Министерство образования и науки Российской Федерации

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

*Кафедра МО ЭВМ*

Дисциплина « Алгоритмы и структуры данных»

**Отчет по лабораторной работе №4**

Факультет: КТИ

Группа: 1381

Выполнил: Козицкий Д.Н.

Проверил: Казаков Б.Б.

Санкт-Петербург

2013

**Задание:**

1. Реализовать один из трех алгоритмов построения минимального остовного дерева заданного графа:

2. ближайшего соседа [Prim; Dijkstra],

С: визуализаця алгоритма (важен не размер задачи и время счета, а

степень понятности и комфортности работы с программой);

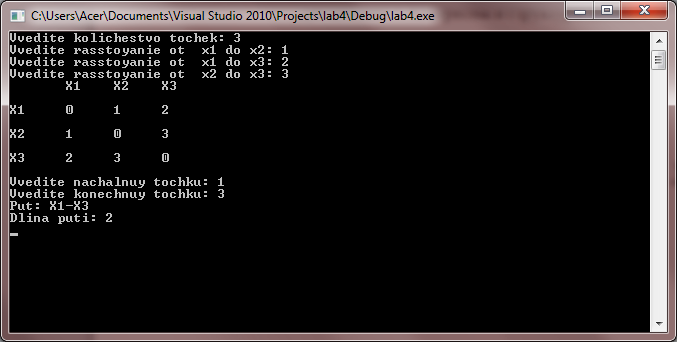
**Содержательная постановка задачи:**

1. Основной задачей является программная реализация алгоритма поиска кратчайшего пути между двумя любыми вершинами графа.
2. Программа должна работать так, чтобы пользователь вводил количество вершин и длины рёбер графа, а после обработки этих данных на экран выводился кратчайший путь между двумя заданными вершинами и его длина.

**Описание алгоритма сортировки слиянием:**

1. Программа выводит минимальный путь между двумя указанными вершинами в графе и его длину.
2. При запуске программы на экран выводится запрос о вводе весов рёбер исследуемого графа. Данные, введённые пользователем, отображаются в виде матрицы смежности, в которой не существующие рёбра обозначаются нулями. После указанным рёбрам присваивается значение 65535, которое принимается за бесконечность.
3. Следующим этапом выполнения программы является запрос о вводе номеров вершин, между которыми необходимо узнать путь. В случае, если начальная и конечная вершины совпадают, отображается соответствующее сообщение и работа программы завершается. В противном случае выполняется непосредственно алгоритм Дейкстры.
4. Результатом программы является вывод на экран вершин, через которые проходит минимальный путь, а также вывод длины маршрута. Если пути между заданными точками не существует – выводится соответствующее сообщение.

**Пример работы программы:**



**Код программы:**

//lab4.cpp

#include "stdafx.h"

#include<iostream>

#include<string.h>

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#define word unsigned int

using namespace std;

int i, j, n, p, xn, xk;

int flag[11];

word c[11][11], l[11];

char s[80], path[80][11];

int min(int n)

{

int i, result;

for(i=0;i<n;i++)

if(!(flag[i])) result=i;

for(i=0;i<n;i++)

if((l[result]>l[i])&&(!flag[i])) result=i;

return result;

}

word minim(word x, word y)

{

if(x<y) return x;

return y;

}

void main()

{

cout<<"Vvedite kolichestvo tochek: ";

cin>>n;

for(i=0;i<n;i++)

for(j=0;j<n;j++) c[i][j]=0;

for(i=0;i<n;i++)

for(j=i+1;j<n;j++)

{

cout<<"Vvedite rasstoyanie ot x"<<i+1<<" do x"<<j+1<<": ";

cin>>c[i][j];

}

cout<<" ";

for(i=0;i<n;i++) cout<<" X"<<i+1;

cout<<endl<<endl;

for(i=0;i<n;i++)

{

printf("X%d",i+1);

for(j=0;j<n;j++)

{

printf("%6d",c[i][j]);

c[j][i]=c[i][j];

}

printf("\n\n");

}

for(i=0;i<n;i++)

for(j=0;j<n;j++)

if(c[i][j]==0) c[i][j]=65535; //бесконечность

cout<<"Vvedite nachalnuy tochku: ";

cin>>xn;

cout<<"Vvedite konechnuy tochku: ";

cin>>xk;

xk--;

xn--;

if(xn==xk)

{

cout<<"Nachalnaya I konechnaya tochki sovpadayt."<<endl;

getch();

return;

}

for(i=0;i<n;i++)

{

flag[i]=0;

l[i]=65535;

}

l[xn]=0;

flag[xn]=1;

p=xn;

itoa(xn+1,s,10);

for(i=1;i<=n;i++)

{

strcpy(path[i],"X");

strcat(path[i],s);

}

do

{

for(i=0;i<n;i++)

if((c[p][i]!=65535)&&(!flag[i])&&(i!=p))

{

if(l[i]>l[p]+c[p][i])

{

itoa(i+1,s,10);

strcpy(path[i+1],path[p+1]);

strcat(path[i+1],"-X");

strcat(path[i+1],s);

}

l[i]=minim(l[i],l[p]+c[p][i]);

}

p=min(n);

flag[p]=1;

}

while(p!=xk);

if(l[p]!=65535)

{

cout<<"Put: "<<path[p+1]<<endl;

cout<<"Dlina puti: "<<l[p]<<endl;

}

else

cout<<"takogo puti ne syshestvuet!"<<endl;

getch();

}

**Вывод:**

В ходе выполнения данной лабораторной работы был освоен алгоритм сортировки Дейкстры и написана программа, которая, используя данный алгоритм, выводит на экран вершины, через которые проходит минимальный путь, а также длину маршрута.