Министерство высшего образования и науки Российской Федерации

Пензенский Государственный Университет

Кафедра «Вычислительная техника»

ОТЧЕТ

Лабораторная работа №6

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Простые структуры данных»

Выполнил студент группы 20ВВ2:

Пантелеев И.А.

Приняли:

Митрохин М. А.

Юрова О. В.

Пенза 2021

**Лабораторная работа №6**

**Поиск расстояний в графе**

**Задание**

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного графа G. Выведите матрицу на экран.

2. Для сгенерированного графа осуществите процедуру поиска

расстояний.

При реализации алгоритма в качестве очереди используйте класс queue из

стандартной библиотеки С++.

**Практическая часть**

#include <iostream>

#include <queue>

using namespace std;

const int n = 6;

int i, j;

int\* DIST = new int[n];

int G[n][n];/\* = {

{0,1,1,0,0,1},

{1,0,0,1,0,0},

{1,0,0,0,1,0},

{0,1,0,0,0,0},

{0,0,1,0,0,0},

{1,0,0,0,0,0}

};\*/

//поиск в ширину

void BFSD(int v)

{

queue <int> Q;

Q.push(v);

DIST[v] = 0;

while (!Q.empty())

{

v = Q.front();

Q.pop();

cout << v + 1 << " ";

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

if (G[v][i] && DIST[i]==-1)

{

Q.push(i);

DIST[i] = DIST[v]+1;

}

}

}

//главная функция

void main()

{

srand(time(NULL));

setlocale(LC\_ALL, "Rus");

int start;

cout << "Стартовая вершина >> "; cin >> start;

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

{

DIST[i] = -1;

G[i][i] = 0;

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

{

G[j][i] = G[i][j] = rand() % 2; //заполнение графа случайными числами

}

}

cout << "Матрица смежности графа: " << endl;

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

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

cout << " " << G[i][j];

cout << endl;

}

cout << "Порядок обхода: ";

BFSD(start - 1);

cout << endl <<"Расстояния до вершин: ";

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

cout << DIST[i] << " ";

}

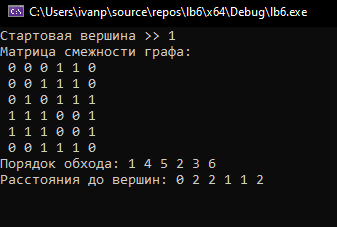
delete[]DIST;

system("pause>>void");

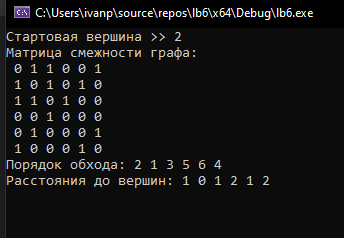
}

**Результаты**

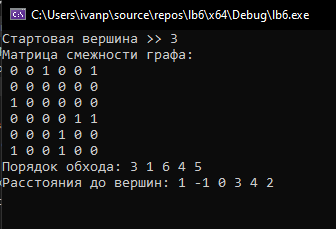
1)



2)



3)



**Вывод**

Реализованный алгоритм, является довольно эффективным способом нахождения расстояния между вершинами, так как в свою основу он берет алгоритм обхода в ширину что позволяет ему находить самый оптимальный путь до заданной вершины. Недостатком является то, что алгоритм не подходит для взвешенных и ориентированных графов.