МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий Государственный Технический Университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ИДЗ

по технологиям программирования

# “Алгоритм Дейкстры”

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Станиславчук С.М.

(подпись, дата)

Группа АС-21-1

Руководитель

Доцент \_\_\_\_\_\_\_\_\_\_\_\_ Хромов А. М.

(подпись, дата)

Липецк 2022 г.

**Содержание:**

1. Титульный лист.

2. Цель работы.

3. Блок-схема алгоритма разработанной программы

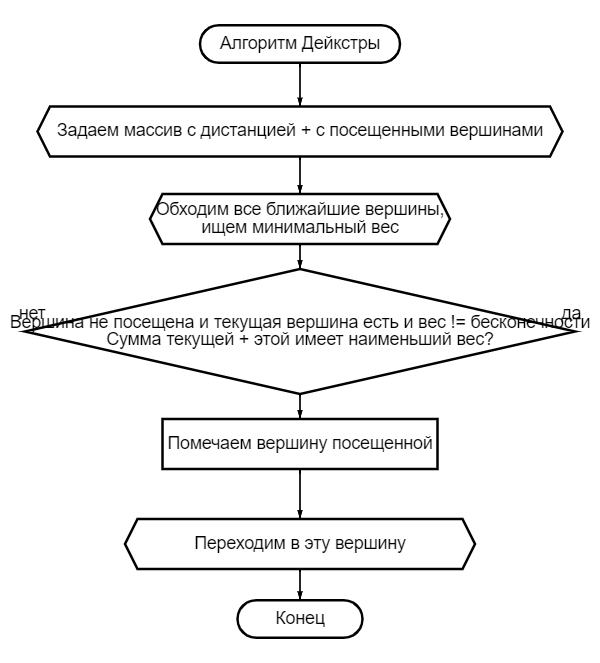
4. Текст программы

5. Контрольный пример

6. Выводы по работе.

**2. Цель работы**

Написать программу, реализующую метод обхода графа алгоритмом Дейкстры.

**3. Блок-схема разработанной программы**

**4. Текст программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <fstream>

using namespace std;

const int V = 4; // Тут задаётся размерность матрицы

int GR[V][V];

//алгоритм Дейкстры

void Dijkstra(int GR[V][V], int st)

{

int distance[V], count, index, i, u, m = st + 1;

bool visited[V];

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

{

distance[i] = INT\_MAX; visited[i] = false;

}

distance[st] = 0;

for (count = 0; count < V - 1; count++)

{

int min = INT\_MAX;

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

if (!visited[i] && distance[i] <= min)

{

min = distance[i]; index = i;

}

u = index;

visited[u] = true;

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

if (!visited[i] && GR[u][i] && distance[u] != INT\_MAX &&

distance[u] + GR[u][i] < distance[i])

distance[i] = distance[u] + GR[u][i];

}

cout << "Стоимость пути из начальной вершины до остальных:\t\n";

for (i = 0; i < V; i++) if (distance[i] != INT\_MAX)

cout << m << " > " << i + 1 << " = " << distance[i] << endl;

else cout << m << " > " << i + 1 << " = " << "маршрут недоступен" << endl;

}

void customGenerate() {

cout << "Введите матрицу веса графа " << "(n = " << V << ")\n";

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

{

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

cin >> GR[i][j];

}

}

}

void randomGenerate() {

cout << "Рандомная матрица веса графа " << "(n = " << V << ")\n";

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

{

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

GR[i][j] = 0 + rand() % 11;

cout << GR[i][j] << ", ";

}

cout << endl;

}

}

void readfile() {

FILE\* f;

f = fopen("file.txt", "r");

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

{

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

fscanf(f, "%d", &GR[i][j]);

}

fclose(f);

}

void showGR() {

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

{

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

cout << GR[i][j] << ", ";

}

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "Rus");

int start; size\_t number;

srand(time(0));

while (1) {

cout << "Как ввести матрицу смежности?";

cout << "\n1. Кастомный ввод";

cout << "\n2. Рандомный ввод";

cout << "\n3. Прочесть из файла";

cout << "\n0. Выход из программы\n";

cin >> number;

if (number == 1) {

customGenerate();

showGR();

cout << "\nНачальная вершина >> "; cin >> start;

Dijkstra(GR, start - 1);

}

if (number == 2) {

randomGenerate();

cout << "\nНачальная вершина >> "; cin >> start;

Dijkstra(GR, start - 1);

}

if (number == 3) {

readfile();

showGR();

cout << "\nНачальная вершина >> "; cin >> start;

Dijkstra(GR, start - 1);

}

if (number == 0)

break;

}

system("pause");

return 0;

}

5. Контрольный пример

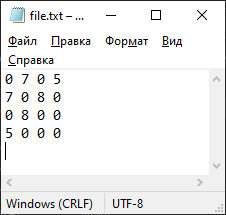
При старте программа просит нас выбрать одну из трех функций ввода матрицы смежности. Выберем кастомный ввод.



Рандомный ввод:



Чтение из файла:





Как видим, программа корректно работает.

**Вывод:** написал программу, реализующую метод обхода графа алгоритмом Дейкстры.