Пензенский государственный университет

Кафедра «Вычислительной техники»

**Отчёт**

По лабораторной №6

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний в графе»

**Выполнил студент группы 20ВВ1:**

Бектемирова А.Э.

**Приняли:**

Юрова О.В.

Акифьев И.В.

Пенза 2021 г.

**Цель работы:** освоить алгоритм поиска расстояний в графе.

**Задание 1:**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного 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++.



1. \*Реализуйте процедуру поиска расстояний для графа, представленного списками смежности.

**Задание 2\*:**

1. Реализуйте процедуру поиска расстояний на основе обхода в глубину.
2. Реализуйте процедуру поиска расстояний на основе обхода в глубину для графа, представленного списками смежности.
3. Оцените время работы реализаций алгоритмов поиска расстояний на основе обхода в глубину и обхода в ширину для графов разных порядков.

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h> //Указание прототи-

#include <conio.h> //пов библиотечных

#include <math.h> //функций

#include <malloc.h>

#include <time.h>

#include <random>

#include <queue>

#include <iostream>

using namespace std;

void BFSD(int\*\* G, int\* vis, int n, int v) {

vis[v] = 0;

queue <int> q;

q.push(v);

while (!q.empty())

{

v = q.front();

q.pop();

printf("%d ", v);

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

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

q.push(i);

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

}

}

}

printf("\n");

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

printf("%d ", vis[i]);

}

void main()

{

int\*\* G;

int\* vis;

int v = 0, n;

setlocale(0, "Rus");

printf("Введите размер графа");

scanf("%d", &n);

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

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

G[i] = (int\*)malloc(n \* sizeof(int));

}

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

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

vis[i] = -1;

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

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

G[i][g] = rand() % 2;

if (i == g) G[i][g] = 0;

G[g][i] = G[i][g];

}

}

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

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

printf("%3d", G[i][g]);

}

printf("\n");

}

BFSD(G, vis, n, 0);

}

**Вывод:** в процессе выполнения лабораторной работы были получены навыки реализации алгоритма поиска расстояний в графе.