

Programiranje I: 3. izpit

28. avgust 2018

Čas reševanja je 150 minut. Veliko uspeha!

1. naloga

a) Napišite funkcijo

```
razlika_kvadratov : int -> int -> int,
```

ki za dani števili izračuna razliko med kvadratom vsote in vsoto kvadratov.

b) Napišite funkcijo

```
uporabi_na_paru : ('a -> 'b) -> 'a * 'a -> 'b * 'b,
```

ki podano funkcijo uporabi na vsakem elementu para.

c) Napišite funkcijo

```
ponovi_seznam : int -> 'a list -> 'a list,
```

za katero ponovi_seznam n sez vrne seznam sestavljen iz n ponovitev podanega seznama. Če je n manjši ali enak 0, naj funkcija vrne prazen seznam.

d) Napišite funkcijo

```
razdeli : int list -> int list * int list,
```

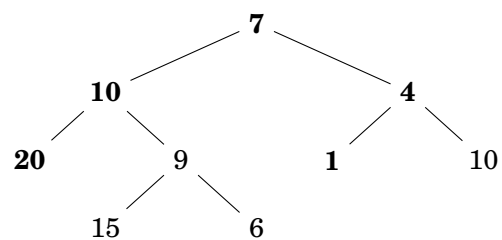
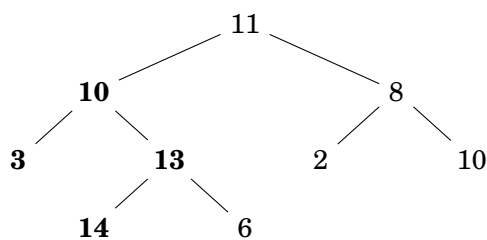
za katero razdeli sez vrne par seznamov, kjer levi seznam v paru vsebuje vse negativne elemente sez, desni pa preostale. Za vse točke naj bo funkcija *repno-rekurzivna*.

2. naloga

Napišite funkcijo

```
monotona_pot : int drevo -> int list,
```

ki v danem dvojiškem drevesu poišče najdaljšo pot, v kateri vrednosti v vozliščih bodisi naraščajo bodisi padajo. Na primer, v spodnjem levem drevesu je to pot 3 – 10 – 13 – 14, v desnem pa 20 – 10 – 7 – 4 – 1:



3. naloga

Konstruiramo podatkovni tip verige filtrov:

```
type 'a veriga = | Filter of ('a -> bool) * 'a list * 'a veriga
                | Ostalo of 'a list,
```

kjer `Filter(f, sez, rep)` predstavlja člen v verigi, ki v seznamu `sez` hrani elemente, za katere funkcija `f` vrne `true`, parameter `rep` pa hrani preostanek verige.

a) Napišite prazen filter (torej takšen, ki še ne hrani nobenega elementa) `test : int veriga`, ki v prvem členu hrani negativna števila, v drugem členu hrani števila manjša od 10 in v tretjem členu preostala števila.

$$[\cdot]_{<0} \sim [\cdot]_{<10} \sim [\cdot]$$

b) Napišite funkcijo `vstavi : 'a -> 'a veriga -> 'a veriga`, ki vrne verigo, v kateri je dani element vstavljen na začetek seznama tistega člena verige, ki prvi zadošča filtracijski funkciji. Pri vstavljanju števil `-5, 7, 100, -7` in `2` v zgornjo verigo bi dobili:

$$[-5, -7]_{<0} \sim [7, 2]_{<10} \sim [100]$$

c) Napišite funkcijo `poisci : 'a -> 'a veriga -> bool`, ki preveri, ali je element vsebovan v katerem od seznamov v verigi filtrov. Pri tem naj preveri zgolj en seznam.

d) Napišite funkcijo `izprazni_filtre : 'a veriga -> 'a veriga * 'a list`, ki kot par vrne verigo filtrov, kjer imajo vsi členi prazne sezname, in seznam vseh elementov, ki so bili hranjeni v filtrih. Na primer, za zgornjo verigo bi poleg prazne verige funkcija vrnila še seznam `[-5, -7, 7, 2, 100]`.

e) Napišite funkcijo `dodaj_filter : ('a -> bool) -> 'a veriga -> 'a veriga`, ki na začetek podane verige filtrov doda nov filter s filtracijsko funkcijo `f`. Elementi v vrnjeni verigi filtrov naj bodo pravilno razporejeni tudi glede na novo dodani člen. Na primer, če bi zgornji verigi dodali še filtracijsko funkcijo, ki preverja sodost, bi dobili:

$$[2, 100]_{\text{je sod}} \sim [-5, -7]_{<0} \sim [7]_{<10} \sim [\cdot]$$