

Računalništvo (fizika): 4. izpit

7. 9. 2017

Naloge rešujte na strežniku Tomo, <https://www.projekt-tomo.si>.

Brskanje po spletni učilnici, nalogah na strežniku Tomo, video zapiskih predavanj in *uradni* dokumentaciji za Python *je dovoljeno*. Brskanje po 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 180 minut. Doseženih 100 točk šteje za maksimalno oceno. Veliko uspeha!

1. naloga (25 točk)

Lego objavlja načrte za sestavljanje raznih izdelkov iz Lego kock. Za vsak načrt predstavimo kocke, ki jih potrebujemo, s slovarjem slovarjev. Na primer,

```
{ '4x1'      : { 'modra':3, 'zelena':5 },  
  'vratca'   : { 'bela':1 },  
  '2x2'      : { 'bela': 10, 'zelena': 6 } }
```

pomeni, da imamo 3 modre in 5 zelenih kock velikosti 4×1 , ena bela vratca ter 10 belih in 6 zelenih kock velikosti 2×2 .

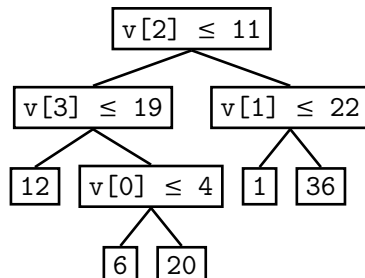
Aljaž ima doma zalogo Lego kock, ki jih prav tako predstavi s takim slovarjem slovarjev. Želi izvedeti, katere modelčke lahko sestavi s svojo zalogo.

Sestavite funkcijo `modelcki(zaloga, modelcki)`, ki sprejme slovar zaloga, ki opisuje zalogo kock, ter seznam slovarjev `modelcki`, ki opisuje vse modelčke, ki jih je objavil Lego. Funkcija naj vrne seznam tistih modelčkov, ki jih lahko sestavimo z dano zalogo. (Mišljeno je, da sestavimo vsak modelček posebej, ne vse nenkrat.)

2. naloga (30 točk)

V podatkovnem rudarjenju imamo opravka s podatki, iz katerih napovedujemo neke vrednosti. En podatek predstavimo kot tabelo števil, npr., $v = [1, 42, 22, 18, 101]$.

Odločitveno drevo je poodatkovna struktura, ki jo lahko uporabimo za napovedovanje vrednosti iz danega podatka. V notranjih vozliščih odločitvenega drevesa (tj., vozliščih, ki niso listi) so shranjene neenačbe oblike $v[i] \leq m$, ki usmerjajo iskanje levo ali desno, v listih pa so shranjena t.i. napovedne vrednosti. Primer:



Za dani vektor napovemo pripadajočo vrednost tako, da začnemo pri korenu drevesa in ponavljamo naslednji postopek: če smo v listu, napovemo vrednost, ki je zapisana v njem, če pa smo v notranjem vozlišču, preverimo neenačbo. Če ta velja, iščemo naprej v levem poddrevesu, sicer pa desnem.

Odločitveno drevo predstavimo z objektom razreda `OdlocitvenoDrevo`. Objekt vsebuje atribut `je_list`. Če je ta enak `True`, potem objekt predstavlja list in vsebuje še atribut `vrednost`. Če je atribut `je_list` enak `False`, potem objekt vsebuje še naslednje attribute:

- `indeks`: indeks, na katerem bomo izvedli test v tem vozlišču
- `meja`: vrednost na desni strani neenačbe
- `levo`: levo poddrevo
- `desno`: desno poddrevo

a) (15 točk) Razredu `OdlocitvenoDrevo`, dodajte metodo `napoved(v)`, ki za dani vektor v poišče njegovo napoved.

b) (15 točk) Razredu `OdlocitvenoDrevo`, dodajte metodo `napoved_manjkajoci(v)`, ki sprejme vektor v , le da ta lahko vsebuje manjkajoče komponente, ki so predstavljene z `None`. Na primer, $v = [1, \text{None}, 22, \text{None}, 101]$ predstavlja vektor z neznanimi komponentama na indeksih 1 in 3. Metoda `napoved_manjkajoci`, naj vrne **množico** vseh napovedi, ki bi jih lahko dobili, če bi manjkajoče vrednosti zamenjali s poljubnimi vrednostmi.

3. naloga (30 točk)

Aljaž je imel *urejeno* tabelo celih števil $a = [a_0, \dots, a_{n-1}]$ z n elementi. Andrej je eno od števil v tabeli naključno spremenil, tako da tabela ni več nujno urejena.

a) (15 točk) Sestavite funkcijo `uredi(a)`, ki sprejme tako tabelo a in jo uredi na mestu. (Funkcija torej ne vrne ničesar, ampak spremeni a , da bo urejena.) Funkcija naj deluje v času $O(n)$.

b) (15 točk) Andrej je naključno spremenil 42 elementov v tabeli. Sestavite funkcijo `uredi42(a)`, ki sprejme tako tabelo a in jo uredi na mestu. Funkcija naj deluje v času $O(n)$.

4. naloga (30 točk)

Vzdolž Nila je imelo egipčansko kraljestvo n polj žita, ob vsakem polju pa veliko skladišče, v katerega so ob žetvi shranili žito s polja. Faraon je zaukazal, da je treba po uspešno opravljeni žetvi vse žito prepeljati v največ k skladišč.

Faronovi podaniki so se odločili, da bodo izbrali k skladišč tako, da bo cena prevoza vanje minimalna. Če z ladjo po Nilu prepeljemo m amfor žita ℓ nilskih milj daleč, to stane $m \cdot \ell$ faronskih zlatnikov. Faronov visoki programer je naredil seznam parov

$$s = [(x_1, m_1), (x_2, m_2), \dots, (x_n, m_k)]$$

kjer je x_i lokacija i -tega skladišča vzdolž Nila in m_i število amfor, ki so tam shranjene.

a) (15 točk) Sestavite funkcijo $\text{prevoz1}(s)$, ki sprejme seznam skladišč in *čim bolj učinkovito* izračuna najmanjšo ceno prevoza žita v *eno* skladišče (se pravi, da je $k = 1$). V komentar zapišite, kakšna je časovna zahtevnost vaše rešitve.

b) (15 točk) Sestavite funkcijo $\text{prevoz2}(s)$, ki sprejme seznam skladišč s in *čim bolj učinkovito* izračuna najmanjšo ceno prevoza žita v *dve* skladišči (se pravi, da je $k = 2$). V komentar zapišite, kakšna je časovna zahtevnost vaše rešitve.