Algoritmi in podatkovne strukture – 2
Pisni izpit 15. svečan 2013 (2011/12)

Pisni izpit morate pisati posamič. Pri reševanju je literatura dovoljena. Pri odgovarjanju bodi natančni in: (i) odgovarjajte *na zastavljena* vprašanja; in (ii) odgovorite na *vsa* zastavljena vprašanja.

Čas pisanja izpita je 90 minut.

Veliko uspeha!

NALOGA	TOČK	OD TOČK	NALOGA	TOČK	OD TOČK
1			3		
2			4		

IME IN PRIIMEK: ______________________

ŠTUDENTSKA ŠTEVILKA: ______________________

DATUM: ______________________

PODPIS: ______________________

1. naloga: Peter Zmeda je našel naslednji koščk programske kode:

```
void krneki (a[] int; b: int)
  c= 0;
  while c < b-1 do
    d= c+1;
    while d < b do
      if a[c] > a[d] then
        x= a[c]; a[c]= a[d]; a[d]= x
      endif
      d= d+1;
    enddo;
    c= c+1;
  enddo;
endfunction;
```

VPRAŠANJA:

1. Peter se ne upa uporabiti funkcije `krneki`, ker ne ve, kaj počne. Kaj počne v resnici funkcija?

   NAMIG: Pomagate si lahko s sledenjem izvajanja kode za majhne primere.

2. Po katerih štirih stvareh se sprašujemo ob vsakem algoritmu (funkciji)? Kakšni so vaši odgovori ob zgornji kodi na ta vprašanja – odgovore utemeljite.

3. Funkcija `krneki` je zapisana nerekurzivno. Zapišite jo v rekurzivni obliki.

2. naloga: Imamo množico elementov $\{1, ..., 10\}$, iz katerih tvorimo 10 disjunktnih množic s po enim elementom, pri čemer uporabimo učinkovito podatkovno strukturo s predavanj.

VPRAŠANJA:

1. Nad tako tvorjenimi množicami izvedemo naslednje operacije:

   `F 5, U 3 5, U 1 5, U 7 1, F 5`

   kjer `F x` vrne ime množice, kateri pripada element `x` in `U x y` naredi unijo množic, katerim pripadata elementa `x` in `y`.

   Za vsako od zgornjih operacij narišite, kako se spreminja podatkovna struktura ter preštejte in zapišite število opravljenih primerjav ob vsaki operaciji.

2. Zapišite psevdokodo operacije (funkcije) `F`.

   NAMIG: Najprej zapišite definicijo podatkovne strukture, ker sicer bo psevdokoda velika zmešnjava.

3. Recimo, da imamo opravka z množico elementov $\{1, ..., n\}$, iz katerih ponovno naredimo n disjunktnih nepraznih množic. Nad temi množicami izvedemo n operacij `F` in n operacij `U` v poljubnem vrstnem redu. Koliko primerjanj vsega skupaj bomo izvedli po zaključku vseh $2n$ operacij? Odgovor utemeljite.

   NAMIG: Za skoraj vse točke ne pričakujem povsem točnega odgovora. Razmišljajte o najboljšem in o najslabšem primeru.

3. naloga: Dinamično programiranje in rekurzivne podatkovne strukture. Na predavanjih smo večkrat srečali Fibonaccijeva števila, ki so bila definirana kot $F_n = F_{n-1} + F_{n-2}$, kjer je $F_0 = F_1 = 1$. Tokrat definirajmo zaporedje števil malce drugače in sicer kot

$$P_n = \left\lfloor \frac{n}{2} \right\rfloor + P_{\lfloor \frac{n}{2} \rfloor} + P_{\lceil \frac{n}{2} \rceil} \ , \tag{1}$$

kjer $P_o = P_1 = 0$.

VPRAŠANJA:

1. Koliko je P_{10}, kjer je P_n definiran v en. (1)? Prikažite izračun!

2. Zapišite algoritem, ki izračuna P_n za poljuben n, kjer je n parameter. Kakšna je časovna in kakšna prostorska zahtevnost vašega algoritma?

3. Imamo običajno dvojiško iskalno drevo definirano kot:

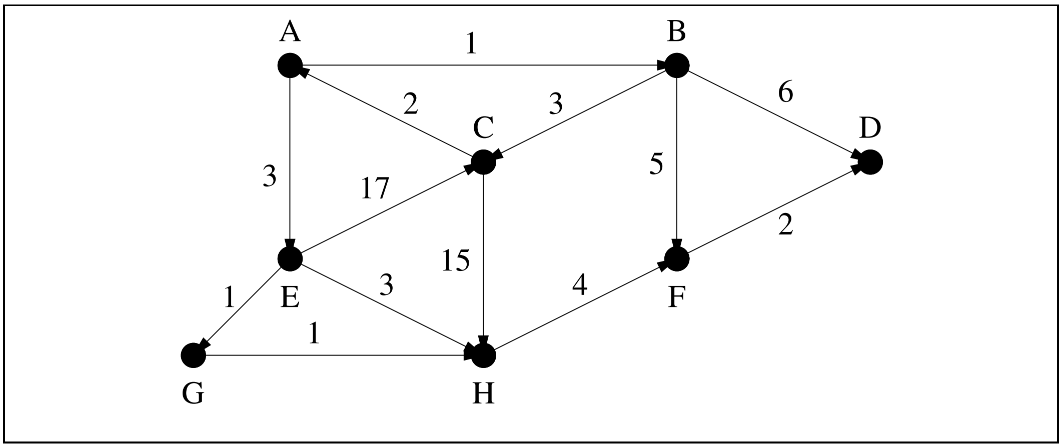
   ```
   public class Drevo {
     Elt   koren;
     Drevo levo, desno
   }
   ```

   Napišite algoritem, ki bo izpisal elemente v vozliščih drevesa na sledeči način:

   koren: n

   kjer je *koren* vrednost elementa in n število elementov v drevesu, ki so manjši njega.

   NAMIG: Uporabite vmesni (*inorder*) obhod z enim ali več dodatnimi parametri.



Slika 1: Primer usmerjenega grafa.

4. naloga: Imamo usmerjen graf na sl. 1. Poleg tega definirajmo podatkovno strukturo za predstavitev grafa $G(V, E)$ ($|V| = n$ in $|E| = m$) s seznami sosedov S, kjer je:

- $S[v].q$ število sosedov vozlišča v; in
- so $S[v].s[i]$ ($0 \leq i < S[v].q$) uteži povezav $(v, S[v].s[i])$.

VPRAŠANJA:

1. Za graf s sl. 1 zapišite vrednosti $S[C].q$, $S[D].q$ in $S[B].s[0]$. Utemeljite odgovora.
2. Na grafu s sl. 1 naredite obhod v širino pričenši v A in zapisujte posamezne korake oziroma vozlišča kot jih obiskujete.
3. Poiščite najkrajše poti iz vozlišča A do vseh ostalih vozlišč z uporabo Dijkstrovega algoritma. Zapisujte vozlišča, kot jih obiskujete pri posameznih korakih. Na koncu zapišite še dolžine najkrajših poti od A do vseh vozlišč.

   Kaj opazite, ko primerjate odgovora na to in na prejšnje vprašanje. Primerjavo dobro utemeljite.