Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

Звіт

про виконання лабораторної роботи №7

з дисципліни «Проектний практикум»

|  |  |
| --- | --- |
| Перевірив:  Асистент кафедри ПЗАС  Півень О.Б. | Виконав:  студент 3-го курсу  групи ПЗ-1644  Близнюк А. О. |

Черкаси 2017

**Тема роботи:** визначення найкоротших шляхів матричним метофом і методом Флойда.

**Мета роботи**: Ознайомлення з методами визначення найкоротших шляхів між вершинами графа.

**Теоретичні відомості**

Граф — це сукупність об'єктів із зв'язками між ними.

Об'єкти розглядаються як вершини, або вузли графу, а зв'язки — як дуги, або ребра. Для різних областей використання види графів можуть відрізнятися орієнтовністю, обмеженнями на кількість зв'язків і додатковими даними про вершини або ребра.

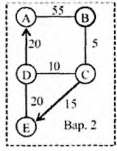
Велика кількість структур, які мають практичну цінність в [математиці](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) та [інформатиці](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), можуть бути представлені графами.

Якщо задано граф G = (V, E) та початкову вершину s, алгоритм пошуку в ширину систематично обходить всі досяжні із s вершини. На першому кроці вершина s позначається, як пройдена, а в список додаються всі вершини, досяжні з s без відвідування проміжних вершин. На кожному наступному кроці всі поточні вершини списку відмічаються, як пройдені, а новий список формується із вершин, котрі є ще не пройденими сусідами поточних вершин списку. Для реалізації списку вершин найчастіше використовується [черга](https://uk.wikipedia.org/wiki/%D0%A7%D0%B5%D1%80%D0%B3%D0%B0_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)). Виконання алгоритму продовжується до досягнення шуканої вершини або до того часу, коли на певному кроці в список не включається жодна вершина. Другий випадок означає, що всі вершини, доступні з початкової, уже відмічені, як пройдені, а шлях до цільової вершини не знайдений.

Алгоритм має назву пошуку в ширину, оскільки «фронт» пошуку (між пройденими та непройденими вершинами) одноманітно розширюється вздовж всієї своєї ширини. Тобто, алгоритм проходить всі вершини на відстані k перед тим як пройти вершини на відстані k+1.

**Завдання:**

Створити програму, яка реалізує пошук найкоротшого шляху між двома довільними вершинами графа згідно методу Флойда. Номери вершин для пошуку шляху між ними користувач повинен ввести самостійно.



#include <iostream>

#include <iomanip>

#define MaxNodes 5

#define B 1000

using namespace std;

class Warshall

{

private:

unsigned Adj[MaxNodes][MaxNodes];

unsigned C[MaxNodes][MaxNodes];

public:

int start, finish;

void Vvod();

void scheme();

void MinDlin();

void Vyvod();

void VyvodValue();

};

int mg[MaxNodes][MaxNodes]={

{0, 55, 0, 0, 0},

{55, 0, 5, 0, 0},

{0, 5, 0, 10, 15},

{20, 0, 10, 0, 20},

{0, 0, 0, 20, 0}

};

void Warshall::scheme(){

cout << setw(4) << "A" << setw(8) << " B " << setw(8) << " C " << setw(8) << " D " << setw(8) << " E " << endl;

for(int i=0;i<MaxNodes;i++){

for(int j=0;j<MaxNodes;j++){

cout << setw(4) <<Adj[i][j]<<" \t";

}

cout<<endl;

}

}

void Warshall::Vvod()

{

for (int i=0;i<MaxNodes;i++){

for (int j=0;j<MaxNodes;j++)

{

Adj[i][j] = mg[i][j];

if (Adj[i][j] == 0) C[i][j] = B;

else C[i][j] = Adj[i][j];

}

}

}

void Warshall::MinDlin()

{

for (int k=0;k<MaxNodes;k++)

for (int i=0;i<MaxNodes;i++)

for (int j=0;j<MaxNodes;j++)

if (C[i][j] > C[i][k] + C[k][j]) C[i][j] = C[i][k] + C[k][j];

}

void Warshall::Vyvod()

{

cout << "Matrix minimum ways:\n";

cout << setw(4) << "A" << setw(4) << " B " << setw(4) << " C " << setw(4) << " D " << setw(4) << " E " << endl << endl;

for (int i=0;i<MaxNodes;i++)

{

for (int j=0;j<MaxNodes;j++){

cout << setw(4) << C[i][j];

}

cout << endl;

}

}

void Warshall::VyvodValue()

{

cout << "min way: ";

cout << C[start][finish] << " ";

cout << endl;

}

int main()

{

setlocale(0, "");

Warshall A;

A.Vvod();

cout<<"Zvagenyi graph:"<<endl;

A.scheme();

A.MinDlin();

cout<<endl;

A.Vyvod();

while(true){

cout<<endl;

cout<<"start: ";

cin>>A.start;

A.start = A.start - 1;

cout<<"finish: ";

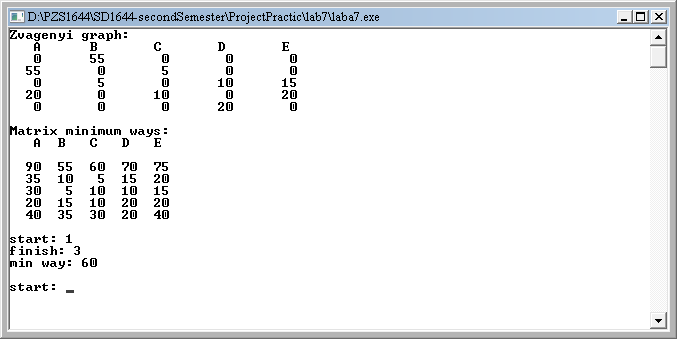
cin>>A.finish;

A.finish = A.finish - 1;

A.VyvodValue();

}

}



Висновки: Під час лабораторної роботи було освоєно навички роботи з графами та алгоритм пошуку за методом Флойда.