Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

по лабораторной работе №7

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний во взвешенном графе»

Выполнили:

студенты группы 20ВВ2

Киреев Б.П.

Верховский М.В.

Лукин В.Д.

Приняли:

д.т.н. Митрохин М.А.

к.т.н. доцент Юрова О.В.

Пенза 2021

Название: Поиск расстояний во взвешенном графе.

Цель работы: Научиться осуществлять поиск расстояний во взвешенном графе.

Лабораторное задание:

**Задание 1:**

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного взвешенного графа G. Выведите матрицу

на экран.

2. Для сгенерированного графа осуществите процедуру поиска

расстояний, реализованную в соответствии с приведенным выше описанием.

При реализации алгоритма в качестве очереди используйте класс queue из

стандартной библиотеки С++.

**Задание 1:**

Листинг:

Файл Lab6.cpp

#include <time.h>

#include <malloc.h>

#include <locale.h>

#include <iostream>

#include <queue>

#include <ctime>

#include <conio.h>

using namespace std;

int i, num = 0;

int\* vis = NULL, n, \*\* p = NULL;

int pogr = 0;

int Levo = 0, Control = 0;

//Поиск уровней

void BFSD(int v, int\*\* p) {

queue <int> q;

q.push(v);

vis[v] = 0;

while (!q.empty()) {

v = q.front();

q.pop();

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

if (vis[i] > vis[v] + p[i][v] && p[i][v] != 0) {

q.push(i);

vis[i] = vis[v] + p[i][v];

}

}

}

cout << endl;

}

int main() {

setlocale(LC\_ALL, "Rus");

srand(time(0));

cout << "Введите размерность графа:";

cin >> n;

vis = (int\*)calloc(n, 3);

p = (int\*\*)calloc(n, 3);

cout.precision(3 \* n);

vis = (int\*)malloc(sizeof(int) \* n);

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

p[i] = (int\*)calloc(n, 3);

vis[i] = 0;

}

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

for (int m = 0 + pogr; m < n; m++) {

p[i][m] = 0 + rand() % 11;

p[m][i] = p[i][m];

p[i][i] = 0;

}

pogr++;

}

cout << "\nМатрица 1:" << endl;

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

for (int m = 0; m < n; m++) { cout << p[i][m] << " "; }

cout << endl;

}

for (int i = 0; i < n; i++) { vis[i] = 1000; } //Обнуление проходимых вершин

BFSD(0, p);

while (Control != n) {

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

if (vis[i] == Levo) {

cout << "Вершина " << i + 1 << " находится от вершины 1 на расстоянии " << Levo << endl;

Control++;

}

}

Levo++;

}

cout << endl;

return 0;

}

Пояснительный текст к программе:

Данная программа осуществляет поиск расстояний во взвешенном графе. Работа алгоритма продолжается до тех пор, пока существуют непосещённые вершины.

Результаты работы программы:

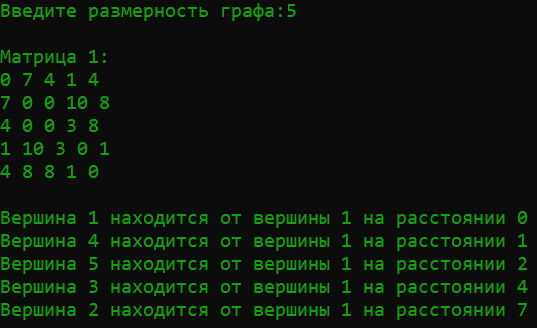


Рисунок №1 – Результат работы программы.

Вывод: При выполнении данной лабораторной работы мы смогли научиться осуществлять поиск расстояний во взвешенном графе.