

Algoritmi i strukture podataka – 1. ispitni rok

5. srpnja 2016.

Nije dopušteno korištenje globalnih i statičkih varijabli te naredbe goto. Ovaj primjerak ispita trebate predati s upisanim imenom i prezimenom te JMBAG-om.

Ispit donosi maksimalno 70 bodova, a prag za prolaz pismenog ispita je 35 bodova uz barem jedan točno riješen zadatak.

Zadatak 1. (10 bodova)

Za zadanu matricu incidencije nacrtajte usmjereni graf.

$$Inc = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & -1 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Zadatak 2. (20 bodova)

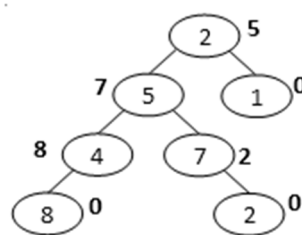
Čvorovi binarnog stabla sadrže binarne brojeve i definirani su strukturom **cvor** :

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

Harmanov zbroj stabla je definiran kao zbroj *cMax* (child maximal) vrijednosti svakog čvora. *cMax* vrijednost za čvor je definirana kao veća od vrijednosti lijevog i desnog djeteta. Ako čvor nema niti jedno dijete, podrazumijeva se da je njegova *cMax* vrijednost 0. Napisati funkciju koja za zadano stablo računa Harmanov zbroj (ulazno stablo mora ostati nepromijenjeno). Prototip funkcije je

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

Na slici je prikazan primjer stabla, a pored svakog čvora dana je njegova *cMax* vrijednost. Harmanov zbroj prikazanog stabla je 22.



Zadatak 3. (20 bodova)

Neka je zadan niz znakova X. Potrebno je preurediti elemente niza X i to tako da se napravi zamjena uzastopnih parova znakova. Napišite dvije rekurzivne funkcije, *zamijeniL* i *zamijeniD*, koje rade traženu promjenu, s time da *zamijeniL* radi promjene uzastopnih parova gledano s lijeva, a *zamijeniD* radi promjene uzastopnih parova gledano s desna. Prototipovi funkcija su:

```
void zamijeniD(char *A, int duljina); void zamijeniL(char *A, int duljina);
```

pri čemu vrijednost varijable *duljina* označava broj znakova u nizu X. Primjer: za X="AbCdEfG" primjena funkcije *zamijeniL* daje "bAdCfEG", dok primjena funkcije *zamijeniD* daje "ACbEdGf".

Napomena: nerekurzivno rješenje se neće priznati.

Zadatak 4. (10 bodova)

Odredite vrijeme izvođenja u O , Ω i, ako je moguće, Θ notaciji za programski odsječak u a) dijelu zadatka i za funkciju **det** u b) dijelu zadatka. Ako se vrijeme izvođenja u Θ notaciji ne može odrediti, navedite tako u rješenju. Rješenja upišite u na to predviđeno mjesto ispod zadatka.

a)

```
/* A je polje n*n cijelih brojeva,
   sve varijable su tipa int, funkcija f
   ima asimptotsku složenost  $\Theta(n)$  */
```

```
tSum=0;
for(i=0;i<n; i++)
    for (j=1; j<=n; j*=2)
        tsum+=A[n*i+j-1]*f(n,i,j);

for(i=0;i<n;i++){
    sSum=A[i*n+0]+A[i*n+1]+A[i*n+2];
    if (sSum>0){
        for(k=0;k<n;k++)
            sSum+= A[n*i+k]*f(n,i,k+1);
        tSum += sSum
    }
}
```

O _____ Ω _____ Θ _____

b)

```
/* A je polje n*n double brojeva, a
   pretpostavljamo da funkcije
   preuredi i free imaju asimptotsku
   složenosti  $\Theta(1)$ . */
```

```
double det(double *A, int n){
    double sm,s, *pom;
    int j;
    if (n == 1) return A[0];
    s = 1; sm = 0;
    for (j = 0; j < n; j++){
        pom = preuredi(A, n, j+1);
        sm += (s*A[j] * det(pom, n - 1));
        s = -s; free(pom);
    }
    return sm;
}
```

O _____ Ω _____ Θ _____

Zadatak 5. (10 bodova)

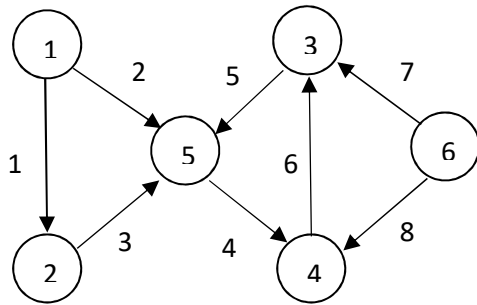
U cjelobrojnomoj polju pohranjen je sljedeći niz brojeva:

15, 2, 12, 8, 13, 4, 5, 9, 16, 17, 3

Ilustrirajte uzlazno sortiranje algoritmom Shellsort s koracima {5, 2, 1}. Potrebno je prikazati sadržaj polja nakon svake promjene.

Rješenja:

1. Zadatak



2. Zadatak

```
int HarmanovZbroj(cvor *korijen){
    if (!korijen) return 0; // nije potrebno ako se vrijednosti u podstablama
                           // računaju samo ako postoje.

    int ld, dd, cMax;
    // ovo su vrijednosti lijeve i desne strane ako odgovarajuće dijete ne postoji
    ld = 0; dd = 0; // ovo su vrijednosti lijeve i desne strane ako ne postoje
    if (korijen->lijevo){ // ako postoji lijevo dijete
        ld = korijen->lijevo->vrijednost; // vrijednost lijevog djeteta
    }
    if (korijen->desno){ // ako postoji desno dijete
        dd = korijen->desno->vrijednost; // vrijednost desnog djeteta
    }
    if (ld > dd){ // izračunaj cMax po definiciji
        cMax = ld;
    }
    else cMax = dd;
    // vrati cMax za promatrani čvor uvećan za zbroj cMax vrijednosti u lijevom,
    // odnosno desnom podstablu.
    return cMax + HarmanovZbroj(korijen->lijevo) + HarmanovZbroj(korijen->desno);
}
```

3. Zadatak

```
void zamijeniD(char *A, int duljina){
    char tmp;
    if (duljina == 1){
        return;
    }
    if (duljina == 2){
        tmp = A[0]; A[0] = A[1]; A[1] = tmp;
        return;
    }
    tmp = A[duljina - 1]; A[duljina - 1] = A[duljina - 2]; A[duljina - 2] = tmp;
    zamijeniD(A, duljina - 2);
}

void zamijeniL(char *A, int duljina){
    char tmp;
    if (duljina == 1){
        return;
    }
    tmp = A[0]; A[0] = A[1]; A[1] = tmp;
    if (duljina == 2){
        return; // bila su dva znaka i zamijenili smo ih
    }
    zamijeniL1(&A[2], duljina - 2); /*ove dvije linije mogu biti i obrnute*/
}
```

Prethodna dva rješenja se mogu i drugačije kodirati, npr. zamijeniD se može napraviti kao u slijedećoj funkciji:

```
void zamijeniD(char *A, int duljina){
    char tmp;
    if (duljina < 1){
        return;
    }
    tmp = A[duljina - 1]; A[duljina - 1] = A[duljina - 2]; A[duljina - 2] = tmp;
    zamijeniD(A, duljina - 2);
}
```

4. Zadatak

a) U najgorem slučaju je $O(n^3)$, u najboljem $\Omega(n^2 \log(n))$, a asimptotska ne postoji

b) Sve tri složenosti su $n!$

neka je $T(n)$ =složenost za $\det(n)$

$$T(n) = 2 \cdot n \text{ (množenja)} + n \text{ (zbrajanja)} + n \cdot T(n-1) = \Theta(n) + n \cdot T(n-1) =$$

$$= \Theta(n) + n (\Theta(n-1) + (n-1) \cdot T(n-2)) = \Theta(n) + \Theta(n \cdot (n-1)) + n \cdot (n-1) \cdot T(n-2)$$

$$= \Theta(n \cdot (n-1)) + n \cdot (n-1) \cdot (\Theta(n-2) + (n-2) \cdot T(n-3)) = \text{(sličan račun)}$$

$$= \Theta(n \cdot (n-1) \cdot (n-2)) + n(n-1)(n-2)T(n-3) = \text{analognim računom se dođe do faktoriijela}$$

5. Zadatak

korak: 5

15, 2, 12, 8, 13, 4, 5, 9, 16, 17, 3
4, 2, 12, 8, 13, 15, 5, 9, 16, 17, 3
4, 2, 9, 8, 13, 15, 5, 12, 16, 17, 3
4, 2, 9, 8, 13, 3, 5, 12, 16, 17, 15

korak: 2

3, 2, 9, 8, 13, 4, 5, 12, 16, 17, 15
3, 2, 9, 4, 13, 8, 5, 12, 16, 17, 15
3, 2, 9, 4, 5, 8, 13, 12, 16, 17, 15
3, 2, 5, 4, 9, 8, 13, 12, 16, 17, 15

korak: 1

3, 2, 5, 4, 9, 8, 13, 12, 15, 17, 16
2, 3, 5, 4, 9, 8, 13, 12, 15, 17, 16
2, 3, 4, 5, 9, 8, 13, 12, 15, 17, 16
2, 3, 4, 5, 8, 9, 13, 12, 15, 17, 16
2, 3, 4, 5, 8, 9, 12, 13, 15, 17, 16
2, 3, 4, 5, 8, 9, 12, 13, 15, 16, 17