Programiranje I: 2. izpit

7. maj 2013

Čas reševanja je 120 minut. Veliko uspeha!

1. naloga (25 točk)

V *Mathematici* lahko sprehod v ravnini predstavimo s seznamom korakov $\{\{e_1,d_1\},\ldots,\{e_n,d_n\}\}$, kjer so e_i enotski vektorji, ki predstavljajo smeri korakov, d_i pa pozitivna realna števila, ki predstavljajo njihove dolžine.

a) Sestavite funkcijo `naloga1a[sez_]`, ki vrne točke, ki jih obišče sprehod, podan s seznamom korakov `sez` v gornji obliki.

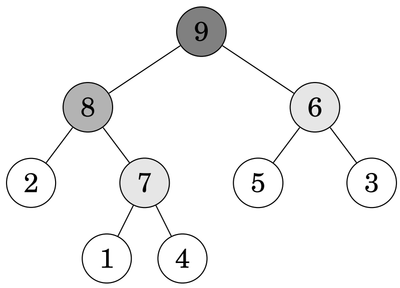
```
In[1]:= naloga1a[{{{0, 1}, 2}, {{1, 0}, 1}, {{-1, 0}, 4}}]
Out[1]= {{0, 0}, {0, 2}, {1, 2}, {-3, 2}}
In[2]:= naloga1a[{{{0, 1}, 2}, {{1, 0}, 1}, {{-1, 0}, 4}, {{3/5, 4/5}, 5}}]
Out[2]= {{0, 0}, {0, 2}, {1, 2}, {-3, 2}, {0, 6}}
```

b) Sestavite funkcijo `naloga1b[sez_]`, ki iz seznama obiskanih točk `sez` izračuna seznam korakov v gornji obliki.

```
In[3]:= naloga1b[{{0, 0}, {0, 2}, {1, 2}, {-3, 2}}]
Out[3]= {{{0, 1}, 2}, {{1, 0}, 1}, {{-1, 0}, 4}}
In[4]:= naloga1b[{{0, 0}, {0, 2}, {1, 2}, {-3, 2}, {0, 6}}]
Out[4]= {{{0, 1}, 2}, {{1, 0}, 1}, {{-1, 0}, 4}, {{3/5, 4/5}, 5}}
```

2. naloga (25 točk)

Vsako dvojiško drevo lahko razdelimo na plasti tako, da so v prvi plasti vsi listi drevesa, v drugi plasti vsa vozlišča, ki imajo otroke v prvi plasti, in tako naprej. Primer razdelitve drevesa na plasti:



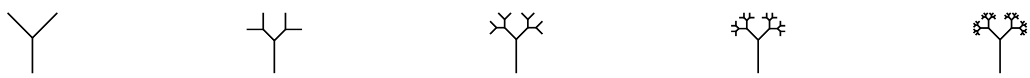
Razredu `Drevo` dodajte metodo `naloga2(self, n)`, ki vrne množico vrednosti vseh vozlišč v plasti n. Metoda naj drevesa ne spreminja.

Pri zgornjem drevesu d tako torej velja:

```
>>> d.naloga2(1)
{1, 2, 3, 4, 5}
>>> d.naloga2(2)
{6, 7}
>>> d.naloga2(3)
{8}
```

3. naloga (25 točk)

V *Mathematici* sestavite funkcijo `naloga3[n_]`, ki izriše sledeče fraktale:



`naloga3[1]` `naloga3[2]` `naloga3[3]` `naloga3[4]` `naloga3[5]`

4. naloga (25 točk)

Sestavite funkcijo `naloga4(t)`, ki za tabelo `t` oblike

$$[a_k, a_{k+1}, \ldots, a_n, a_1, a_2, \ldots, a_{k-1}],$$

kjer velja

$$a_1 < a_2 < \cdots < a_n,$$

v času $O(\log n)$ poišče število k.