

Programiranje I: 1. izpit

2. februar 2016

Čas reševanja je 150 minut. Doseženih 80 točk šteje za maksimalno oceno. **Funkcij v Haskellu ne pozabite opremiti z ustrezno signaturo.** Veliko uspeha!

~~1. naloga (Funkcije, 30 točk)~~

~~a) (10 točk)~~ V Haskellu sestavite funkcijo `naloga1a`, ki na danem elementu zaporedoma uporabi vse funkcije iz danega seznama. Na primer:

```
naloga1a [negate, (+ 1), subtract 3, (2 -), (* 2)] 5
```

vrne 18, saj je $(2 - (((-5) + 1) - 3)) \cdot 2 = 18$.

~~b) (10 točk)~~ Sestavite funkcijo `naloga1b`, ki deluje podobno kot prejšnja, le da funkcijo uporabi na elementu le tedaj, kadar bi se s tem povečala vrednost. Na primer:

```
naloga1b [negate, (+ 1), subtract 3, (2 -), (* 2)] 5
```

vrne 12, saj vrednost povečata le funkciji `(+ 1)` in `(* 2)`. Pozor, pri različnih elementih vrednost povečajo različne funkcije. Na primer:

```
naloga1b [negate, (+ 1), subtract 3, (2 -), (* 2)] (-3)
```

vrne 8, saj vrednost tokrat povečajo funkcije `negate`, `(+ 1)` in `(* 2)`.

~~c) (10 točk)~~ Sestavite še funkcijo `naloga1c`, ki deluje podobno kot prejšnji, le da uporabi tiste funkcije, s katerimi doseže največji možen rezultat. Na primer:

```
naloga1c [negate, (+ 1), subtract 3, (2 -), (* 2)] 5
```

vrne 20, saj največji možen rezultat dosežemo kot $(2 - (((-5) - 3)) \cdot 2 = 20$.

2. naloga (Drobencljava podzaporedja, 20 + 10 točk)

Pravimo, da je zaporedje *drobencljavo*, če se vsaka dva zaporedna člena med seboj po absolutni vrednosti razlikujeta za natanko 1. Na primer, zaporedje 1,2,3,2 je drobencljavo, zaporedje 1,2,3,1 pa ne, ker se tretji in četrti člen razlikujeta za 2.

~~a) (10 točk)~~ Sestavite funkcijo `naloga2a`, ki učinkovito izračuna število vseh nepraznih drobencljavih podzaporedij danega zaporedja. Na primer, v zaporedju 1,2,3,1 je 9 takih podzaporedij:

1; 2; 3; 1; 1,2; 2,3; 2,1; 1,2,1; 1,2,3.

Bodite pozorni na to, da se lahko kakšno podzaporedje v danem zaporedju pojavi tudi večkrat, ter na to, da niso vsa podzaporedja nujno strnjena. V komentarju kode tudi utemeljite, kakšna je časovna zahtevnost vaše rešitve.

~~b) (10 točk)~~ Sestavite funkcijo `naloga2b`, ki učinkovito izračuna množico vseh *različnih* drobencljavih podzaporedij danega zaporedja. V komentarju kode tudi utemeljite, kakšna je časovna zahtevnost vaše rešitve.

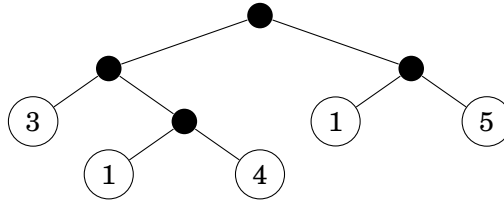
~~c) (+10 točk)~~ Sestavite funkcijo `naloga2c`, ki izračuna dolžino najdaljšega *strnjenega* drobencljavega podzaporedja, ki se začne in konča z istim številom. Vaša rešitev naj deluje v času $O(n)$, kjer je n dolžina zaporedja.

3. naloga (Drevesa, 30 + 10 točk)

Neprazna dvojiška drevesa, v katerih so vrednosti spravljene zgolj v listih, lahko v Haskellu predstavimo s podatkovnim tipom:

```
data Drevo a = List a
             | Razvejano (Drevo a) (Drevo a)
```

Na primer, drevo

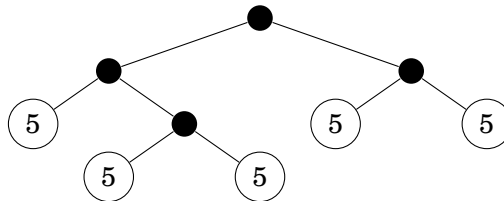


bi predstavili kot

```
Razvejano
  (Razvejano (List 3) (Razvejano (List 1) (List 4)))
  (Razvejano (List 1) (List 5))
```

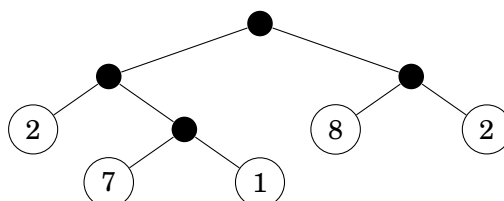
~~a) (10 točk)~~ Sestavite funkcijo naloga3a, ki vrne seznam vseh vrednosti, ki se pojavljajo v listih drevesa. Vrednosti naj bodo našteje od leve proti desni, kakor si sledijo v drevesu. Na primer, za zgornje drevo bi funkcija vrnila seznam [3,1,4,1,5].

~~b) (10 + 10 točk)~~ Sestavite funkcijo naloga3b, ki vrne drevo z enako obliko kot dano drevo, le da so vse vrednosti v listih zamenjane z največjo vrednostjo. Na primer, za zgornje drevo bi funkcija vrnila drevo:



Za dodatnih 10 točk funkcijo napišite tako, da naredi *zgolj en* obhod po drevesu.

~~c) (10 točk)~~ Sestavite funkcijo naloga3c, ki vrne drevo z enako obliko kot dano drevo, le da so vrednosti v listih zaporedoma zamenjane s tistimi v danem seznamu. Na primer, za zgornje drevo in seznam [2,7,1,8,2] bi funkcija vrnila drevo:



Če se število listov v drevesu ne ujema z dolžino seznama, naj funkcija javi napako.

Namig: Drevo a -> [b] -> (Drevo b, [b])