

## Programiranje I: 2. izpit

19. februar 2016

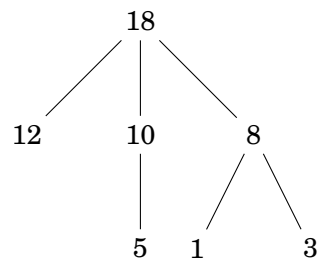
Čas reševanja je 120 minut. **Funkcij ne pozabite opremiti z ustrezno signaturo.** Veliko uspeha!

### ~~1. naloga (Kopice, 40 točk)~~

Drevesa s poljubnim številom sinov in vrednostmi v vozliščih lahko v Haskellu predstavimo s podatkovnim tipom:

```
data Drevo a = Drevo a [Drevo a]
```

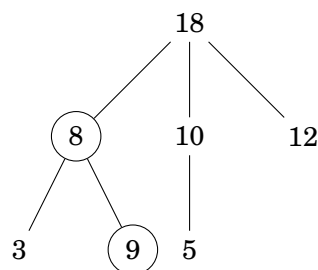
Na primer, drevo



bi predstavili kot

```
Drevo 18 [
  Drevo 12 [],
  Drevo 10 [Drevo 5 []],
  Drevo 8 [Drevo 1 [], Drevo 3 []]
]
```

Za takšno drevo pravimo, da je *kopica*, če je vrednost korena večja od vseh vrednosti korenov sinov, ti pa so prav tako kopice. Na primer, zgornje drevo je kopica, spodnje pa ne, ker je vrednost 9 večja od 8.



~~a) (10 točk)~~ Sestavite funkcijo `vsota`, ki vrne vsoto vseh elementov v drevesu. Na primer, za prvo drevo bi funkcija vrnila 57, za drugo pa 65.

~~b) (10 točk)~~ Sestavite funkcijo `jeKopica`, ki vrne `True`, če je dano drevo kopica, in `False` sicer.

~~c) (10 točk)~~ Sestavite funkcijo

```
odstraniMax :: Ord a => Drevo a -> (a, Maybe (Drevo a))
```

ki vrne največji element dane kopice ter kakršnokoli kopico, ki vsebuje vse preostale sinove. Če iz drevesa odstranimo zadnji element, naj bo druga komponenta vrnjenega para enaka `Nothing`.

~~d) (10 točk)~~ S pomočjo prejšnje funkcije napišite funkcijo `padajociElementi`, ki vrne seznam vseh elementov kopice, urejen od največjega proti najmanjšemu.

## 2. naloga (Zlaganje kock, 20 točk)

Nalogo lahko rešujete v Haskellu ali v Pythonu.

~~a) (10 točk + 5 točk)~~ Na voljo imamo poljubno število kock velikosti  $1 \times 1 \times 1$ ,  $2 \times 2 \times 2$  in  $3 \times 3 \times 3$ . Na koliko različnih načinov lahko sestavimo stolp višine  $n$ ? Pri tem lahko postavimo tudi večjo kocko na manjšo. Kock istih velikosti ne ločimo med seboj. Sestavite funkcijo, ki izračuna število vseh različnih stolpov višine  $n$ , ki jih lahko sestavimo iz danih kock. Na primer:

```
>>> st_stolpov(3)
4
```

Načine, na katere lahko sestavimo stolp višine 3, lahko predstavimo z naslednjimi nabori:

(3)    (2, 1)    (1, 2)    (1, 1, 1)

Funkcija naj ima časovno zahtevnost  $O(n)$ . Za rešitev s časovno zahtevnostjo  $O(\log n)$  dobite še dodatnih 5 točk.

~~b) (10 točk)~~ Na voljo imamo kocke velikosti  $1 \times 1 \times 1$  in  $3 \times 3 \times 3$  v rdeči barvi ter kocke velikosti  $1 \times 1 \times 1$  in  $2 \times 2 \times 2$  v modri barvi. Stolp mora biti zgrajen tako, da se nobeni dve kocki iste barve ne dotikata. Na koliko načinov lahko to naredimo? Primer:

```
>>> barvni_stolpi(3)
5
```

Načine, na katere lahko sestavimo barvni stolp višine 3, lahko predstavimo z naslednjimi nabori:

(3)    (1, 2)    (1, 1, 1)    (2, 1)    (1, 1, 1)

Funkcija naj ima časovno zahtevnost  $O(n)$ .

## ~~3. naloga (Neurejen seznam, 20 točk)~~

Nalogo lahko rešujete v Haskellu ali v Pythonu.

Dan je *neurejen* seznam dolžine  $n$ . Radi bi poiskali vse elemente v seznamu od vključno  $k$ -tega najmanjšega do vključno  $(k + m - 1)$ -tega najmanjšega. Pri tem seveda velja  $1 \leq k \leq k + m - 1 \leq n$ . Napišite funkcijo, ki kot argumente dobi seznam ter števili  $k$  in  $m$ . Funkcija naj vrne seznam z iskanimi elementi. Primer:

```
>>> l = [2, 7, 1, 9, 2, 3, 2, 5]
>>> poisci(l, 3, 4)
[2, 2, 3, 5]
>>> poisci(l, 5, 2)
[3, 5]
```

Funkcija naj ima časovno zahtevnost  $O(n \log m)$ .