

Strukture podataka i algoritmi 2
Januar, 2012

1. Napraviti sledeće funkcije za manipulaciju B-stablom, ukoliko se zna da dimenzije stabla znatno premašuju veličinu radne memorije:
 - a. Pretraživanje.
(10 poena)
 - b. Dodavanje čvora u stablo.
(20 poena)

U nastavku je dato jedno od rešenja studenata.

Rešenje 1

(Tomislav Mrđa 77/10)

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

#define MAX 4

struct cvor{
    int n;
    int k[MAX-1];
    char s[MAX][20];
};

#define novi(x) x=(struct cvor*)malloc(sizeof(struct cvor))

void ucitaj(struct cvor **p,char s[]){
    struct cvor *t=*p;
    FILE *f;
    int i;
    f=fopen(s,"r");
    if(f){
        if(!t) novi(t);
        fscanf(f,"%d",&t->n);
        for(i=0;i<t->n;i++) fscanf(f,"%d",&t->k[i]);
        for(i=0;i<t->n;i++) fscanf(f,"%s",t->s[i]);
        *p=t;
        fclose(f);
    }
    else *p=NULL;
}

void upisi(struct cvor *p,char s[]){
    FILE *f;
    int i;
    f=fopen(s,"w");
    fprintf(f,"%d\n",p->n);
    for(i=0;i<p->n;i++) fprintf(f,"%d ",p->k[i]);
    fprintf(f,"\n");
    for(i=0;i<p->n;i++) fprintf(f,"%s ",p->s[i]);
    fclose(f);
}

void novoime(int nivo,int br,char ime[]){
    int i,n=0,m=0;
    char s[20];
    while(nivo>0){
```

```

        s[n++]=nivo%10+48;
        nivo=nivo/10;
    }
    for(i=0;i<n;i++) ime[i]=s[n-1-i];
    ime[n]='-';
    while(br>0){
        s[m++]=br%10+48;
        br=br/10;
    }
    for(i=0;i<m;i++) ime[n+1+i]=s[n-1-i];
    ime[n+m+1]='\0';
}

struct cvor *trazi(int kljuc,int *indeks){
    struct cvor *p;
    int i;
    novi(p);
    if(!p){
        printf("Greska pri alokaciji\n");
        exit(0);
    }
    ucitaj(&p,"1-1");
    while(p){
        for(i=0;i<p->n && p->k[i]<kljuc;i++);
        if(i<p->n && p->k[i]==kljuc){
            *indeks=i;
            return p;
        }
        else ucitaj(&p,p->s[i]);
    }
    return NULL;
}

void dodaj(int kljuc,char ime[],int nivo,int br){
    struct cvor *p=NULL;
    int i;
    ucitaj(&p,ime);
    //dodajemo cvor
    if(!p){
        novi(p);
        p->n=1;
        p->k[0]=kljuc;
        strcpy(p->s[0],"null");
        strcpy(p->s[1],"null");
        novoime(nivo,br,ime);
        upisi(p,ime);
        free(p);
    }
    else{
        //da li podatak moze da se smesti u cvor
        if(p->n<MAX-1){
            for(i=p->n;i>0 && p->k[i-1]>kljuc;i--){
                p->k[i]=p->k[i-1];
                strcpy(p->s[i+1],p->s[i]);
            }
            p->k[i]=kljuc;
            (p->n)++;
            strcpy(p->s[i+1],"null");
            upisi(p,ime);
        }
        else{
            for(i=0;i<p->n && p->k[i]<kljuc;i++);
            dodaj(kljuc,p->s[i],nivo+1,i+1);
            upisi(p,ime);
            free(p);
        }
    }
}

```

```
}
```

```
main(){  
    char s[20];  
    int n,i,k;  
    struct cvor *p;  
    strcpy(s,"1-1");  
    printf("n: ");  
    scanf("%d",&n);  
    for(i=0;i<n;i++){  
        scanf("%d",&k);  
        dodaj(k,s,1,1);  
    }  
    printf("Trazi: ");  
    scanf("%d",&n);  
    p=trazi(n,&i);  
    if(!p) printf("Trazeni broj ne postoji\n");  
    else printf("%d\n",p->k[i]);  
}
```