

Programiranje I: 2. izpit

7. februar 2022

Čas reševanja je 120 minut. Veliko uspeha!

1. naloga

a) Napišite predikat `sta_pravokotna : int * int * int -> int * int * int -> bool`, ki pove, ali sta dva vektorja pravokotna. Vektorje predstavimo s trojicami celih števil.

b) Napišite funkcijo `postkompozicija : ('a -> 'b) -> ('b -> 'c) -> 'a -> 'c`, ki sprejme dve funkciji in vrne njun kompozitum.

```
# let deli_z m n = n / m;;
# postkompozicija ((+) 82) (deli_z 82) 3362;;
- : int = 42
# postkompozicija (deli_z 82) ((+) 82) 3362;;
- : int = 123
```

c) Napišite funkcijo `dopolni : 'a -> 'a option list -> 'a list`, ki sprejme privzeto vrednost in seznam morebitnih vrednosti ter vrne seznam, v katerem namesto manjkajočih vrednosti nastopa privzeta, obstoječe vrednosti pa ostanejo nespremenjene. Za vse točke naj bo funkcija repno rekurzivna.

```
# dopolni 3 [Some 1; None; Some 2];;
- : int list = [1; 3; 2]
```

d) Naravno število v drugačni bazi predstavimo s seznamom celih števil, kjer so na zadnjem mestu enice, na predzadnjem "desetice" in tako naprej.

Napišite funkcijo `pretvori : int -> int list -> int`, ki sprejme seznam pozitivnih celih števil in bazo ter vrne podano število v desetiškem sistemu kot celo število.

```
# pretvori 8 [6; 4; 4; 2];; (* 6*8^3 + 4*8^2 + 4*8 + 2 *)
- : int = 3362
```

2. naloga

V datotečnih sistemih, sistemih za nadzor različic in kriptovalutah za učinkovito zagotavljanje pristnosti podatkov uporabljamo *Merkleova drevesa*. To so običajna dvojiška drevesa, le da vsako vozlišče poleg podatka in otrok hrani še svojo *zgostitev* (angleško *hash*), ki je celo število, ki tvori nekakšen podpis celotne vsebine drevesa.

```
type 'a merkle = List | Vozlisce of 'a vozlisce
and 'a vozlisce = {
  levo : 'a merkle;
  podatek : 'a;
  desno : 'a merkle;
  zgostitev : int;
}
```

Zgostitev drevesa izračunamo z zgostitvenimi funkcijami, ki sprejmejo zgostitev levega otroka, podatek v vozlišču in zgostitev desnega otroka:

```
type 'a zgostitev = int -> 'a -> int -> int
```

Za zgostitev praznega drevesa izberemo število 0.

Na primer, za zgostitveno funkcijo:

```
let primer_h l p d = ((l * 3) + (p * 5) + (d * 7)) mod 11
```

je spodnje drevo veljavno Merkleovo drevo:

```
let drevo : int merkle = Vozlisce {
  levo = Vozlisce {
    levo = Vozlisce { levo = List; podatek = 10; desno = List; zgostitev = 6 };
    podatek = 14;
    desno = Vozlisce { levo = List; podatek = 474; desno = List; zgostitev = 5 };
    zgostitev = 2;
  };
  podatek = 57;
  desno = Vozlisce {
    levo = List;
    podatek = 12;
    desno = Vozlisce { levo = List; podatek = 513; desno = List; zgostitev = 2 };
    zgostitev = 8;
  };
  zgostitev = 6;
}
```

a) Napišite predikat `preveri : 'a zgostitev -> 'a merkle -> bool`, ki sprejme zgostitveno funkcijo in drevo ter preveri, ali so vse zgostitve v drevesu pravilno izračunane.

b) Napišite funkcijo `prestej_napacne : 'a zgostitev -> 'a merkle -> int`, ki vzame zgostitveno funkcijo in drevo ter vrne število napačno poračunanih zgostitev v drevesu. Zgostitev vozlišča je napačna, če ne ustreza definiciji računanja zgostitve za vozlišče, ne glede na to, ali sta zgostitvi otrok pravilni ali ne.

c) Napišite funkcijo `popravi : 'a zgostitev -> 'a merkle -> 'a merkle`, ki sprejme (potencialno napačno) Merkleovo drevo in vrne drevo iste oblike z istimi podatki, le da so vse zgostitve v njem pravilno izračunane.

3. naloga

France 9. februarja zjutraj prazni dolgo mizo, za katero je prejšnji večer praznovalo N gostov (vsi v eni sami vrsti). Vsak izmed njih je za sabo pustil nekaj (nenegativnih) enot hrane in pijače. Ker je Francetu med pospravljanjem dolgčas, se je odločil, da se bo namesto pospravljanja raje igral igrico.

Pospravljanje začne skrajno levo s praznim vrčem in se premika v desno. Ko pride na posamezno mesto, si vso pijačo, ki je ostala tam, pretoči v svoj kozarec in se vrne nazaj (v levo) za toliko mest, kot je gost pustil hrane. Ob vrnitvi na desno si ne natoči dodatne pijače, niti ne ponovi vračanja. Če bi se slučajno moral premaknit preko levega roba mize, potem se bo ob levem robu mize tudi ustavil. Ker je od prejšnjega dne še hudo utrujen, ob vsakem premiku (tako naprej kot nazaj) polije toliko enot pijače kot je le-ta dolg. Pri tem mora paziti, da ima v kozarcu vedno vsaj nekaj pijače. Dovoljeno je na primer, da s 3 enotami v kozarcu naredi premik dolg 3 enote, 4 pa je že preveč. Pri tem mora upoštevati tudi možne premike zaradi hrane in dodajanje pijače na posameznem mestu. S štirimi enotami se lahko premakne na mesto ki ima dve enoti hrane in je oddaljeno tri mesta, le, če ima to mest vsaj eno enoto pijače (drugače bi ob premiku nazaj ostal s praznim kozarcem). Zanima ga najmanj koliko mest bo obiskal, preden mu bo uspelo priti čez desni konec mize (z nenegativno količino pijače)? Pri tem ga budno opazuje celotna zbrana družba in skrbi, da se stanje na mizi ne spremeni (če bi dvakrat obiskal isto mesto, se je pijača tam vrnila na enako stanje, kot je bila na začetku).

Napišite funkcijo pospravi, ki sprejme seznam parov celih števil. Prva komponenta para pove količino hrane, druga pa pijače na posameznem mestu in vrne celo število, najmanjše število premikov, ki jih mora France opraviti, da se prebije do konca mize. Vaša rešitev lahko predpostavi, da bo to vedno mogoče v končno mnogo korakih.

Na spodnji mizi je najbolje, da najboljša pot, ki začne na indeksu 0 (si natoči dve enoti pijače), od koder se premakne na indeks 2 (kjer ima prazen kozarec) in se vrne nazaj na začetek, kjer ima sedaj 3 enote pijače. V drugo se spet premakne na indeks 2, in vrne nazaj na začetek s 4 enotami pijače. V tretje pa se premakne na indeks 1, in se vrne nazaj na začetek s 5 enotami pijače, s čimer ima dovolj zaloge da lahko s četrtem premikom doseže konec mize. (Alternativno bi lahko v tretjem premiku šel na indeks 2 in še vedno v enakem številu premikov dosegel konec mize).