Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №6

# по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний в графе»

**Выполнили:**

студенты группы 20ВВ3

Баулин А.Д.

**Приняли:**

Юрова О.В.

Митрохин М.А.

Пенза 2021

# Название

Поиск расстояний в графе

**Цель работы** – изучение функций, предназначенных для поиска расстояний в графе.

# Лабораторное задание

Задание 1

1. Сгенерируйте (используя генератор случайных чисел) матрицу

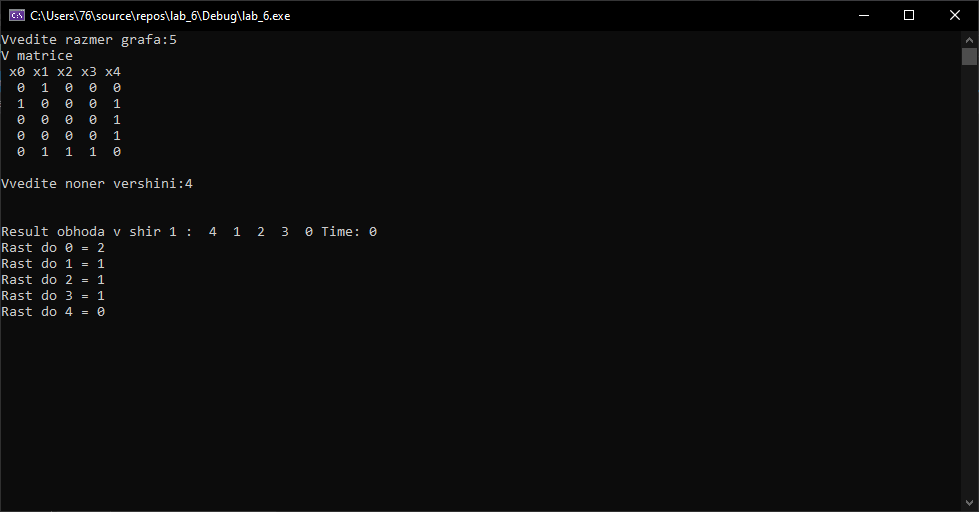
смежности для неориентированного графа G. Выведите матрицу на экран.

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++.

# Результат работы программы



**Рисунок 2 – Результат работы программы**

**Вывод**

Использование обхода в глубину для поиска расстояний не эффективно. Чем больше граф, тем менее эффективен обход в глубину.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <conio.h>

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <stdbool.h>

#include <queue>

using namespace std;

int\*\* gen\_m(int\*\* M, int n)

{

int i, j, z = 1;

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

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

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

//srand(time(NULL));

srand(12);

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

{

M[i][i] = 0;

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

{

M[i][j] = rand() % 2;

M[j][i] = M[i][j];

}

z++;

}

return M;

}

void print(int\*\* M, int n)

{

int i, j;

printf("V matrice\n");

for (i = 0; i < n; i++) printf(" x%d", i);

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

{

printf("\n");

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

{

printf("%\*d", 3, M[i][j]);

}

}

}

void ob\_m\_1(int\*\* M, int n, int k, int\* m)

{

queue <int> Q;

int i, j = k;

Q.push(k);

m[k] = 0;

while (!Q.empty())

{

j = Q.front();

printf("%\*d", 3, j);

Q.pop();

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

{

if (M[j][i] == 1 && m[i] == -1)

{

m[i] = m[j] + 1;

Q.push(i);

}

}

}

}

void hod1(int\*\* M, int n, int k)

{

int i, \* m;

clock\_t start, end;

start = clock();

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

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

{

m[i] = -1;

}

ob\_m\_1(M, n, k, m);

end = clock();

printf(" Time: %d", (end - start) / 1);

for (i = 0; i < n; i++) printf("\nRast do %d = %d", i, m[i]);

free(m);

}

void main()

{

int\*\* M = NULL, n, k, i;

printf("Vvedite razmer grafa:");

scanf("%d", &n);

M = gen\_m(M, n);

print(M, n);

printf("\n\nVvedite noner vershini:");

scanf("%d", &k);

printf("\n\nResult obhoda v shir 1 :");

hod1(M, n, k);

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

{

free(M[i]);

}

free(M);

\_getch();

}