

Programiranje I: 3. izpit

29. avgust 2017

Čas reševanja je 150 minut. **Funkcij v Haskellu ne pozabite opremiti z ustrezno signaturo.** Veliko uspeha!

1. naloga (Matrike, 20 točk)

Matriko v Haskellu predstavimo kot seznam seznamov.

```
type Matrika a = [[a]]
x = [[1, 2], [3, 4]] :: Matrika Int
y = [[1, 3], [2, 4]] :: Matrika Int
z = [[0, 1], [-2, 1]] :: Matrika Int
```

a) Napravite podatkovni tip `Matrika a` za element razreda `Num`. Pri tem naj operatorja `+` in `*` ustrezata običajnemu seštevanju in množenju matrik.

b) Definirajte funkcijo `zmnozi`, ki sprejme seznam matrik in vrne njihov produkt.

```
ghci> zmnozi [x,y,z]
[[-22, 16], [-50, 36]]
```

2. naloga (Seznam množic, 20 točk)

Množice števil v Haskellu predstavimo s seznamom.

```
type Mnozica = [Int]
a = [1, 2, 3] :: Mnozica
b = [2, 3, 4] :: Mnozica
c = [1..10] :: Mnozica
```

a) Sestavite funkcijo `presek`, ki sprejme neprazen seznam množic in vrne njihov presek.

```
ghci> presek [a, []]
[]
ghci> presek [a, b, c]
[2, 3]
```

b) Sestavite funkcijo `sodokrat`, ki sprejme seznam množic in vrne množico, ki vsebuje natanko tiste elemente, ki se pojavijo v sodo mnogo množicah vhodnega seznama.

```
ghci> sodokrat [a, b, c]
[]
ghci> sodokrat [a, b]
[2, 3]
ghci> sodokrat [b, c]
[2, 3, 4]
ghci> sodokrat [a, a]
[1, 2, 3]
ghci> sodokrat [c, c, c]
[]
```

3. naloga (Bombaž, 20 točk)

Vzdolž Evfrata je n bombažnih polj. Z i -tega polja pridobimo w_i kilogramov bombaža. Ves ta bombaž želimo porazdeliti po natanko k poljih, od koder bo odpeljan dalje. To porazdelitev napravimo v skladu z naslednjimi pravili:

1. Bombaž lahko premaknemo med katerima koli poljema.
2. Pri prenašanju bombaža z enega polja na drugo moramo prenesti ves bombaž, ki je na voljo na prvem polju.
3. Prenos w kilogramov bombaža s polja na lokaciji x_i na polje na lokaciji x_j nas stane $w \cdot |x_i - x_j|$ evrov.

Sestavite funkcijo `bombaz`, ki sprejme seznam lokacij bombažnih polj $[x_i]_{i=1}^n$, seznam kilogramov bombaža na vsakem od polj $[w_i]_{i=1}^n$ in število k ter učinkovito izračuna najmanjšo ceno, ki je potrebna, da ves bombaž porazdelimo po natanko k poljih.

```
>>> bombaz([20, 30, 40], [1, 1, 1], 1)
20
>>> bombaz([10, 12, 16, 18, 30, 32], [15, 17, 18, 13, 10, 1], 2)
182
```

4. naloga (Pogost element, 20 točk)

Seznam objektov ima *pogost element*, če je več kot pol njegovih elementov enakih. Elementi seznama med sabo niso nujno primerljivi z operatorjema `<` in `>`, lahko pa jih v konstantnem času primerjamo z operatorjem `=`.

a) Sestavite funkcijo `pogost`, ki kot vhod sprejme seznam objektov in vrne njegov pogost element, če ta obstaja. V nasprotnem primeru naj funkcija vrne `None`. Funkcija naj deluje v času $O(n \log n)$, pri čemer je n dolžina vhodnega seznama.

```
>>> pogost([1, 2, 3])
None
>>> pogost([1, 1, 2, 1, 3, 4])
None
>>> pogost([1, 1, 2, 1, 3, 1])
1
```

b) Ali obstaja algoritem, ki opravi enako kot funkcija v prejšnji točki, deluje pa v linearnem času?