JMBAG **IME I PREZIME**

Algoritmi i strukture podataka – Završni ispit

18. lipnja 2013.

Nije dopušteno korištenje globalnih i statičkih varijabli te naredbe ***goto****. Neefikasna rješenja mogu donijeti manje bodova. Nerekurzivne funkcije se ne priznaje kao rješenja u zadatcima u kojima se traži rekurzivna funkcija. Ispit nosi maksimalno 30 bodova, a prag za prolaz završnog ispita je* ***10*** *bodova. 4. zadatak rješavate na ovom obrascu dok ostale zadatke rješavate na svojim listovima papira. Ovaj obrazac morate predati.*

Zadatak 1. (6 bodova)

Za tip podatka `Red` definirane su funkcije (nije ih potrebno pisati) za inicijalizaciju reda, dodavanje elementa u red te za skidanje elementa iz reda. Prototipovi navedenih funkcija su:

```
void init_red(Red *red);
int dodaj(Element element, Red *red);
int skini(Element *element, Red *red);
```

Funkcije **`dodaj`** i **`skini`** vraćaju 1 ako je operacija dodavanja ili skidanja uspjela, a 0 inače. Elementi reda su podaci tipa **`Element`**. Tipovi **`Element`** i **`Red`** definirani su redom sljedećim programskim odsječkom:

```
#define MAXNIZ 20
typedef struct Element{
    char niz[MAXNIZ + 1];
} Element;
```

```
typedef struct at {
      Element element;
      struct at *sljed;
} atom;
```

```
typedef struct {
      atom *ulaz;
      atom *izlaz;
} Red;
```

Napišite funkciju koja će iz zadanog reda u novi red kopirati sve elemente čija varijabla `niz` sadrži podniz koji funkcija primi kao ulazni argument. Funkcija pozivatelju vraća taj novi red, a prototip joj je:

```
Red* trazi(Red *red, char* podniz)
```

Poredak elemenata u novom redu mora biti isti kao i poredak u ulaznom redu. Po završetku izvođenja funkcije, ulazni red mora sadržavati iste podatke u istom redoslijedu kao i prije poziva funkcije `trazi`.

Ako npr. zadani red sadrži elemente sa sljedećim nizovima znakova: ''234dd6'', ''aBc569'', ''1234d'', ''ab45de'', a traženi podniz je ''34d'', funkcija mora vratiti novi red čiji elementi sadrže nizove ''234dd6'' i ''1234d''.

Napomene: Rješenja koja neće koristiti zadane funkcije za rad s redom donijet će 0 bodova. Možete koristiti pomoćne redove unutar funkcije **`trazi`**. *Skrećemo pozornost da bi funkcije `strcpy` i `strstr` iz standardne programske knjižnice C-a mogle biti od koristi.*

Zadatak 2. (6 bodova)

U memoriji postoji jednostruko povezana sortirana lista ostvarena (realizirana) strukturom:

```
struct el {
      struct el* sljedeci;
      int vrijednost;
};
typedef struct el element;
```

Napišite funkciju koja će iz liste ukloniti sva ponavljanja nekog elementa ostavljajući u listi samo **prvi** takav element gledano od početka liste. Memoriju koju su zauzimali uklonjeni elementi treba osloboditi. Funkcija prima pokazivač na prvi član liste, a prototip joj je:

```
void izbaci(element* glava);
```

Zadatak 3. (6 bodova)

Zadan je niz brojeva: **`30, 45, 31, 60, 87, 95, 12, 69`**

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(nlogn).*

Zadatak 4. (6 bodova)

Binarno stablo za pretraživanje organizirano je po pravilu da su brojevi u lijevom podstablu manji od brojeva u desnom. **Postorder** ispis binarnog stabla za pretraživanje je:

`3, 5, 4, 1, 8, 15, 16, 11, 7, 6`

a) **(2 boda) Rekonstruirajte** binarno stablo za pretraživanje (nacrtajte izgled) u prostor s desna

Izgled stabla:

b) **(4 boda)** Binarno stablo za pretraživanje, zadano postorder ispisom, ostvareno je statičkom strukturom polje, pri čemu se prvi element polja (indeks nula) ne koristi, nego je korijen stabla na indeksu 1. Napišite, u odgovarajućem redoslijedu, niz indeksa **(i_1, i_2, i_3, ... i_{10})** u polju na kojima se nalaze elementi ispisani postorder obilaskom stabla. Na primjer, i_3 je indeks na kojem se u polju nalazi 3. po redu ispisani element stabla. Odgovor napišite na crtu ispod.

_____

Zadatak 5. (6 bodova)

Zadan je tip podatka **`Stog`** za koji su definirane funkcije za inicijalizaciju stoga, dodavanje elementa na stog i brisanje elementa sa stoga. Elementi stoga su podaci **tipa `Student`** koji sadrže prezime studenta (40 znakova), ime studenta (30 znakova) te broj bodova ostvarenih na predmetu ASP. Prototipovi navedenih funkcija su:

```
void init_stog(Stog *stog);
int dodaj(Student element, Stog *stog);
int skini(Student *element, Stog *stog);
```

Funkcije `dodaj` i `skini` vraćaju 1 ako je operacija dodavanja ili skidanja uspjela, a 0 inače.

a) **(1 bod)** Napišite **definicije** odgovarajućih struktura (tipovi podataka **`Student`, `Stog`** i **`atom`**) za stog realiziran vezanom listom.

b) **(5 bodova)** Napišite funkciju koja će sa stoga ukloniti sve studente koji imaju broj bodova manji od prosjeka bodova svih studenata na stogu. Redoslijed studenata koji su ostavljeni na stogu mora biti isti kao i prije poziva funkcije. Funkcija vraća broj studenata uklonjenih sa stoga. Prototip funkcije treba biti:

```
int MakniIspodProsjeka (Stog *stog);
```

Napomene: Rješenja koja neće koristiti zadane funkcije za rad s stogom donijet će 0 bodova. Možete koristiti pomoćne stogove unutar funkcije **`MakniIspodProsjeka`**.

Rješenja:

Zadatak 1. (6 bodova)

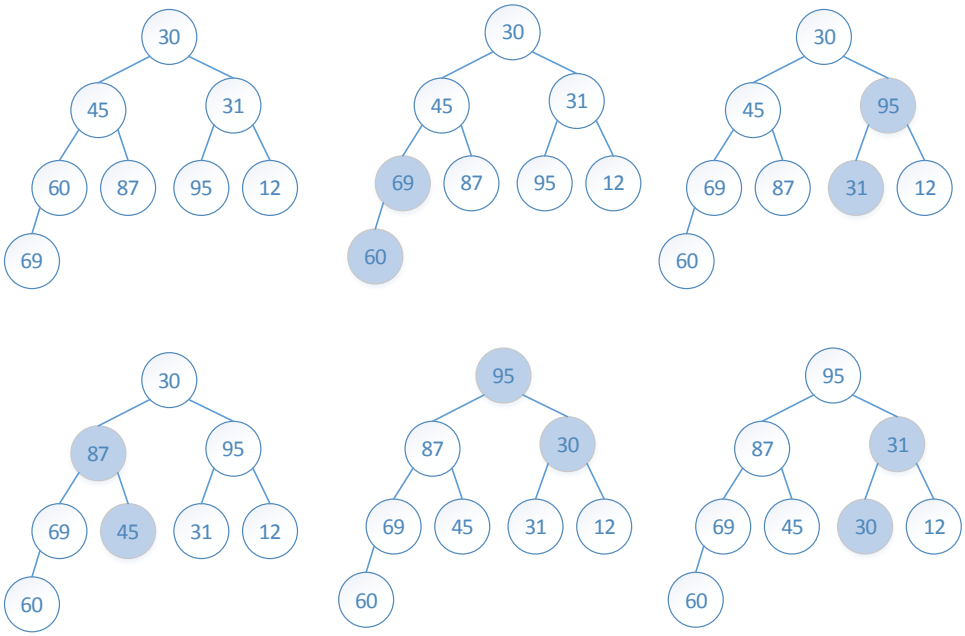
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

Red* trazi(Red *red, char* podniz){
    Red *rezultat = NULL, pom;
    Element element;
    /*Inicijaliziraj redove*/
    rezultat = (Red*)malloc(sizeof(Red));
    init_red(rezultat);
    init_red(&pom);
    /*Iteracija po svim elementima reda*/
    while(skini(&element, red)){
        /*Provjeri sadrzi li element reda zadani podniz*/
        if(strstr(element.niz, podniz)){
            /*Dodaj novi element u red koji moramo vratiti.
           Ovdje se pretpostavilo da 'dodaj' radi kopiranje.
           Rješenje koje to ne pretpostavlja je isto u redu*/
           dodaj(element, rezultat);
        }
        /*Dodaj element na pomocni red*/
        dodaj(element, &pom);
    }
    /*Vrati elemente na originalni red*/
    while(skini(&element, &pom)){
        dodaj(element, red);
    }
    return rezultat;
}
```

Zadatak 2. (6 bodova)

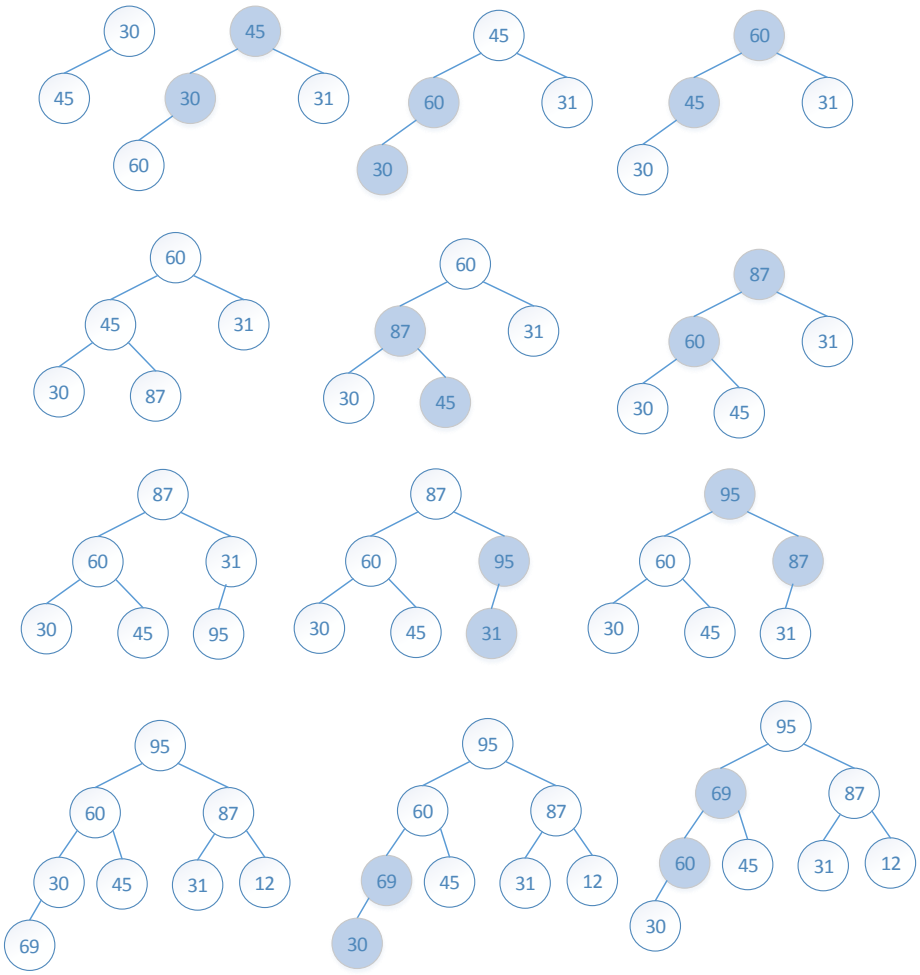
```c
void izbaci(element *glava){
    element *prvi, *drugi, *preth;
    for(prvi = glava; prvi; prvi = prvi->sljedeci){
        drugi = prvi->sljedeci;
        preth = prvi;
        while(drugi){
            if(drugi->vrijednost == prvi->vrijednost){
                preth->sljedeci = drugi->sljedeci;
                free(drugi);
                drugi = preth->sljedeci;
            }else{
                break;
            }
        }
    }
}
```

Zadatak 3. (6 bodova)

a)

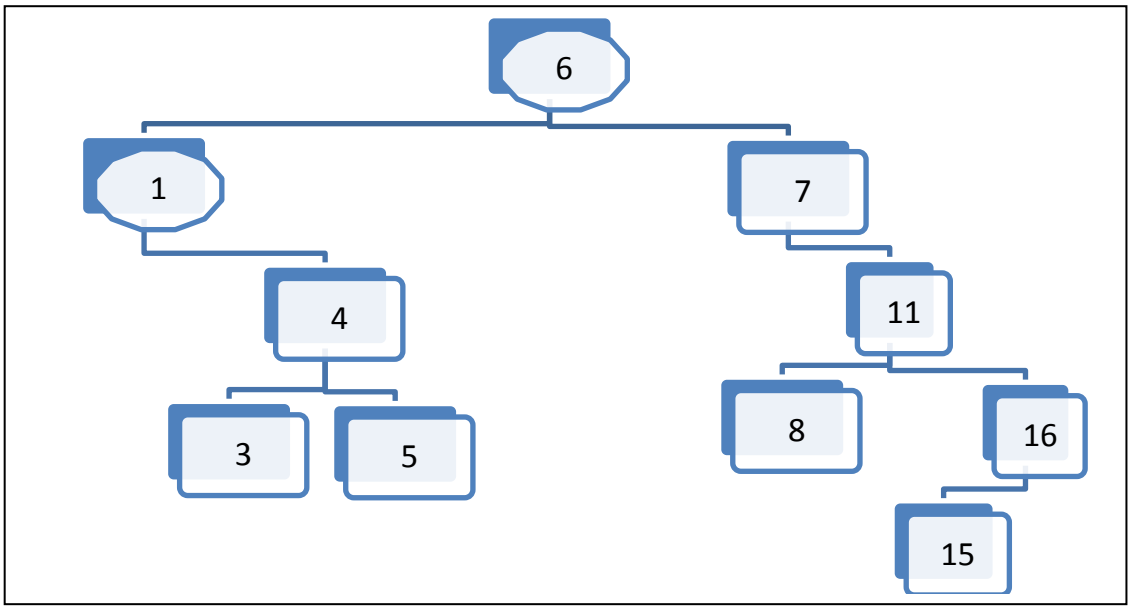


b)



Zadatak 4. (bodova)

a)



b) (10,11,5,2,14,30,15,7,3,1)

Zadatak 5. (bodova)

```
typedef struct
{
    char Prezime[40+1];
    char Ime[30+1];
    int Bodova;
} Student;

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

typedef struct at atom;

typedef struct{
    atom *vrh;
} Stog;

int MakniIspodProsjeka(Stog *stog){
    Stog pom;
    Student el;
    int br=0; float sum = 0;float pros;
    init_stog(&pom);
    while (skini(&el,stog)){
        sum+=el.Bodova;
        br++;
        dodaj(el, &pom);
    }
    pros=sum/br;
    br = 0;
    while (skini(&el,&pom)){
        if(pros>=el.Bodova){
            dodaj(el, stog);
        }else{
            br++;
        }
    }
    return br;
}
```