

Programiranje I: 1. izpit

6. februar 2011

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

1. naloga (35 točk)

Stanje na bančnih računih oseb predstavimo s slovarjem, ki ime računa preslika v trenutno stanje na računu, na primer:

```
{'Ana': 5, 'Bine': 120, 'Cene': 310, 'Darko': 42}
```

Prenos med računoma je urejena trojica (A,B,X) , ki pomeni, da iz računa A na račun B prenesemo X .

a) (15 točk) Sestavite funkcijo `naloga1a(s, p)`, ki sprejme stanje s na bančnih računih in seznam prenosov p ter osveži stanje. Na primer

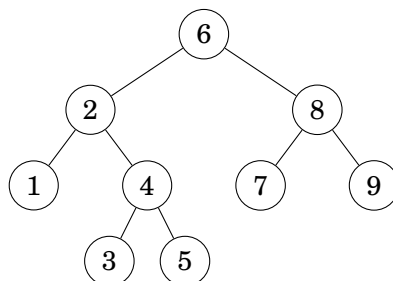
```
>>> s = {'Ana': 5, 'Bine': 120, 'Cene': 310, 'Darko': 42}
>>> naloga1a(s, [('Ana', 'Bine', 10), ('Bine', 'Cene', 50), ('Cene', 'Bine', 10)])
>>> s
{'Ana': -5, 'Bine': 90, 'Cene': 350, 'Darko': 42}
```

b) (20 točk) Sestavite funkcijo `naloga1b(s1, s2)`, ki sprejme dve stanji in vrne seznam prenosov, s katerim iz $s1$ dobimo $s2$. Predpostaviti smete, da imata $s1$ in $s2$ enake ključe. Funkcija naj vrne `None`, če tak seznam prenosov ne obstaja. Primer:

```
>>> naloga1b({'Ana': 10, 'Bine': 20, 'Cene': 40}, {'Ana': 20, 'Bine': 25, 'Cene': 25})
[('Cene', 'Bine', 5), ('Cene', 'Ana', 10)]
```

2. naloga (25 točk)

Pot v iskalnem drevesu lahko predstavimo s seznamom vrednosti `True` in `False`. Vrednost `True` pove, da naredimo korak v levo poddrevo, vrednost `False` pa, da ga naredimo v desno poddrevo. Tako v drevesu



seznam `[True, False, True]` predstavlja pot do 3, seznam `[False]` pot do 8, seznam `[]` pot do korena 6, seznam `[False, False, True]` pa ne predstavlja poti v drevesu.

a) (10 točk) Razredu `IskalnoDrevo` dodajte metodo `naloga2a(self, p)`, ki vrne element v danem drevesu, do katerega pridemo s potjo p . Če v drevesu take poti ni, naj metoda vrne `None`.

b) (15 točk) Razredu `IskalnoDrevo` dodajte metodo `naloga2b(self, x)`, ki vrne seznam, ki predstavlja pot do elementa x . Če v drevesu danega elementa ni, naj metoda vrne `None`.

3. naloga (25 točk)

V Mathematici sestavite funkcijo `naloga3[z_, v_]`, ki sprejme zmagovalno kombinacijo z , izžrebano na Lotu, ter seznam v vplačanih kombinacij. Funkcija vrne število posameznih zadetkov, torej koliko štiric, petic, šestic in sedmic je bilo vplačanih. Dobitka $6 + 1$, Lotka in ostalih nagrad ne upoštevamo. Na primer:

```
In[1]:= naloga3[{14, 16, 19, 20, 27, 31, 38}, {
      {14, 16, 19, 20, 26, 31, 35},
      {14, 16, 19, 20, 27, 30, 38},
      {14, 16, 19, 20, 26, 30, 37},
      {10, 14, 18, 22, 26, 30, 34},
      {14, 16, 19, 20, 27, 31, 38},
      {14, 16, 19, 20, 27, 30, 37}}]
Out[1]= {{5, 2}, {6, 1}, {4, 1}, {7, 1}}
In[2]:= naloga3[{14, 16, 19, 20, 27, 31, 38}, {
      {38, 31, 27, 20, 19, 16, 14},
      {10, 14, 18, 22, 26, 30, 34}}]
Out[2]= {{7, 1}}
```

Namig: Oglejte si, kaj počne funkcija `Tally`.

4. naloga (25 točk)

Transpozicija v tabeli t je zamenjava dveh elementov. Transpozicijo, ki zamenja $t[i]$ in $t[j]$ predstavimo z urejenim parom (i, j) . Sestavite funkcijo `naloga4(t)`, ki vrne seznam transpozicij, s katerimi uredimo dano tabelo t . Na primer

```
>>> naloga4([9, 7, 5, 3])
[(0, 3), (1, 2)]
>>> naloga4([3, 5, 7, 9])
[]
```