**Računalništvo (fizika): 4. izpit**

29. 8. 2016

Naloge rešujte na strežniku Tomo (`https://tomo.fmf.uni-lj.si/`). V sistem se prijavite s svojim uporabniškim računom na omrežju FMF.

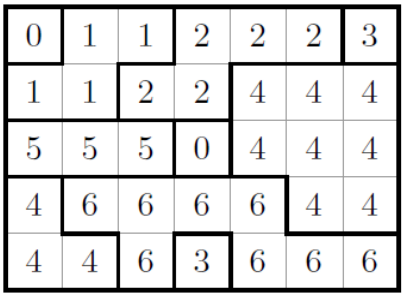
Brskanje po spletni učilnici, video zapiskih predavanj in *uradni* dokumentaciji za Python *je dovoljeno*. Brskanje po `stackoverflow.com`, `rosettacode.org` in ostalih spletnih straneh, ki vsebujejo rešitve nalog in nasvete o tem, kako se rešuje naloge, *ni dovoljeno*. Komunikacija s komerkoli, razen z asistentom, *ni dovoljena*.

Čas reševanja je 150 minut. Doseženih 100 točk šteje za maksimalno oceno. Veliko uspeha!

## 1. naloga (25 točk)

Na Slovenskem uradu za vpeljavo novih davkov so ugotovili, da ima vsak Slovenec svoje posestvo obdano z ograjo. Ker za to še ni predpisanega nobenega davka, so se odločili, da bodo nemudoma pripravili nov zakon. Svojim strokovnjakom so naročili, naj naredijo informativne izračune za posamezne občine.

Zaradi enostavnosti so predpostavili, da je vsaka občina pravokotne oblike in razdeljena na kvadratke velikosti 1 km × 1 km. Vsak kvadratek pripada natanko enemu izmed $n$ lastnikov. Ograje stojijo tako na občinski meji, kot tudi med vsemi kvadratki, ki pripadajo različnima lastnikoma. Podatki so zbrani v matriki velikosti $h \times w$, elementi pa so identifikacijske številke lastnikov. Primer (ograje so predstavljene z debelo črto):



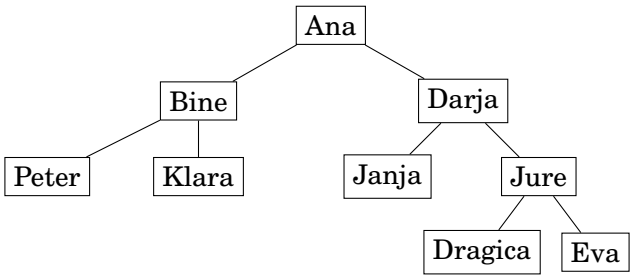

Sestavite funkcijo `davek`($m$), ki kot argument dobi matriko $m$ s podatki in vrne slovar. Ključi v slovarju so identifikacijske številke lastnikov zemljišč, vrednosti pa skupna dolžina ograje, ki omejuje posestvo tega lastnika. Ker je vsak Slovenec prepričan, da ograja stoji na njegovem zemljišču in ne na sosedovem, bomo skupno ograjo seveda zaračunali dvojno. Zgled:

```
>>> davek([[0, 1, 1, 2, 2, 2, 3],
           [1, 1, 2, 2, 4, 4, 4],
           [5, 5, 5, 0, 4, 4, 4],
           [4, 6, 6, 6, 6, 4, 4],
           [4, 4, 6, 3, 6, 6, 6]])
{0: 8, 1: 10, 2: 12, 3: 8, 4: 20, 5: 8, 6: 18}
```

**2. naloga (25 točk)**

Neko podjetje ima direktorja, ki sta mu podrejena dva pomočnika direktorja. Pomočnika imata svoje podrejene, ki imajo spet podrejene itn. Takšno hierarhijo lahko predstavimo z drevesom.

Podatkovna struktura `Drevo` ima v vsakem vozlišču shranjeno ime zaposlenega ter seznam njemu podrejenih. V korenu drevesa je direktor. Delno implementiran razred `Drevo` najdete na strežniku. Vsako vozlišče ima atributa `ime` in `podrejeni`. Zgled:




Podjetje med poletjem organizira piknik, kjer zaposleni vlečejo vrv. Na vsako stran vrvi se postavi en pomočnik direktorja, nato pa se mu pridružijo še vsi njegovi (neposredno in posredno) podrejeni. Tako dobimo dve ekipi.

Sestavite metodo `ekipi(self)`, tako da `direktor.ekipi()` vrne par množic z imeni zaposlenih, ki sestavljajo obe ekipi. (Vseeno je, kaj metoda počne, če ima podjetje več ali manj kot dva pomočnika direktorja.) Zgled (če je `direktor` takšno drevo, kot je prikazano zgoraj):

```
>>> direktor.ekipi()
({'Bine', 'Klara', 'Peter'}, {'Eva', 'Janja', 'Jure', 'Darja', 'Dragica'})
```

**3. naloga (25 točk)**

Med poletnimi počitnicami asistent za Računalništvo odpotuje v neko priljubljeno letovišče na jugu Balkanskega polotoka. Tam cele dneve poseda v krčmi in pije sok. Naroča si po en kozarec naenkrat in sicer bodisi kozarec pomarančnega bodisi kozarec jabolčnega soka. To počne od jutra do poznega večera. Gostilničar je opazil, da vsak dan popije med $a$ in $b$ kozarcev pijače (seveda velja $a \leq b$). Asistent ima tudi to posebno navado, da pomarančni sok pije izključno v serijah po $k \geq 1$ zaporednih kozarcev. Vsaki takšni seriji vedno sledi vsaj eno kozarec jabolčnega soka ali pa asistent zaključi s popivanjem za ta dan.

Ko je tako posedal in pil sok, se je asistent vprašal, na koliko različnih načinov lahko naroči pijačo za dane $a$, $b$ in $k$? Menil je, da je to izvrstna izpitna naloga.

Sestavite metodo `popivanje`$(a,b,k)$, ki pove, na koliko načino lahko asistent naroči pijačo pri danih $a$, $b$ in $k$. Zgled:

```
>>> popivanje(5, 7, 3)
16
```

**4. naloga (35 točk)**

Med poletjem je na dopustu profesorju dolgčas, zato se igra z $n$ barvnimi kartonskimi karticami. Vsaka stran kartice je pobarvana z neko izmed $k$ barv. Karte premeša in jih razporedi na mizo, kar predstavimo s seznamom parov števil, na primer:

$$[(3,9),\ (9,3),\ (7,5),\ (1,9),\ (2,2)].$$

Prva komponenta para je oznaka barve na zgornji strani in druga oznaka barve na spodnji strani kartice. Torej je v zgornjem zgledu na mizi 5 kartic; prva ima zgoraj barvo 3 in spodaj barvo 9, druga zgoraj barvo 9 in spodaj 3 itn.

Profesor lahko v eni potezi igre obrne eno od kartic. Igra se konča, ko je na mizi vsaj polovica kartic enake barve (gledamo zgornjo stran). Profesorja je zanimalo najmanjše število potez, ki jih potrebuje, da dokonča igro. V zgornjem primeru mora obrniti prvo in četrto kartico. Tako bodo na mizi 3 kartice, katerih vidna stran je barve 9. Profesor je menil, da je to izvrstna izpitna naloga.

Sestavite funkcijo `min_poteze`($\ell$), ki za dani začetni nabor kart $\ell$ ugotovi minimalno število potez do zmage. Če igre ni mogoče končati, naj funkcija vrne `None`. Za vse točke naj ima rešitev časovno zahtevnost $O(n)$, kjer je $n$ število kartic. Zgled:

```
>>> min_poteze([(3, 9), (9, 3), (7, 5), (1, 9), (2, 2)])
2
```