

Algoritmi i strukture podataka

Završni ispit

23. lipnja 2009.

Napomena za sve zadatke:

- Nije dopušteno korištenje naredbe **goto** te statičkih i globalnih varijabli.

Zadatak 1. (8 bodova)

U jednostruko povezanu nesortiranu listu spremljeni su podaci o studentima. Lista je zadana sljedećim strukturama:

```
struct zapis {
    char imeprezime[80+1];
    int ocjena;
};

struct at {
    struct zapis element;
    struct at *sljed;
};

typedef struct at atom;
```

Napišite funkciju koja će iz zadane liste sve čvorove koji sadrže studente čija je ocjena veća ili jednaka prosjeku ocjena svih studenata u listi prebaciti (bez stvaranja novih čvorova) u novu jednostruko povezanu listu. Funkcija treba vratiti pokazivač na glavu nove liste. Funkcija mora imati prototip:

```
atom * razdvojiListe(atom **glava);
```

Napomena: U izvornoj listi na koju pokazuje glava ostaju čvorovi koji sadrže studente čija je ocjena manja od prosjeka ocjena svih studenata.

Primjer: glava prije poziva funkcije pokazuje na listu čvorova koji sadrže sljedeće ocjene: 5, 2, 4, 3, 4, 3, 2, 4, 5, 1.

Prosjek ocjena je 3,3 pa nakon poziva funkcije `razdvojiListe` varijabla `glava` pokazuje na listu čvorova koji sadrže sljedeće ocjene: 2, 3, 3, 2, 1. Funkcija vraća pokazivač na novu listu koja sadrži čvorove sa sljedećim ocjenama: 5, 4, 4, 4, 5.

Zadatak 2. (7 bodova)

Binarno stablo sadrži cjelobrojne elemente te je čvor definiran sljedećim kodom:

```
typedef struct s{
    int x;
    struct s *lijevo, *desno;
} cvor;
```

Zadana je funkcija

```
int magicnaTrojka(int a, int b, int c);
```

koja vraća 1 ako a, b i c čine magičnu trojku, a 0 inače.

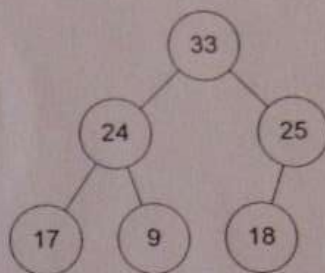
Napišite funkciju koja će vratiti koliko čvorova u stablu čini magičnu trojku sa svojom djecom. Prototip tražene funkcije je:

```
int brojMagicnih(cvor *korijen);
```

Zadatak 3. (8 bodova)

Zadan je niz brojeva: 12, 5, 7, 1, 50, 22, 5, 43.

- a) (3 boda) Ilustrirajte (nacrtajte stablo nakon **svake** promjene) stvaranje gomile od zadanog polja brojeva algoritmom čija je složenost za najgori slučaj $O(n)$.
- b) (3 boda) Ilustrirajte (nacrtajte stablo nakon **svake** promjene) stvaranje gomile od zadanog polja brojeva algoritmom čija je složenost za najgori slučaj $O(n \log n)$.
- c) (2 boda) Zadana je gomila:



Ilustrirajte izgled gomile nakon **svake** promjene dva čvora, za prva 3 koraka uzlaznog *heap sort*a.

Zadatak 4. (7 bodova)

Implementirajte razred *Kugla* s dvije (privatne) varijable jednostruke preciznosti koje predstavljaju polumjer kugle i gustoću materijala od kojeg je kugla napravljena.

Implementirajte metode za dohvat i postavljanje vrijednosti polumjera i gustoće.

Implementirajte i dvije metode koje računaju volumen i masu kugle.

Napišite odsječak glavnog programa u kojem instancirajte objekte na dva načina:

- na stogu,
- na hrpi (engl. *heap*).

Demonstrirajte njihovo korištenje pozivima svih metoda obaju objekata.

Za instanciranje objekata koristite podrazumijevani konstruktor (ne trebate pisati vlastite konstruktore).

Formule:

$$\text{masa} = \text{volumen} * \text{gustoca}$$

$$\text{volumen} = \frac{4}{3} * \text{polumjer}^3 * \pi$$