E) F)

Rešitev

Pravilni odgovor je F. Najprej poglejmo, kje je posamezna majica popoldne:

Dan	Borut obleče	Umazano	Omara	Se suši
Ponedeljek				
Torek				
Sreda				
Četrtek				
Petek				
Sobota				

Enako lahko naredimo tudi za hlače:

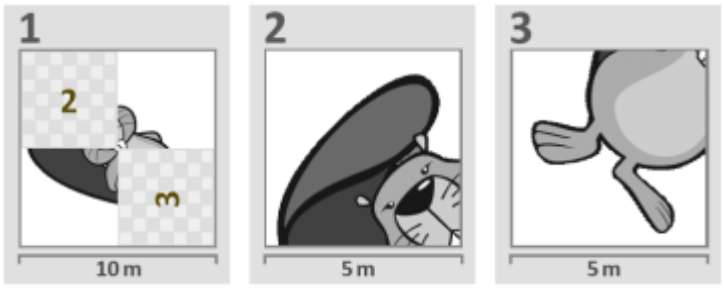
Dan	Borut obleče	Umazano	Omara	Se suši
Ponedeljek				
Torek				
Sreda				
Četrtek				
Petek				
Sobota				

Računalniško ozadje

Sklad je podatkovna struktura, ki pri shranjevanju in vračanju podatkov upošteva vrstni red shranjevanja. Tako zadnje na sklad shranjene podatke vedno dobimo najprej, šele nato starejše. Podobno Borut nalaga perilo v koš v obratnem vrstnem redu, kot jih je nosil (pravkar slečeno gre na vrh kupa perila), mama jih po pranju ponovno zloži v takem vrstnem redu, kot jih je nosil.

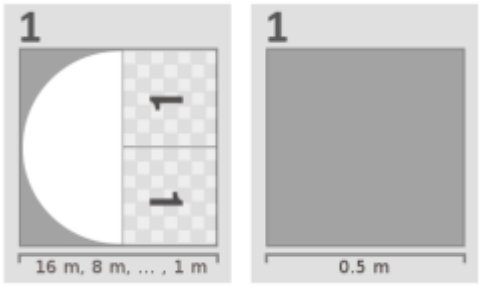
Slika v sliki

srednja šola (državno)

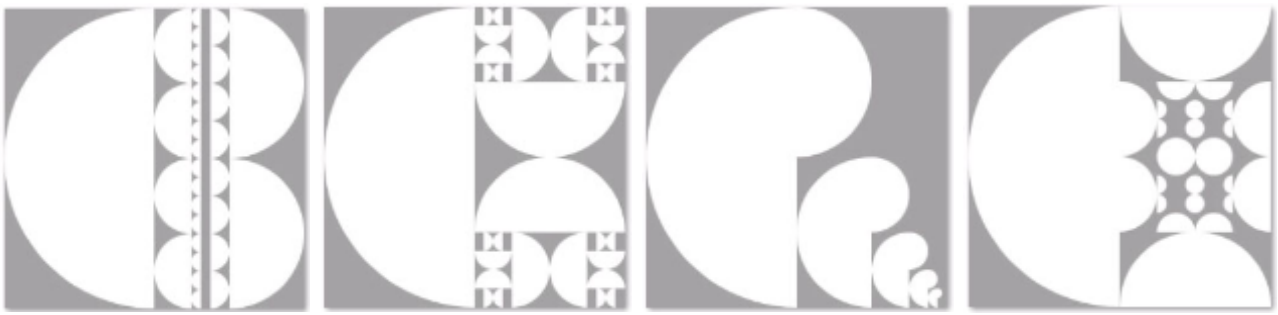


Začnemo pri sliki s številko 1. Velika bo 10 metrov. Na mesta, označena z 2 in 3, vstavimo sliki 2 in 3, ki ju obrnemo tako, kot sta obrnjeni številki. Kaj dobimo?

To vprašanje je bilo seveda preveč preprosto za tekmovanje z imenom Bober. Poglej tale navodila.



V sliko s številko 1 (velika bo 16 m) vstavimo sliki s številko 1. V obe sliki s številko 1 vstavimo sliko s številko 1. Če je vstavljena slika velika 8, 4, 2 ali 1 m vstavimo levo različico slike 1, sicer desno. Katero od spodnjih slik dobimo?



Rešitev

Drugo. Dovolj je opazovati drugi korak in razmisliti, kako se obrneta sliki.

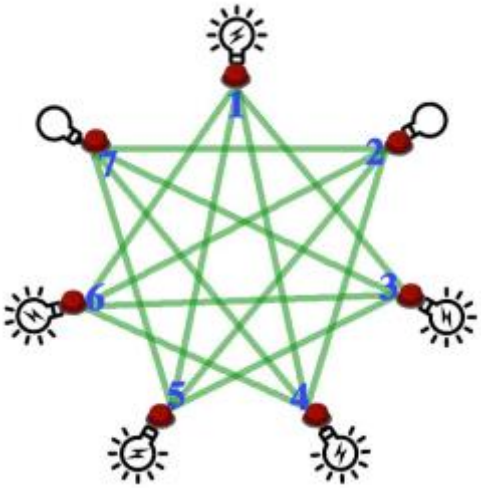
Računalniško ozadje

Navodilom – ali, če uporabimo bolj resno ime, funkcijam –, ki se sklicujejo nase, točneje, kličejo same sebe, pravimo “rekurzivne funkcije”. Čeprav se jih programerji začetniki bojijo, jih resni programerji radi uporabljajo, saj jim pogosto olajšajo delo.

Svetilke

srednja šola (državno)

Na vrtu imamo postavljenih sedem svetilk. Vsaka svetilka ima tudi gumb za prižiganje oz. ugašanje. Vsak gumb je povezan z gumbi vseh ostalih svetilk, razen z gumboma obeh sosednjih svetilk (kot to prikazuje spodnja slika).



Kadar pritisnemo na en gumb, spremeni stanje vseh pet povezanih svetilk (če je svetilka prižgana, se ugasne, če pa je ugasnjena, se prižge).

Tako v primeru, ko so vse svetilke ugasnjene in pritisnemo na gumb 1, zasvetijo svetilke 1, 3, 4, 5 in 6 (stanje prikazuje zgornja slika). Če nato pritisnemo še gumb 2, se prižgeta svetilki 2 in 7, svetilke 4, 5 in 6 pa se ugasnejo.

Na začetku so vse svetilke ugasnjene. Če želimo prižgati vse svetilke, najmanj kolikokrat moramo pritisniti na gumbe svetilk?

Rešitev

Vsak gumb moramo pritisniti enkrat, torej potrebujemo sedem pritiskov na gumbe. Pri tem vrstni red pritisnjenih gumbov ni pomemben.

Pri razmišljanju nam pomagata dve dejstvi:

1) Če na isti gumb pritisnemo dvakrat, nam to ni v pomoč.
2) Vrstni red, v katerem pritiskamo na gumbe, ni pomemben.

Torej iščemo množico preklopov svetilk. Začetna pozicija in tudi želeno stanje sta simetrična, zato bodo zelo verjetno tudi zahtevane operacije simetrično porazdeljene. Če na vsak gumb pritisnemo natanko enkrat, bo vsaka svetilka spremenila stanje natanko petkrat: enkrat neposredno (pritisk na gumb pri tej

svetilki) in štirikrat posredno (pritisk na povezane gumbe). Če torej vsaki svetilki spremenimo stanje x-krat, kjer je x liho število, potem morajo biti vse svetilke na koncu prižgane.

Računalniško ozadje

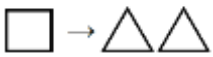
Delovanje povezanih svetilk iz naloge je podobno delovanju operacije XOR (izključujoči ali). Svetilka je lahko prižgana ali ugasnjena, torej ima le dve stanji, kar lahko zapišemo z uporabo 0 in 1 (binarna logika).

Predvidevanje dinamike sistema stanj je zelo pomemben del računskega razmišljanja, saj ne moremo vedno našteti vseh možnih kombinacij stanj sistema (ker jih je enostavno preveč). Zato moramo ugotoviti zakonitosti takega sistema in jih uporabiti pri iskanju rešitve.

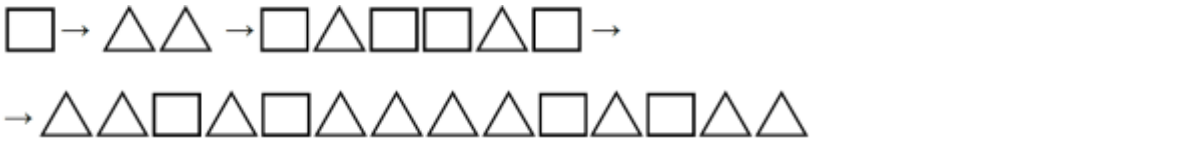
Zamenjave

srednja šola (državno)

Anja igra igro, v kateri uporablja karte v obliki geometričnih likov (kvadrat, trikotnik in krog). Začne z eno karto, nato pa uporablja naslednji dve pravili za zamenjavo kart:

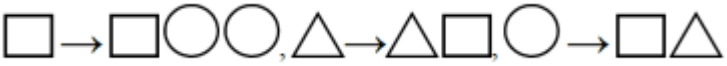
- karta s kvadratom se zamenja z dvema kartama s trikotnikom: 
- karta s trikotnikom se zamenja s tremi kartami, kjer je prva s kvadratom, druga s trikotnikom in tretja ponovno s kvadratom: 

Če začne igro z eno karto s kvadratom in v naslednjem koraku pri vsaki karti uporabi zgornji pravili za zamenjavo, bo po treh korakih dobila naslednje karte:



Katero od navedenih množic pravil zamenjave moramo uporabiti, da od začetne ene karte pridemo do zaporedja na desni? 

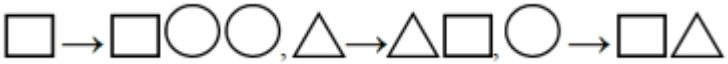
A. 

B. 

C. 

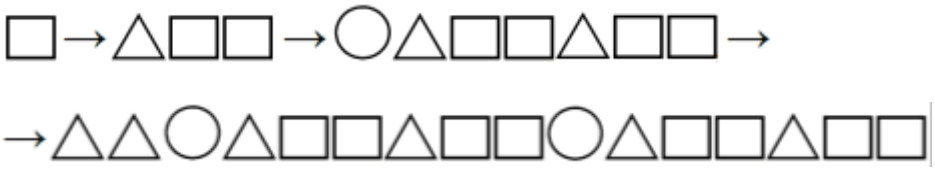
D. 

Rešitev

Pravilni odgovor je B: 

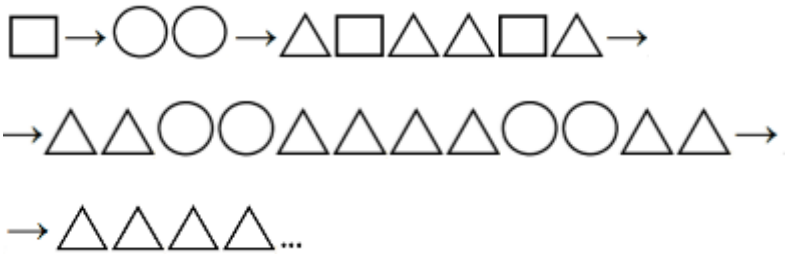
Če začnemo s trikotnikom, lahko z uporabo pravil pod B dobimo iskano zaporedje. Pri tem velja:

Razmislimo še o ostalih možnostih. Če uporabimo množico pravil pod A in začnemo s trikotnikom ali krogom, ne moremo nikoli dobiti kvadrata. Če pa začnemo s kvadratom, potem po treh korakih dobimo:



Dobljeno zaporednje je že predolgo, vsak naslednji korak pa nam ustvari še daljše zaporedje.

Če uporabimo pravila pod C in začnemo s trikotnikom, lahko ustvarimo le zaporedje trikotnikov. Če začnemo s kvadratom, po nekaj korakih dobimo:



Prvih štirih trikotnikov ne moremo nikoli več spremeniti v zaporedje  , saj se trikotniki vedno zamenjujejo z dvema trikotnikoma. Če pa začnemo s krogom, ponovno po nekaj korakih dobimo na začetku štiri trikotnike, ki jih, podobno kot v predhodnem primeru, ne moremo več spremeniti v ustrezno zaporedje  :

Tudi z množico pravil pod D ne moremo ustvariti iskanega zaporedja. Če začnemo s trikotnikom, dobimo v tretjem koraku na začetku zaporedja kvadrat, v četrtem koraku pa na začetku tri trikotnike, celo zaporedje pa je že predolgo. Če začnemo s kvadratom, dobimo po tretjem koraku predolgo zaporedje, ki tudi ni pravo (začne se s tremi trikotniki). Tudi če pa igro začnemo s krogom, je zaporedje po treh korakih še prekratko (in se začne s kvadratom), naslednji korak pa nam prinese že predolgo zaporednje, ki se poleg tega začne s tremi trikotniki.

Računalniško ozadje

Pravila, ki smo jih uporabili v nalogi, so primer pravil za prepisovanje nizov, kot jih uporabljamo v *kontekstno neodvisnih gramatikah* ali drugih sistemih gramatik. Z njimi opisujemo različne stvari, kot so:

- naravni pojavi (na primer rast rastlin);
- naravni jeziki (na primer slovnična pravila za sestavljanje povedi);

- formalni jeziki (na primer struktura programskih jezikov).

Vprašanje v nalogi sprašuje po *izpeljavi* (angleško *derivation*) ali *razčlembi* (angleško *parse*) podane besede s pomočjo različnih pravil. *Razčlenjevanje* (angleško *parsing*) je eden pomembnejših korakov v procesu prevajanja programa iz besed, ki jih razumejo ljudje, v binarna števila, ki jih razumejo računalniki.

Rdeče in modre frnikole

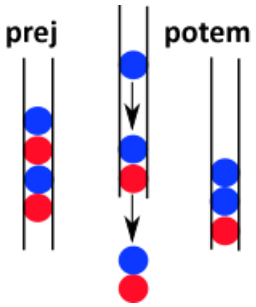
srednja šola (državno)

Ernest preizkuša novo igro na svojem telefonu. Igra se začne z najmanj tremi barvnimi frnikolami v stolpcu, ki ga pred začetkom igre pripravi Ernest. Vsaka frnikola je ali rdeče ali modre barve.

Po pritisku na gumb GO se stanje stolpca frnikol spremeni tako, da spodnji dve frnikoli izpadeta, na vrh stolpca pa se dodajo nove frnikole glede na naslednji dve pravili:

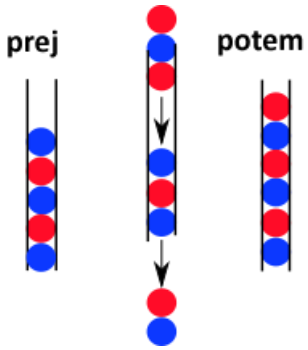
Če je prva izpadla frnikola **rdeča**,

se na vrhu stolpca pojavi ena modra frnikola.



Če je prva izpadla frnikola **modra**,

se na vrhu stolpca pojavijo tri nove frnikole: ena rdeča, ena modra in še ena rdeča.



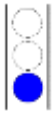
Če so v stolpcu vsaj tri frnikole, lahko igro nadaljujemo s pritiskom na gumb GO. Igra se konča, če oz. ko sta v stolpcu le dve frnikoli ali manj.

Če na primer Ernest pripravi začetni stolpec frnikol, ki ga prikazuje slika na desni, bosta na koncu ostali v stolpcu le dve modri frnikoli (po petih pritiskih na gumb GO) in igra se bo zaključila.

Kakšen začetni stolpec treh frnikol mora uporabiti Ernest, da se igra ne bo nikoli končala?

Rešitev

Igra se nikoli ne zaključi, če je spodnja frnikola v stolpcu modra.

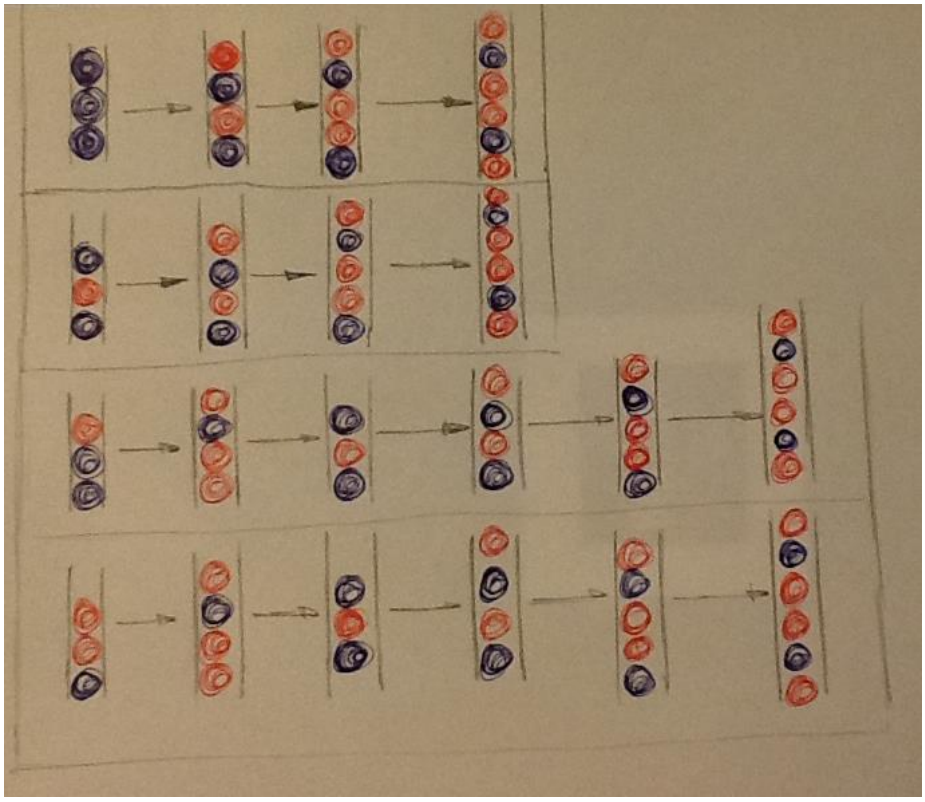


Če ima začetni stolpec le tri frnikole in je spodnja frnikola rdeča, se igra zaključi že v naslednjem koraku.

Če pa je v stolpcu treh frnikol spodnja frnikola modra, bo po najmanj 5 korakih igre (po petih pritiskih na gumb GO) v stolpcu naslednja kombinacija frnikol: RMRRMR (glej spodnjo sliko).

Če barve frnikol v stolpcu zapišemo od zgoraj navzdol, dobimo naslednje možnosti razvoja igre:

MMM → RMRM → RMRRM → RMRRMR
MRM → RMRM → RMRRM → RMRRMR
RMM → RMRR → MRM → RMRM → RMRRM → RMRRMR
RRM → RMRR → MRM → RMRM → RMRRM → RMRRMR



Nato se igra začne ponavljati, saj po štirih pritiskih na gumb GO ponovno dobimo enako kombinacijo frnikol RMRRMR: RMRRMR → MRMRR → MMRM → RMRMM → RMRRMR

Računalniško ozadje

Naloga predstavlja *Postov produkcijski sistem* (angleško *Post production system*), model računanja na nizih simbolov, ki ga je razvil Emil Leon Post v dvajsetih letih prejšnjega stoletja, prvič pa je bil objavljen leta 1943.

Model računanja temelji na prepisovanju nizov, kjer s pomočjo različnih pravil zamenjujemo podnize v nizih z drugimi nizi in tako dobimo nove nize. Na model lahko gledamo tudi kot na (končni) sistem objektov s (končno) množico relacij, ki definirajo možne manipulacije in transformacije podanih objektov.

V teoretičnem računalništvu take sisteme imenujemo *kontekstno neodvisne gramatike* (angleško *context-free grammars*). Tako lahko na primer sistem množenj in seštevanj definiramo kot enostavno kontekstno neodvisno gramatiko z le nekaj pravili. Drug uporaben primer je uporaba primerne gramatike za definicijo programskega jezika.