

Algoritmi in podatkovne strukture – 2

Prvi kolokvij (2013/14)

Kolokvij 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 – če boste odgovarjali na vsa vprašanja, lahko dobite dodatne točke.

Čas pisanja izpita je 60 minut.

Veliko uspeha!

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

IME IN PRIIMEK:

ŠTUDENTSKA ŠTEVILKA:

DATUM:

PODPIS:

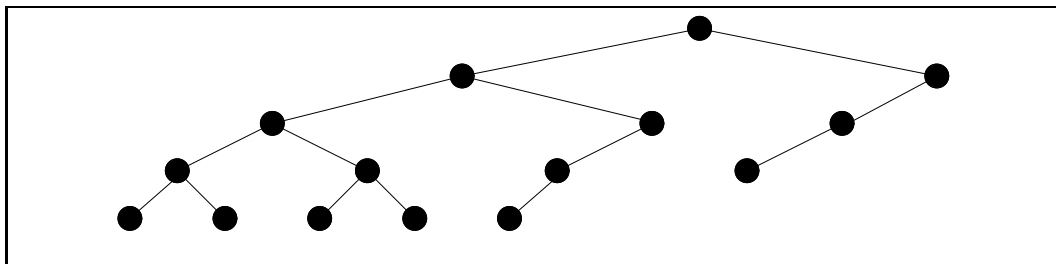
1. naloga: *Uvod in osnove.* Imamo naslednji program:

```
int array foo (int array a) {
    int n= a.length();
    for i= 1 to n-1
        for j= i+1 to n
            if a[i] > a[j] {
                x= a[i]; a[i]= a[j]; a[j] = x;
            }
    return a;
}
```

VPRAŠANJA:

1. Opisno utemeljite, kaj dela zgornji program.
2. Kakšna je invarianca pred začetkom notranje zanke? Utemeljite odgovor.
3. Kakšna je časovna zahtevnost programa? Dokažite odgovor.

2. naloga: Levo kanonično dvojiško drevo je drevo, ki je levo poravnano. To pomeni, da, če vozlišča v neki plasti pregledujemo z leve proti desni, velja: najprej ima prvih $p - 1$ vozlišč 2 naslednika; nato ima p -to vozlišče šteto z leve enega ali nobenega naslednika; ter preostala vozlišča nimajo naslednikov. Vozliščem brez naslednikov rečemo *list*. Drevo na sl. 1 ima na plasti 4 z leve dve vozlišči z dvema



Slika 1: Primer levega kanoniničnega drevesa.

naslednikoma, nato eno z enim naslednikom ($p = 3$) ter na koncu dve vozlišči brez naslednikov.

VPRAŠANJA:

1. Naj bo drevo s sl. 1 številsko drevo nad abecedo $\{0, 1\}$ in naj bodo v njegovih listih posamezni elementi. Zapišite ključne vse elementov.

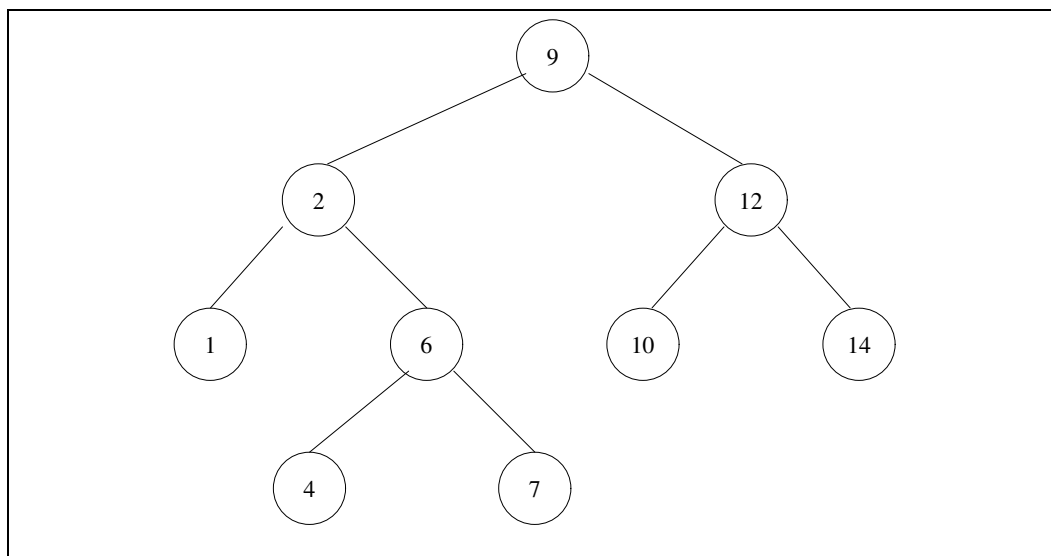
DODATNA: Ali kaj opazite, ko primerjate poljubne elemente? Utemeljite odgovor.

2. Stisnite drevo najprej po plasteh z $\alpha = 0,90$ in nato še po poteh (PATRICIA).
3. Vrnimo se k drevesu na sl. 1, ki ga bomo sedaj shranili v *implicitni podatkovni strukturi*. Opišite kako takšno drevo shranimo v implicitno podatkovno strukturo. Za polovico točk zapišite kako bi implicitno shranili drevo s sl. 1.

3. naloga:

VPRAŠANJA:

1. Narišite najmanjše AVL drevo višine 5 – to je AVL drevo višine 5, ki ima najmanj elementov.
2. V AVL drevo na sl. 2 vstavite element s ključem 5. Narišite drevo, po



Slika 2: Primer AVL drevesa.

vstavljanju in utemeljite korake.

3. Če vstavljamo ključ v naraščajočem vrstnem redu v AVL drevo, bo višina drevesa monotono naraščala. Dokažite ali je izjava pravilna ali ne.

4. naloga: Peter Zmeda se je odločil za malce drugačno rešitev Union-Find problema. Za množico n elementov je definiral polje oznak $id[0..n-1]$, v katerem je za vsak element i , zapisano ime predstavnika množice $id[i]$, kateri pripada i . Recimo, v primeru polja, kjer je $n = 10$

i	0	1	2	3	4	5	6	7	8	9
$id[i]$	1	1	2	2	6	2	6	9	9	9

pripada element 3 množici, katere predstavnik je 2 in prav tako elementa 2 ter 5. Element 4 pa po drugi strani pripada množici, katere predstavnik je 6.

VPRAŠANJA:

1. Napišite funkcijo `MakeSet()`, ki na začetku ustvari vse množice, v katerih je seveda po eden element.
2. Napišite funkcijo `SameSet(p, q)`, ki vrne `TRUE`, če sta p in q v isti množici, in `FALSE` sicer.
3. S protiprimerom (i) pokažite, da spodnja funkcija za unijo ne deluje pravilno in (ii) jo popravite.

```
void Union (p, q) {
    if SameSet(p, q) return;
    for i= 0 to n-1
        if id[i] == id[p] id[i] = id[q];
}
```