

Algoritmi i strukture podataka – 1. ispitni rok

4. srpnja 2017.

Nije dopušteno korištenje globalnih i statičkih varijabli te naredbe goto. Ispit donosi maksimalno 70 bodova. Ovaj primjerak ispita trebate predati s upisanim imenom i prezimenom te JMBAG-om.

Zadatak 1. (14 bodova)

Ulaznu dvostruko povezanu listu koja sadrži brojeve preuredite tako da bude uzlazno sortirana. Ne smijete kopirati čvorove ulazne liste, ni uređivati sadržaj postojećih čvorova, izuzev pokazivača na prethodni i sljedeći element.

Čvor liste sljedeće je strukture:

```
typedef struct atom{
    int el;
    struct atom* sljed, *pret;
} cvor;
```

Prototip funkcije jest:

```
void sortiraj (cvor ** glava, cvor ** rep);
```

Zadatak 2. (14 bodova)

Zadan je niz brojeva: **3, 68, 40, 87, 29, 33, 26, 51, 28, 60**.

a) **(6 bodova)** Sortirajte niz silazno shell sortom za niz koraka {4,3,1}.

b) **(8 bodova)** Sortirajte niz silazno heapsortom. Prikažite postupak do trenutka kada ste sortirali barem 4 elementa.

Potrebno je prikazati polje nakon svake zamjene dvaju elemenata u oba dijela zadatka.

Zadatak 3. (12 bodova)

Odredite, gdje je moguće, vrijeme izvođenja u O , Ω i Θ notaciji za funkcije **f1** i **f2**. Ako se vrijeme izvođenja ne može odrediti, navedite tako u rješenju. Rješenja upišite u tablice pored zadataka.

a)

```
int f1(cvor* korijen) {
    int suma=0;
    if (korijen==NULL) return 0;
    while(korijen){
        suma+=f1(korijen->desno)+f1(korijen->lijevo)
    }
    return suma;
}
```

O	
Ω	
Θ	

b)

```
//ulaz je uređeno binarno stablo
int f2(cvor* korijen, int el) {
    if (korijen==NULL) return 0;
    if (el==korijen->el) return 1;
    if (el<korijen->el) return f2(korijen->lijevo,el);
    else return f2(korijen->desno,el);
}
```

O	
Ω	
Θ	

Zadatak 4. (15 bodova)

Zadani su tipovi podatka **Red** i **Stog** za koje su definirane funkcije za inicijalizaciju, te za dodavanje i skidanje elementa podatkovnog tipa **char**. Prototipovi navedenih funkcija su:

```
void init_red(Red *red);  
int dodajURed(char element, Red *red);  
int skiniIzReda(char *element, Red *red);  
void init_stog(Stog *stog);  
int dodajNaStog(char element, Stog *stog);  
int skiniSaStoga(char *element, Stog *stog);
```

Funkcije **dodajURed**, **dodajNaStog**, **skiniIzReda** i **skiniSaStoga** vraćaju 1 ako je operacija uspjela, a 0 ako nije.

Napišite funkciju koja će sve znamenke ('0'-'9') iz zadanog reda izdvojiti na stog na način da se prva znamenka koja se pojavljuje u redu nalazi na vrhu stoga, a zadnja na dnu. U redu ostaju svi ostali znakovi u nepromijenjenom redoslijedu. Prototip funkcije je:

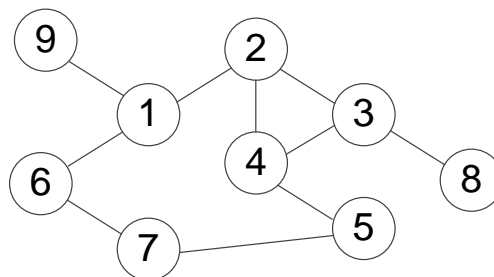
```
void izdvojiZnamenke(Red *red, Stog *stog);
```

Napišite i dio glavnog programa u kojem ćete definirati red i stog i ispravno pozvati napisanu funkciju. Dio koda u kojem se puni red nije potrebno pisati.

Primjer: Za ulazni red sadržaja **1, a, b, 2, c, 3, 4, d** poziv funkcije **izdvojiZnamenke** puni stog sadržajem (gledano od dna prema vrhu) **4, 3, 2, 1** dok u starom redu ostaju **a, b, c, d**.

Zadatak 5. (15 bodova)

Zadan je graf prikazan slikom:



Prikažite kako izgleda ispis vrhova grafa korištenjem DFS i BFS načina obilaska grafa, uz pretpostavku da se kreće od čvora s vrijednošću 4. Prikažite korake u ispisu vrijednosti vrhova grafa, redoslijed posjećivanja vrhova te sadržaj pratećih pomoćnih struktura stog i red u svakom koraku.

Napomena: prilikom odabira neobiđenih susjednih vrhova za oba načina obilaska grafa vrhove odabirati prema **rastućoj** vrijednosti vrha.

Rješenja:

1. zadatak (14 bodova)

```
void sortiraj (cvor ** glava, cvor ** rep)
{
    //ovdje koristimo insertion sort, no i drugi izbori su ok
    cvor *tren, *bouncer, *sljed;
    if (!*glava) return; //prazna lista
    tren=(*glava)->sljed; //od drugog elementa
    while (tren)
    {
        bouncer=tren->preth;
        sljed=tren->sljed;
        while(bouncer && bouncer->el>tren->el) bouncer=bouncer->preth;
        if (bouncer!=tren->preth){
            // zamjene!
            //izbacivanje trenutnog sa starog mjesta
            tren->preth->sljed=tren->sljed;
            if (tren->sljed) tren->sljed->preth=tren->preth;
            else *rep=tren->preth;
            //ubacivanje trenutnog na pravo mjesto
            tren->preth=bouncer;
            if (bouncer){
                tren->sljed=bouncer->sljed;
                bouncer->sljed=tren;
            }
            else
            {
                tren->sljed=*glava;
                *glava=tren;
            }
        }
        tren=sljed;
    }
}
```

2. zadatak (14 bodova)

a) shell {4,3,1}

3, 68, 40, 87, 29, 33, 26, 51, 28, 60

29, 68, 40, 87, 3, 33, 26, 51, 28, 60

29, 68, 40, 87, 28, 33, 26, 51, 3, 60

29, 68, 40, 87, 28, 60, 26, 51, 3, 33

k=3

29, 68, 40, 87, 28, 60, 26, 51, 3, 33

87, 68, 40, 29, 28, 60, 26, 51, 3, 33

87, 68, 60, 29, 28, 40, 26, 51, 3, 33

87, 68, 60, 29, 51, 40, 26, 28, 3, 33

87, 68, 60, 33, 51, 40, 29, 28, 3, 26

k=1

87, 68, 60, 33, 51, 40, 29, 28, 3, 26

87, 68, 60, 51, 33, 40, 29, 28, 3, 26

87, 68, 60, 51, 40, 33, 29, 28, 3, 26

87, 68, 60, 51, 40, 33, 29, 28, 26, 3

b) heapsort

izgradnja prvog min heapa

3, 68, 40, 87, 29, 33, 26, 51, 28, 60

3, 68, 40, 28, 29, 33, 26, 51, 87, 60

3, 68, 26, 28, 29, 33, 40, 51, 87, 60

3, 28, 26, 51, 29, 33, 40, 68, 87, 60

sortiranje

3, 28, 26, 51, 29, 33, 40, 68, 87, 60

60, 28, 26, 51, 29, 33, 40, 68, 87, 3

26, 28, 33, 51, 29, 60, 40, 68, 87, 3

87, 28, 33, 51, 29, 60, 40, 68, 26, 3

28, 29, 33, 51, 87, 60, 40, 68, 26, 3

68, 29, 33, 51, 87, 60, 40, 28, 26, 3

29, 51, 33, 68, 87, 60, 40, 28, 26, 3

40, 51, 33, 68, 87, 60, 29, 28, 26, 3

sortirana su 4 elementa, dalje nije potrebno

3. zadatak (8 bodova)

a) ne može se odrediti niti jedan odgovor –jer ne završava u konačnom broju koraka kako se dobije beskonačna petlja za svako neprazno stablo

b)

$O(n)$ //koso stablo

$\Omega(1)$ //u korijenu je traženi element

Θ se ne može odrediti

4. zadatak (15 bodova)

```
void izdvojiZnamenke(Red *red, Stog *stog){

    Red pomR;
    Stog pomS;
    int element;

    init_red(&pomR);
    init_stog(&pomS);

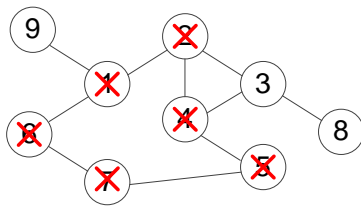
    while(skini(&element, red)){
        if (element>='0' && element<='9')
            dodajNaStog(element, &pomS);
        else dodajURed(element, &pomR);
    }
    while(skiniIzReda(&element, &pomR)
        dodajURed(element, red);
    while(skiniSaStoga(&element, &pomS)
        dodajNaStog(element, stog);
    return;
}

int main (){
    Red red, Stog stog;

    init_red(&red);
    init_stog(&stog);
    /*... napuni red... */
    izdvojiZnamenke(&red, &stog);
    ...
}
```

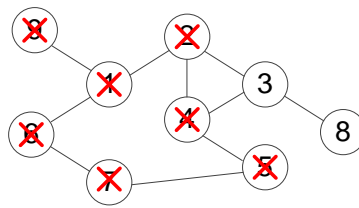
5. zadatak (15 bodova)

Rješenje za DFS:



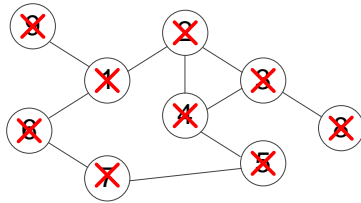
5
7
6
1
2
4

4 2 1 6 7 5



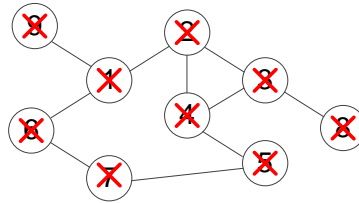
9
1
2
4

4 2 1 6 7 5 9



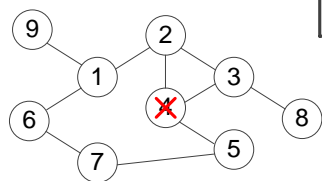
8
3
2
4

4 2 1 6 7 5 9 3 8

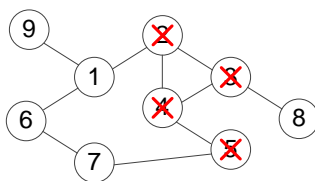


4 2 1 6 7 5 9 3 8

Rješenje za BFS:



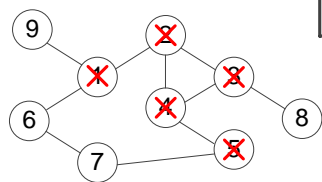
4



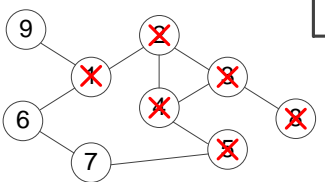
2 3 5

4

4



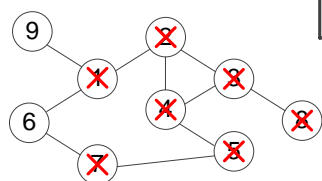
3 5 1



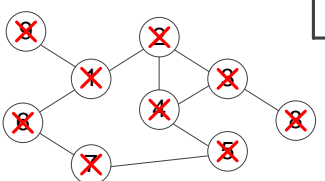
5 1 8

4 2

4 2 3



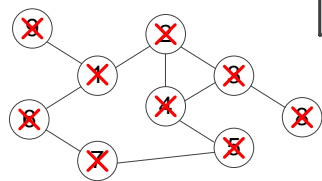
1 8 7



8 7 6 9

4 2 3 5

4 2 3 5 1



4 2 3 5 1 8 7 6 9