

Strukture podataka i algoritmi 1
Jun, 2009

1. Kompanija IMI-Soft je odlučila da razvije sopstveni paket za vektorsku grafiku pod nazivom ŠKRABALO, po ugledu na poznate programe kao što su Corel Draw i Adobe Illustrator. Iz tog razloga je od Vas naručila da napravite jezgro budućeg programa za crtanje, tako da ono zadovolji sledeće zahteve:
- a. U programu je moguće crtati tačke i duži.
Svaka **tačka** je definisana rednim brojem (ceo broj) i koordinatama x i y (realni brojevi).
Svaka **duž** je definisana rednim brojem (ceo broj) i početnom i krajnjom tačkom (pokazivači na početnu i krajnju tačku).
Imati u vidu da dve duži mogu imati jednu ili dve zajedničke tačke.
(3 poena)

 - b. Podaci o nekom crtežu su zapisani u datoteci čiji je format:

```
n          // broj tacaka
rbt1 x1 y1
rbt2 x2 y2
...
rbtn xn yn
m          //broj duzi
rbd1 t11 t12
rbd2 t21 t22
...
rbdm tm1 tm2
```

Napisati funkciju *ReadPoints* koja iz zadate datoteke učitava podatke o tačkama.
(3 poena)

Napisati funkciju *FindPoint* koja vraća pokazivač na tačku sa zadatim rednim brojem.
(3 poena)

Napisati funkciju *ReadLines* koja iz zadate datoteke učitava podatke o dužima i svaku duž povezuje sa odgovarajućim tačkama.
(5 poena)

Napisati funkciju *ReadDrawing* koja iz zadate datoteke učitava crtež pozivanjem prethodne tri funkcije.
(1 poen)

 - c. Napisati funkciju *DeletePoint* koja briše tačku sa zadatim rednim brojem. Podrazumeva se da ni jedna duž ne koristi tačku koja se briše.
(5 poena)

 - d. Napisati funkciju *Merge* koji vrši "stapanje" svih tačaka koje su na manjem međusobnom rastojanju od zadatog. Svake dve tačke čije je međusobno rastojanje manje od zadatog "stapaju" se u jednu tačku čije su koordinate aritmetička sredina koordinata ovih dveju tačaka. U svim dužima koje su sadržale neku od dveju uništenih tačaka potrebno je izvršiti zamenu ovih tačaka novonastalom tačkom.
(7 poena)

 - e. Napisati program u kome se zadaju naziv datoteke u kojoj se nalazi crtež i minimalno rastojanje između tačaka koje treba "stopiti". Pozivanjem funkcija *Read* i *Merge* izvršiti učitavanje crteža iz datoteke i stapanje tačaka između kojih je rastojanje manje od zadatog.
(3 poena)

U nastavku su data neka od rešenje studenata.

Rešenje 1

(Stefan Nikolić 60/08)

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

struct tacka
{
    int id;
    float x,y;
};

struct duz
{
    int id;
    struct tacka *t1;
    struct tacka *t2;
};

struct tacka *niztacaka;
struct duz *nizduzi;
int m,n;

void ReadPoints(FILE *f)
{
    int i;

    fscanf(f,"%d",&n);
    niztacaka=(struct tacka *)malloc(n*sizeof(struct tacka));

    for (i=0;i<n;i++)
    {
        fscanf(f,"%d",&niztacaka[i].id);
        fscanf(f,"%f",&niztacaka[i].x);
        fscanf(f,"%f",&niztacaka[i].y);
    }
}

struct tacka *FindPoint(int id)
{
    int i;
    for (i=0;i<n;i++)
        if (niztacaka[i].id==id) return &niztacaka[i];
}

void ReadLines(FILE *f)
{
    int i;
    int pom;

    fscanf(f,"%d",&m);
    nizduzi=(struct duz *)malloc(m*sizeof(struct duz));

    for (i=0;i<m;i++)
    {
        fscanf(f,"%d",&nizduzi[i].id);
        fscanf(f,"%d",&pom);
        nizduzi[i].t1=FindPoint(pom);
        fscanf(f,"%d",&pom);
        nizduzi[i].t2=FindPoint(pom);
    }
}
```

```

    }
}

void ReadDrawing(FILE *f)
{
    ReadPoints(f);
    ReadLines(f);
}

void DeletePoint(int id)
{
    int i;
    int j;

    for (i=0;i<n;i++)
        if (niztacaka[i].id==id) break;

    for (;i<n;i++)
    {
        niztacaka[i]=niztacaka[i+1];

        for (j=0;j<m;j++)
        {
            if (nizduzi[j].t1->id==niztacaka[i].id) nizduzi[j].t1-=1;
            if (nizduzi[j].t2->id==niztacaka[i].id) nizduzi[j].t2-=1;
        }
    }
}

float Dist(float x1,float y1,float x2,float y2)
{
    return sqrt((x1-x2)*(x1-x2)-(y1-y2)*(y1-y2));
}

void Merge(float rast)
{
    int i;int j;int k;

    for (i=0;i<n-1;i++)
    {
        for (j=i+1;j<n;j++)
        {
            if (Dist(niztacaka[i].x,niztacaka[i].y,niztacaka[j].x,niztacaka[j].y)<rast)
            {
                niztacaka[i].x=(niztacaka[i].x+niztacaka[j].x)/2;
                niztacaka[i].y=(niztacaka[i].y+niztacaka[j].y)/2;

                for (k=0;k<m;k++)
                {
                    if (nizduzi[k].t1->id==niztacaka[j].id) nizduzi[k].t1=&niztacaka[i];
                    if (nizduzi[k].t2->id==niztacaka[j].id) nizduzi[k].t2=&niztacaka[i];
                }
                DeletePoint(niztacaka[j].id);
            }
        }
    }
}

main ()
{
    FILE *f; char s[50]; float rast; int i;

```

```

scanf("%s",&s);
f=fopen(s,"r");
ReadDrawing(f);
fscanf(f,"%f",&rast);
Merge(rast);
}

```

Rešenje 2

(Slaviša Krstić 70/08)

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

typedef struct
{
    int ID;
    float x;
    float y;
} Tacka;

```

```

typedef struct
{
    int ID;
    Tacka *X;
    Tacka *Y;
} Duz;

```

```

Tacka *FindPoint(int p,Tacka *NizT,int a)
{
    int i;

    for(i=0;i<a;i++)
        if(p == NizT[i].ID) return &NizT[i];
}

```

```

void ReadPoint(FILE *f,Tacka **NizT,int *n)
{
    int i;
    Tacka *pom;

    fscanf(f,"%d",n);
    pom = (Tacka *)malloc((*n)*sizeof(Tacka));

    for(i=0;i<*n;i++)
    {
        fscanf(f,"%d",&pom[i].ID);
        fscanf(f,"%f",&pom[i].x);
        fscanf(f,"%f",&pom[i].y);
    }

    *NizT = pom;
}

```

```

void ReadLine(FILE *f,Duz **NizD,int *b,Tacka *NizT,int a)
{
    Duz *pom;
    int i,p,j=0;

    fscanf(f,"%d",b);

```

```

pom = (Duz *)malloc((*b)*sizeof(Duz));

for(i=0;i<*b;i++)
{
    fscanf(f,"%d",&pom[i].ID);
    fscanf(f,"%d",&p);
    pom[i].X = FindPoint(p,NizT,a);
    fscanf(f,"%d",&p);
    pom[i].Y = FindPoint(p,NizT,a);
}

*NizD = pom;
}

void ReadDrawing(char *file,Tacka **NizTacaka,Duz **NizDuzi,int *n,int *m)
{
    FILE *f=fopen(file,"r");
    int a,b;
    Tacka *NizT;
    Duz *NizD;

    ReadPoint(f,&NizT,&a);
    *NizTacaka = NizT;
    *n = a;

    ReadLine(f,&NizD,&b,NizT,a);
    *NizDuzi = NizD;
    *m = b;

    fclose(f);
}

void Redirect(Tacka *NizT,Duz *NizD,int m,int n)
{
    int i,j;

    for(i=0;i<m;i++)
    {
        if (NizD[i].X->ID==NizT[n].ID) (NizD[i].X)--;
        if (NizD[i].Y->ID==NizT[n].ID) (NizD[i].Y)--;
    }
}

void DeletePoint(int tacka,Tacka *NizT,Duz *NizD,int *n,int m)
{
    int i;
    for(i=0;i<*n;i++)
        if(NizT[i].ID == tacka) break;

    for(;i<(*n)-1;i++)
        NizT[i] = NizT[i+1];

    (*n)=(*n)-1;
    Redirect(NizT,NizD,m,*n);
}

void Merge(Tacka *NizT,Duz *NizD,int *n,int m,float k)
{
    int i,j,l;

    for(i=0;i<(*n)-1;i++)

```

```

        for(j=i+1;j<*n;j++)
            if((sqrt((NizT[i].x - NizT[j].x)*(NizT[i].x - NizT[j].x))+((NizT[i].y -
NizT[j].y)*(NizT[i].y - NizT[j].y))) <= k)
                {
                    NizT[i].x = (NizT[i].x + NizT[j].x)/2;
                    NizT[i].y = (NizT[i].y + NizT[j].y)/2;

                    for(l=0;l<m;l++)
                        {
                            if(NizD[l].X->ID == NizT[j].ID) NizD[l].X = &NizT[i];
                            if(NizD[l].Y->ID == NizT[j].ID) NizD[l].Y = &NizT[i];
                        }
                    DeletePoint(NizT[j].ID,NizT,NizD,n,m);
                }
    }

int main(int argc,char *argv[])
{
    if(argc != 2)
        {
            printf("GRESKA");
            return 1;
        }

    int n,m;
    float k;
    Tacka *NizTacaka;
    Duz *NizDuzi;

    ReadDrawing(argv[1],&NizTacaka,&NizDuzi,&n,&m);

    printf("Unesite rastojanje: ");
    scanf("%f",&k);
    Merge(NizTacaka,NizDuzi,&n,m,k);

    return 0;
}

```