

## Programiranje I: 1. izpit

24. januar 2018

Čas reševanja je 150 minut. Veliko uspeha!

### 1. naloga

**a)** Definirajte funkcijo `izpisi_vsa_stevila`, ki sprejme seznam celih števil in zaporedoma izpiše vse njegove elemente.

```
# izpisi_vsa_stevila [1; 2; 3; 4];;  
1234- : unit = ()  
# izpisi_vsa_stevila [];;  
- : unit = ()
```

**b)** Definirajte funkcijo `map2_opt`, ki sprejme funkcijo dveh argumentov in dva seznama. Vrne naj seznam rezultatov, ko dano funkcijo zaporedoma uporabimo na istoležnih elementih seznamov. Funkcija naj vrne rezultat le, kadar sta oba vhodna seznama enake dolžine, zato uporabite tip `option`.

```
# map2_opt (+) [1; 2; 3] [7; 5; 3];;  
- : int list option = Some [8; 7; 6]  
  
# map2_opt (+) [1; 2; 3] [7; 5];;  
- : int list option = None
```

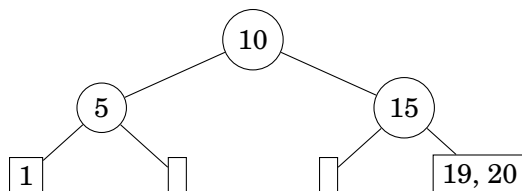
Za maksimalno število točk naj bo funkcija **repno rekurzivna**.

## 2. naloga

Filtracijsko drevo ima dve vrsti osnovnih gradnikov:

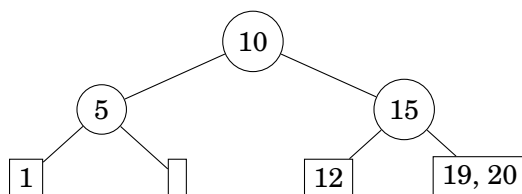
- Vozlišča imajo celoštevilsko vrednost, levo poddrevo in desno poddrevo.
- Listi oz. škatle vsebujejo seznam celoštevilskih vrednosti.

a) Definirajte tip filtracijskih dreves `filter_tree` ter drevo primer : `filter_tree`, ki predstavlja sledeče filtracijsko drevo:



b) Filtracijsko drevo razvršča števila v škatle glede na njihovo vrednost. Vozlišče z vrednostjo  $k$  razvrsti število  $n$  v levo poddrevo, če velja  $n \leq k$ , oz. v desno poddrevo, če velja  $n > k$ . Ko število doseže škatlo, ga dodamo v seznam števil v škatli. Škatle lahko vsebujejo ponovitve in niso nujno urejene.

Napišite funkcijo `vstavi`, ki sprejme število in filtracijsko drevo in vrne filtracijsko drevo z vstavljenim številom. Na primer, `vstavi 12 primer` vrne drevo:



c) Napišite funkcijo `vstavi_seznam`, ki sprejme seznam celih števil in filtracijsko drevo ter vrne filtracijsko drevo z vstavljenimi elementi seznama. Vrstni red vstavljanja ni pomemben.

d) Definirajte funkcijo, ki sprejme filtracijsko drevo in preveri, ali so vsa števila v pravih škatlah glede na način razvrščanja. Na primer, v levem drevesu so števila razvrščena pravilno, v desnem pa ne:



### 3. naloga

Linearne preslikave na  $\mathbb{Z}^2$  lahko predstavimo tako z matrikami kot s funkcijami, obe predstavitvi pa zadoščata signaturi Linearna, ki določa:

- type t — tip linearnih preslikav;
- val id : t — identiteta;
- val uporabi : t -> vektor -> vektor — dano preslikavo uporabi na vektorju;
- val iz\_matrike : matrika -> t — vrne linearno preslikavo, določeno z matriko;
- val iz\_funkcije : (vektor -> vektor) -> t — vrne linearno preslikavo, določeno s funkcijo (predpostavite lahko, da je funkcija linearna);
- val kompozitum : t -> t -> t — vrne kompozitum danih preslikav.

Pri tem vektorje in matrike predstavimo s pari oz. četvorci celih števil:

$$\begin{bmatrix} x \\ y \end{bmatrix} = (x, y) \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} = (a, b, c, d)$$

Tako definiramo tipa

```
type vektor = int * int
type matrika = int * int * int * int
```

**a)** Napišite modul Matrika : Linearna, ki linearne preslikave predstavi z matrikami tipa matrika.

**b)** Napišite modul Funkcija : Linearna, ki linearne preslikave predstavi s funkcijami tipa vektor -> vektor.

### 4. naloga (Nalogo lahko rešite v OCamlu ali Pythonu)

Lisjaček krađe jabolka v sadovnjaku, kjer so jablane razporejene v vrstah. Okrađe lahko zgolj  $N$  jablan, preden ga prežene lastnik sadovnjaka.

Ker je lisjaček še mlad in ne vešče krađe jabolka, v vsaki vrsti obira jablane eno za drugo in pri tem nobene ne preskoči. Ko se odloči, da se bo lotil naslednje vrste, se ne vrne več nazaj. V primeru, da se znajde na koncu zadnje vrste, svojo pustolovščino zaključi, ne glede na to, koliko časa mu je še preostalo.

Sadovnjak predstavimo z matriko, kjer vrstice predstavljajo vrste sadovnjaka, vrednosti pa števila jabolka na jablanah. Pri tej predstavitvi se lisjaček v vsakem koraku premakne bodisi v desno bodisi na začetek naslednje vrstice.

**a)** Napišite algoritem, ki **učinkovito** izračuna največje število jabolka, ki jih lisjaček lahko ukrade. Na primer, če je  $N = 6$  in je sadovnjak oblike

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ 3 & 2 & 0 & 5 \\ 8 & 0 & 7 & 2 \end{bmatrix},$$

lahko lisjaček ukrade največ 24 jabolka, kar stori tako, da izbere sledeče jablane:

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ 3 & 2 & 0 & 5 \\ 8 & 0 & 7 & 2 \end{bmatrix}$$

**b)** V komentarju napišite in utemeljite časovno zahtevnost vašega algoritma.