Міністерство освіти і науки України Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт

з лабораторної роботи №2

з дисципліни «Дискретна математика»

Виконав: Перевірив:

ITIHФ-20-1 Асистент

Самченко С. А. Бродецький Ф.А.

Харків

2021

**Мета:** навчитися реалізоувати алгоритм Дейксти програмним шляхом.

**Завдання:** написати програму, що реалізовує алгоритм Дейкстри, знайти найкоротші шляхи на графі від заданої вершини до решти. Для реалізації програми необхідно вихідний граф представити у вигляді матриці ваг.

Вариант 21 вершина А

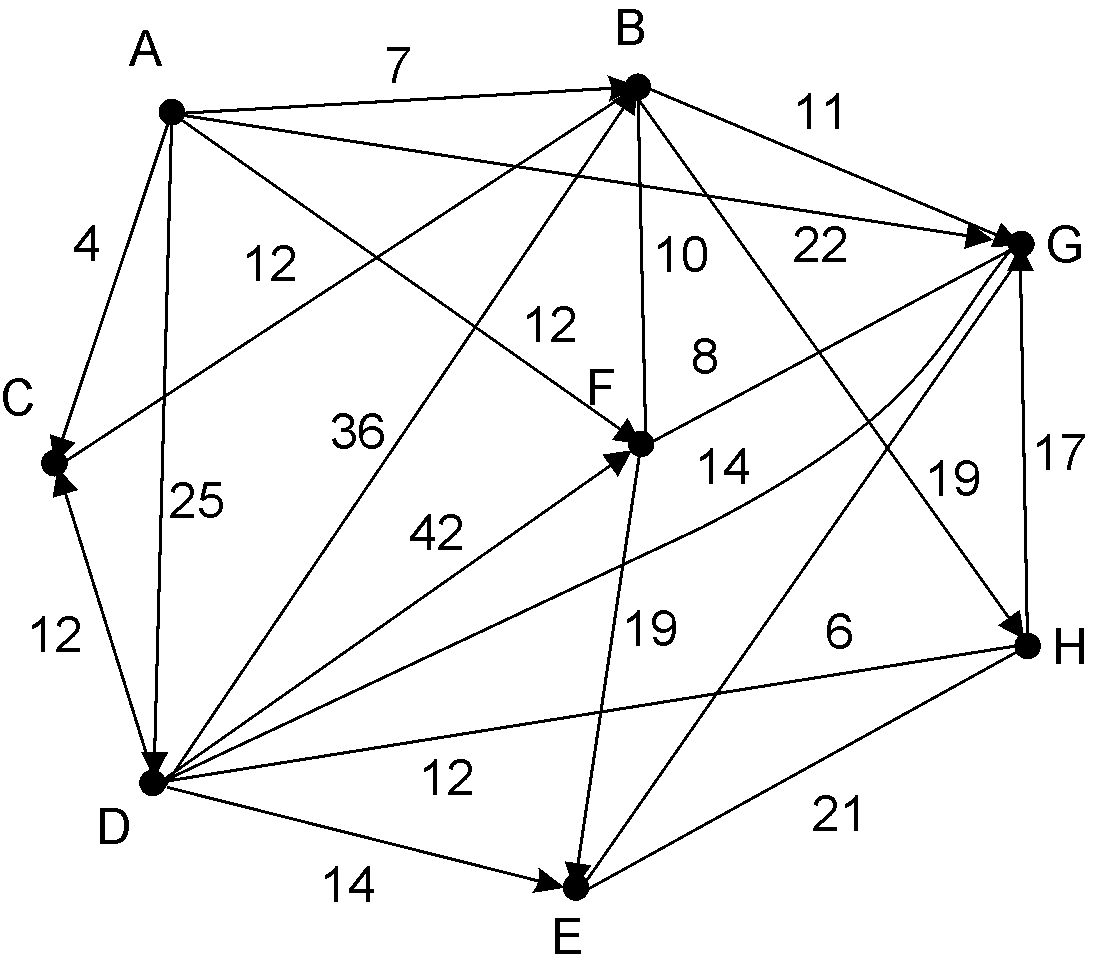


Рисунок 1

**Реалізація:**

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

struct Graph {

int start = 0;

int end = 0;

int weight = 0;

};

int main() {

setlocale(LC\_ALL, "");

const int couple(30), VertNumb(8);

int a, inf(1000), start, end;

Graph graph[couple] = {

{ 1, 2, 7}, {1, 3, 4}, {1, 4, 25}, {1, 6, 12}, {1, 7, 22}, { 2, 3, 12}, {2, 6, 10}, {2, 8, 19}, {2, 7, 11},

{ 3, 2, 12}, { 4, 3, 12}, {4, 2, 36}, {4, 6, 42}, {4, 7, 14}, {4, 8, 6}, {4, 5, 14}, { 5, 7, 19}, {5, 8, 21},

{ 6, 2, 10}, {6, 5, 8}, {6, 5, 19}, { 7, 6, 8}, {7, 4, 14}, {7, 5, 6}, { 8, 4, 6}, {8, 5, 21}, {8, 7, 17}

};

int matrix[VertNumb][VertNumb] = {};

cout << "Матрица весов:\n";

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

matrix[graph[i].start - 1][graph[i].end - 1] = graph[i].weight;

}

int x[100], path[100], Vert[100];

string word = "ABCDEFGH";

cout << " ";

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

cout << setw(4) << word[i];

}

cout << endl;

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

{

cout << " " << word[i];

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

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

}

cout << endl;

}

cout << endl;

char enter;

N: cout << ("введите вершину (A - H): ");

cin >> enter;

switch (enter) {

case 'A': start = 1; break;

case 'B': start = 2; break;

case 'C': start = 3; break;

case 'D': start = 4; break;

case 'E': start = 5; break;

case 'F': start = 6; break;

case 'G': start = 7; break;

case 'H': start = 8; break;

default : start = 9; break;

}

if (start > (VertNumb - 1) && start < 0) { cout << ("в данном графе нет такой вершины") << endl; goto N; }

start = start - 1;

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

end = i;

if (end == start) continue;

else {

int b;

for (b = 0; b < VertNumb; b++) {

path[b] = inf;

x[b] = 0;

}

Vert[start] = 0;

path[start] = 0;

x[start] = 1;

a = start;

while (1) {

for (b = 0; b < VertNumb; b++) {

if (matrix[a][b] == 0)continue;

if (x[b] == 0 && path[b] > path[a] + matrix[a][b]) {

path[b] = path[a] + matrix[a][b];

Vert[b] = a;

}

}

int w = inf;

a = -1;

for (b = 0; b < VertNumb; b++) {

if (x[b] == 0 && path[b] < w) {

a = b;

w = path[b];

}

}

if (a == -1) {

cout << ("нет пути из ") << word[start]; cout << (" в ") << word[end] << endl;

break;

}

if (a == end) {

cout << ("кратчайший пути из ") << word[start]; cout << (" в ") << word[end] << ":";

b = end;

while (b != start) {

cout << " " << b + 1;

b = Vert[b];

}

cout << " " << start + 1 << (" \tдлина пути = ") << path[end]; cout << endl;

break;

}

x[a] = 1;

}

}

}

system("pause");

return 0;

}

**Результат роботи:**

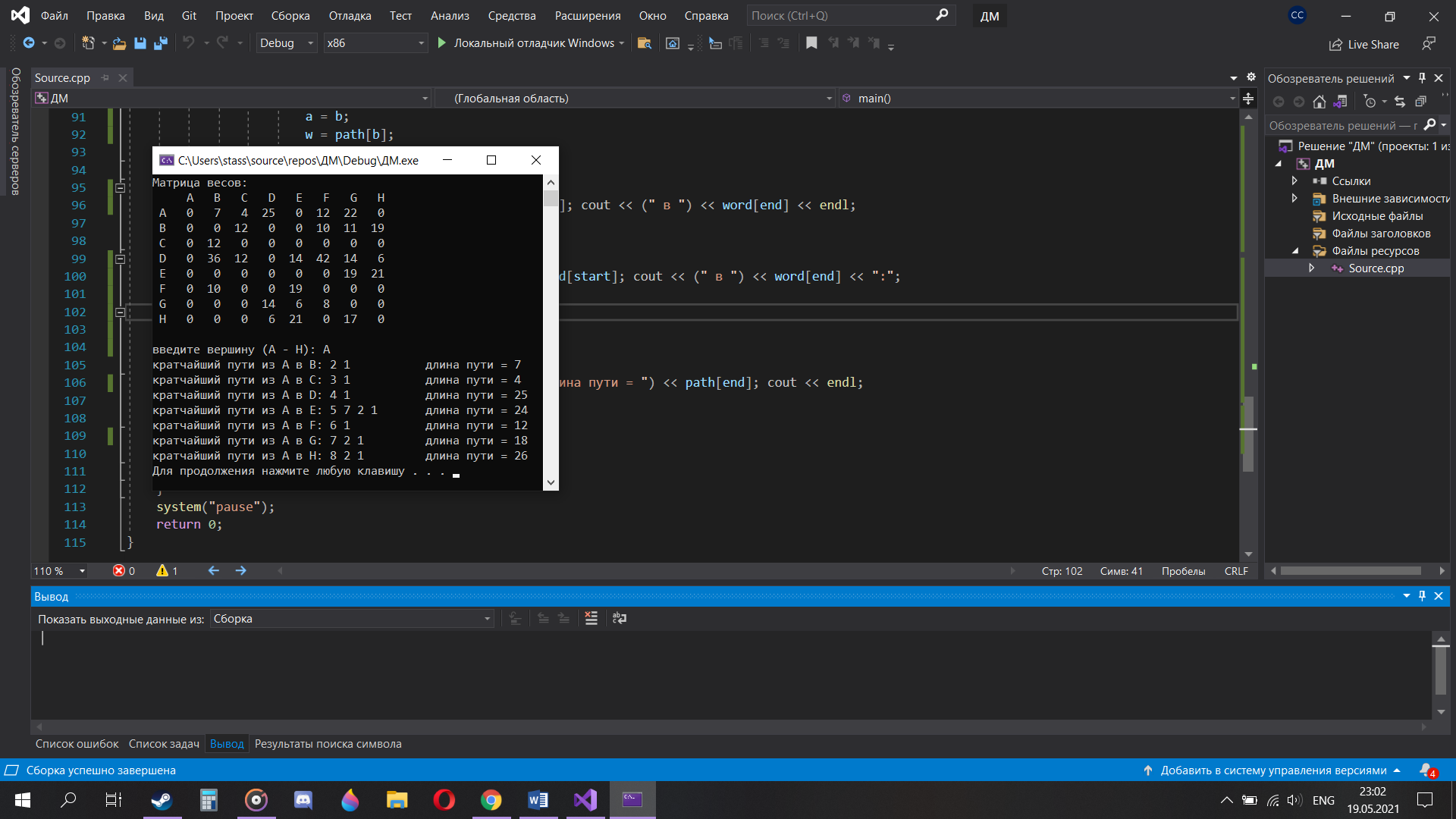


Рисунок 2

**Висновок:** В цій лабораторній роботі, за допомогою мови програмування С++, отримано навички побудови матриці ваг, яка представляє собою граф (рисунок 1) у матричному вигляді, завдяки якій можна знайти найкоротший шлях з будь-якої обраної вершини графа до всіх інших вершин.