

## Algoritmi i strukture podataka

Završni ispit

13. lipnja 2011.

Rješenja za prva tri zadatka napišite na svojim papirima i predajte ih u košuljici. Rješenja za ostale zadatke napišite na za to predviđenom mjestu.

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

### Zadatak 1. (8 bodova)

U dvostruko povezanoj listi nalaze se podaci o studentima opisani sljedećom strukturom:

```
struct student {
    char imeprezime[80+1];
    int godRod;
};

struct at {
    struct student element;
    struct at *sljed;
    struct at *preth;
};

typedef struct at atom;
```

Lista je sortirana silazno (od glave) po godini rođenja. Napišite funkciju koja će iz liste izbaciti sve studente koji imaju zadani broj godina. Kao trenutačnu godinu uzmite 2011. Prototip funkcije je:

```
void izbaci(atom **glava, atom **rep, int brGodina);
```

### Zadatak 2. (8 bodova)

Binarno stablo sadrži cjelobrojne elemente, a čvor je definiran sljedećom strukturom:

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

Napišite funkciju **Listopad** koja briše sve listove zadanog stabla.

### Zadatak 3. (6 bodova)

Zadan je niz brojeva: 27, 49, 35, 19, 87, 53, 67, 29.

- Ilustrirajte (nacrtajte stablo nakon svake promjene) stvaranje gomile od zadanog polja brojeva algoritmom čija je složenost za najgori slučaj  $O(n)$ .
- Za gomilu iz a) zadatka prikažite postupak uzlaznog *heapsorta*. Prikažite svaki korak sortiranja (nacrtajte stablo nakon svake zamjene dva elementa).

#### Zadatak 4. (3 boda)

Upišite u prazan prostor s desne strane rezultat izvođenja funkcije **radiNesto** nad binarnim sortiranim stablom stvorenim iz ulaznog niza brojeva: **9, 4, 2, 6, 12, 11, 5, 19, 7, 10**.

```
int radiNesto(cvor *korijen){  
  
    int t;  
    if (korijen == NULL) return 0;  
  
    if (!korijen->lijevo && !korijen->desno) return korijen->broj;  
    t = radiNesto(korijen->desno) + radiNesto(korijen->lijevo);  
    printf("%d ",t);  
    return t;  
}
```

Rješenje:

#### Zadatak 5. (4 boda)

Dopunite programski odsječak tako da funkcija **f** rekurzivno zbraja neparne elemente u jednostruko povezanoj listi te vraća dobiveni zbroj.

```
int f(cvor *glava){  
    if(glava) {  
        if (_____  
            _____  
        else  
            _____  
    }  
    else{ return 0; }  
}
```

#### Zadatak 6. (3 boda)

Binarno stablo karakterizirano je sljedećim podacima:

- ispis **inorder**: **4,5,7,9,10,11,13,17,18,19**
- ispis **postorder**: **4,7,10,9,5,18,17,13,19,11**

Potrebno je:

a) **rekonstruirati** binarno stablo (nacrtati izgled)

b) na liniju napisati **preorder** ispis stabla: \_\_\_\_\_

Izgled stabla:

## RJEŠENJA:

### Zadatak 1. RJEŠENJE

```
void izbaci(atom **glava, atom **rep, int brGodina){
    atom2 *pom, *temp;

    pom = *glava;

    while(pom != NULL){
        if (2011- pom->element.godRod == brGodina){
            temp = pom;
            if (pom->preth == NULL){/*briše li se glava*/
                *glava = (*glava)->sljed;
                (*glava)->preth = NULL;
            }
            else{
                pom->preth->sljed = pom->sljed; /*prespoji*/
                if (pom->sljed){
                    pom->sljed->preth = pom->preth;
                }
                else{/*brisao se dotadašnji rep*/
                    *rep = pom->preth;
                }
            }
            /*pripremi se za sljedeći korak*/
            pom = pom->sljed;
            free(temp);
        }
        else if (2011 - pom->element.godRod > brGodina) break;
        else
            pom = pom->sljed;
    }
}
```

Alternativno...

```
void izbaci(atom **glava, atom **rep, int brGodina){
    atom *p = *glava;
    atom *p1 = NULL;
    atom *p2 = NULL;
    if (!p) { // ako je lista prazna
        return;
    }
    while (p && p->el.godRod > (2011 - brGodina)) {
        p = p->sljed;
    }
    if (p && p->el.godRod == (2011-brGodina)){
        p1 = p;
        p = p->preth;
    } else {
        //Ne postoji element s jednakim brojem godina
        return;
    }
    //p je sada na zadnjem elementu koji ima manje od zadanog //broja godina ili je
    NULL (ako nema zapisa s manjim brojem //godina)
    //p1 je na prvom koji ima zadani broj godina
    while (p1 && p1->el.godRod == (2011 - brGodina)) {
        //Ako postoji sljedeći, postavi njegov prethodni na p
        if (p1->sljed) {
            p1->sljed->preth = p;
        } else {
            //Ne postoji sljedeći - izbacujemo zadnji element - //moramo urediti I rep
            *rep = p;
        }
    }
}
```

```

}
//Ako postoji prethodni, postavi njegov sljedeći na
//sljedeći od trenutnog
if (p1->preth) {
    p1->preth->sljed = p1->sljed;
} else {
    //Ne postoji prethodni - glava gleda na element koji //izbacujemo
    *glava = p1->sljed;
}
p2 = p1;
p1 = p1->sljed;
free(p2);
}
}

```

## Zadatak 2. RJEŠENJE

```

cvor *Listopad(cvor *k)
{
    if(k)
    {
        if(!k->lijevo && !k->desno)
        {
            free(k);
            return NULL;
        }
        else
        {
            k->lijevo=Listopad(k->lijevo);
            k->desno=Listopad(k->desno);
        }
    }
    return k;
}

```

**Alternativno:**

```

void listopad2(cvor **k)
{
    if(*k)
    {
        if(!(*k)->lijevo && !(*k)->desno)
        {
            free(*k);
            *k=NULL;
        }
        else
        {
            listopad2(&(*k)->lijevo);
            listopad2(&(*k)->desno);
        }
    }
}

```

## Zadatak 3. RJEŠENJE

- a) napravi se potpuno stablo i radi se gomila. Rješenje:

```

    27
  49 35
19 87 53 67
29

```

```

    27
  49 35
29 87 53 67
19

```

```

    27
  49 67
29 87 53 35
19

```

```

    27
  87 67
29 49 53 35
19

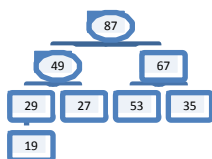
```

87  
 27 67  
 29 49 53 35  
 19

GOMILA:

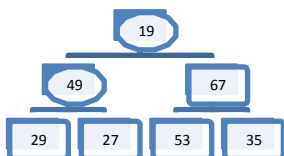
87  
 49 67  
 29 27 53 35  
 19

b)



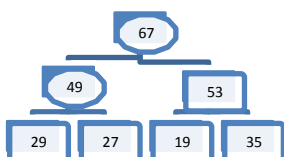
87	49	67	29	27	53	35	19
----	----	----	----	----	----	----	----

Zamjena



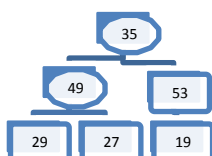
19	49	67	29	27	53	35	87
----	----	----	----	----	----	----	----

Podešavanje:



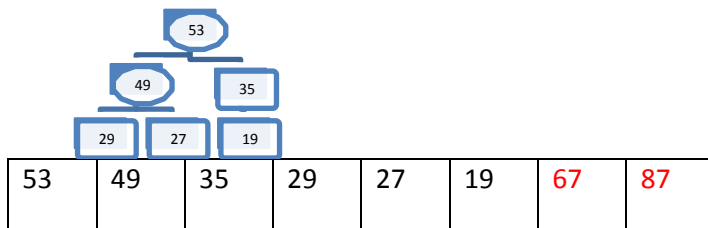
67	49	53	29	27	19	35	87
----	----	----	----	----	----	----	----

Zamjena

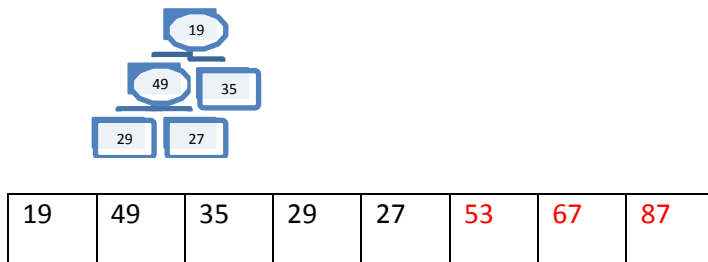


35	49	53	29	27	19	67	87
----	----	----	----	----	----	----	----

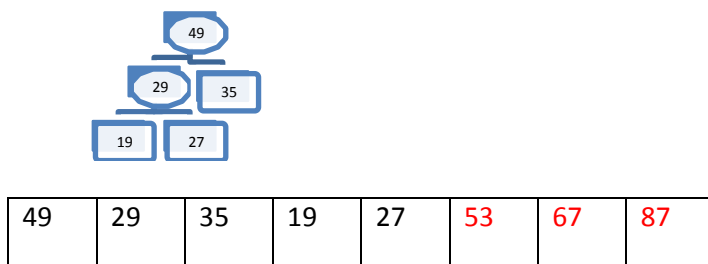
Podešavanje:



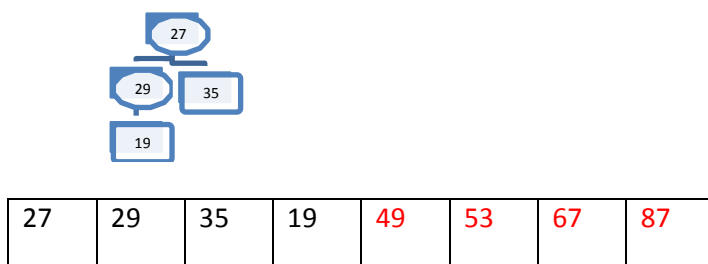
Zamjena:



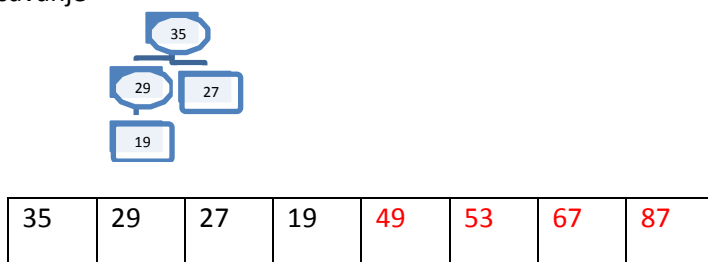
Podešavanje:



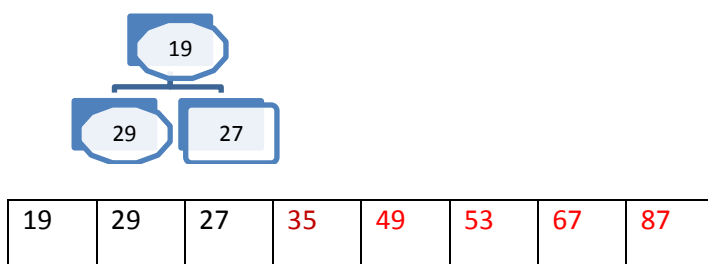
Zamjena



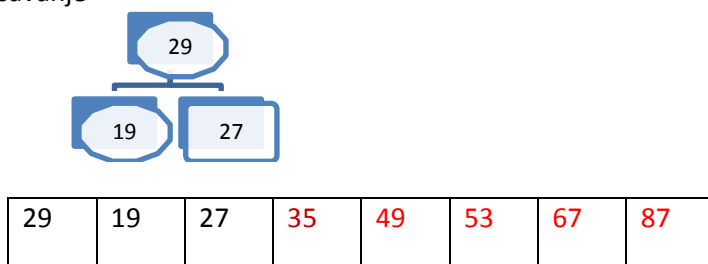
Podešavanje



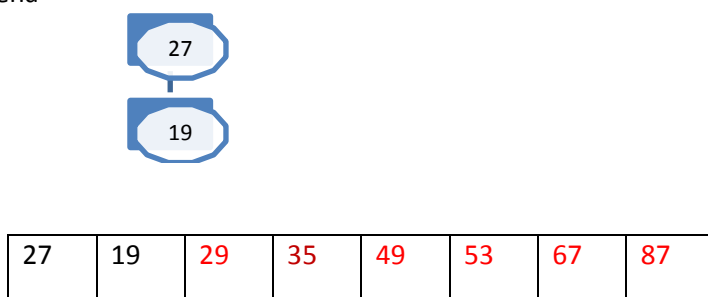
Zamjena



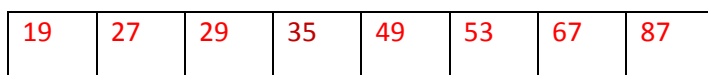
Podešavanje



Zamjena



Podešavanje – ok  
Zamjena



#### **Zadatak 4. RJEŠENJE**

RJEŠENJE:

Ispisuje u svakom koraku zbroj vrijednosti listova podstabala trenutnog čvora (osim u slučaju kad je trenutni čvor list). Također valjda primjetiti da se kreće prvo s desnim podstablom.

ISPIS: 10 29 12 14 43

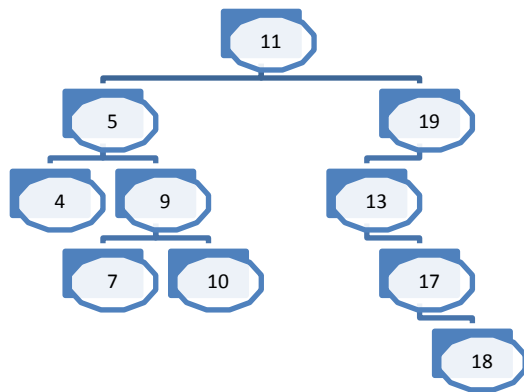
### **Zadatak 5. RJEŠENJE**

Rješenje:

```
(glava->element)%2  
  
return glava->element + f(glava->sljed);  
return f(glava->sljed);
```

### **Zadatak 6. RJEŠENJE**

RJEŠENJE



Preorder: 11,5,4,9,7,10,19,13,17,18

(\*stablo stvoreno iz : 11, 19, 5, 13, 9, 17, 18, 10, 4, 7)