

Programiranje I: 4. izpit

13. september 2016

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

1. naloga (Frizerka, 20 točk)

Nalogo lahko rešujete v Haskellu ali v Pythonu.

K edini frizerki v Dolenji vasi hodi na striženje vseh n vaških babnic. Prva pride ob času t_1 , naslednja ob času t_2 itd. Velja $t_1 \leq t_2 \leq \dots \leq t_n$. Frizerski salon se zapre točno ob času Z . (Seveda je $t_n \leq Z$.) Frizerka začne s striženjem takoj, ko je mogoče (tj. če je ob času prihoda i -te babnice še vedno striže $(i-1)$ -to, mora i -ta malo počakati, sicer pa nemudoma prične s striženjem). Frizerka ve, da mora vsem posvetiti enako količino časa, saj bi sicer lahko prišlo do kakšnih zamer. Po drugi strani pa bi frizerka rada vsako stranko strigla čim dlje časa, saj bo tako izvedela več novih tračev. Koliko je največja količina časa T_{\max} , ki ga lahko nameni posamezni stranki, da bo končala najkasneje ob času Z ?

Napišite funkcijo, ki kot argumenta dobi seznam $[t_1, t_2, \dots, t_n]$ in število Z ter izračuna in vrne čas T_{\max} . Funkcija naj vrne rezultat do natančnosti 10^{-6} . Primer:

```
>>> strizenje([0.1, 2.5, 6.2, 8.0, 10.0, 13.0], 16.0)
2.45
```

Komentar: Prvo začne striči takoj in konča ob času 2.55. Druga mora malce počakati in je končana ob času 5.0. Tretjo začne striči takoj ob njenem prihodu in konča ob času 8.65, zato mora četrta malo počakati itd. Zadnjo začne striči ob času 13.55 in tako konča točno ob času $Z = 16$.

Namig: Najprej napišite funkcijo, ki poleg seznama $[t_1, t_2, \dots, t_n]$ in števila Z kot argument prejme še število T . Funkcija naj vrne True natanko tedaj, ko lahko frizerka konča pravočasno, če za vsako babnico porabi T časa. Nato uporabite bisekcijo.

2. naloga (Pisarji, 20 točk)

Nalogo lahko rešujete v Haskellu ali v Pythonu.

Imamo n starih knjig. Prva knjiga ima p_1 strani, druga p_2 strani in tako dalje, zadnja pa ima p_n strani. Te knjige je potrebno prepisati na roke, zato jih bomo razdelili med k ($1 \leq k \leq n$) pisarjev. Pri tem zahtevamo, da mora prvi pisar prepisati prvih nekaj knjig, drugi naslednjih nekaj knjig itd. (Ne smemo torej dati prve in tretje knjige enemu od pisarjev, če mu ne damo hkrati tudi druge knjige.) Čas, ki ga porabi nek pisar, da opravi delo, je sorazmeren skupnemu številu strani, ki jih mora prepisati. Vsi pisarji so enako hitri. Ker bi radi, da čim prej končajo, želimo knjige razdeliti na optimalen način, tj. radi bi, da bi bilo skupno število strani, ki jih mora prepisati najbolj obremenjeni pisar, čim manjše.

Napišite funkcijo, ki kot argumenta prejme število pisarjev k in seznam $p = [p_1, p_2, \dots, p_n]$ ter vrne skupno število strani, ki jih bo moral prepisati najbolj obremenjeni pisar (če seveda knjige razdelimo optimalno). Primer:

```
>>> pisarji(3, [10, 20, 50, 130, 120, 70, 20])
210
>>> pisarji(4, [10, 20, 50, 130, 120, 70, 20])
130
```

Namig: Naj bo $f(i, j)$ skupno število strani pri najbolj obremenjenem med pisarji $1, \dots, j$, če knjige $1, 2, \dots, i$ razdelimo med pisarje $1, \dots, j$ na optimalni način. Ali znate učinkovito izračunati $f(n, k)$?

3. naloga (Redki polinomi, 40 točk)

Polinom ene spremenljivke je redek, če je večina njegovih koeficientov enakih 0. Tak polinom namesto s seznamom koeficientov rajši predstavimo s tipom `Fractional t => [(Int, t)]`, tj. s seznamom parov, kjer so prve komponente nenegativna cela števila (eksponenti), druge komponente pa ustrezni koeficienti. Elemente seznama uredimo naraščajoče glede na eksponente, ničelnih koeficientov pa seveda ne hranimo. Polinom $x^{341} - 2x^{82} + 35$ bi tako predstavili s seznamom `[(0,35), (82,-2), (341,1)]`.

a) (10 točk) Sestavite funkcijo `vrednost`, ki izračuna vrednost polinoma v podani točki. Funkcija naj ima časovno zahtevnost $O(n)$, kjer je n dolžina seznama, s katerim polinom predstavimo. Zgled:

```
Prelude> let p = [(0,35), (82,-2), (341,1)]
Prelude> vrednost p 1
34
Prelude> vrednost p (-0.999)
32.44659407363115
```

b) (10 točk) Sestavite funkcijo `vsota`, ki sešteje dva redka polinoma. Rezultat naj bo nov redek polinom. Funkcija naj ima časovno zahtevnost $O(m+n)$, kjer sta m in n dolžini seznamov, ki predstavljata polinoma. Zgled:

```
Prelude> let p = [(0,35), (82,-2), (341,1)]
Prelude> q = [(5,17), (82,2), (341,3)]
Prelude> vsota p q
[(0,35), (5,17), (341,4)]
Prelude> vsota [(0,2), (1,1)] [(0,-2), (1,-1)]
[]
```

c) (10 točk) Sestavite funkcijo `produkt`, ki zmnoži redka polinoma. Rezultat naj bo nov redek polinom. Zgled:

```
Prelude> let p = [(0,35), (82,-2), (341,1)]
Prelude> let q = [(5,-4), (18,3)]
Prelude> produkt p q
[(5,-140), (18,105), (87,8), (100,-6), (346,-4), (359,3)]
Prelude> produkt [(0,2), (1,1)] [(0,-2), (1,-1)]
[(0,-4), (1,-4), (2,-1)]
```

d) (10 točk) Sestavite funkcijo `koeficienti`, ki vrne seznam koeficientov polinoma in sicer po vrsti od prostega do vodilnega člena, skupaj z vsemi ničlami, ki so vmes med neničelnimi členi. Funkcija naj ima časovno zahtevnost $O(n)$, kjer je n dolžina izhodnega seznama. Zgled:

```
Prelude> koeficienti [(1,1), (4,-3)]
[0,1,0,0,-3]
Prelude> koeficienti []
[]
```