Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

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

на тему «Обход графа в глубину»

**Выполнили:**

студенты группы 20ВВ3

Баулин А.Д.

**Приняли:**

Юрова О.В.

Митрохин М.А.

Пенза 2021

# Название

Обход графа в глубину

**Цель работы**–изучение функций, предназначенных для обхода графа в глубину.

# Лабораторное задание

Задание 1

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного графа G. Выведите матрицу на экран.

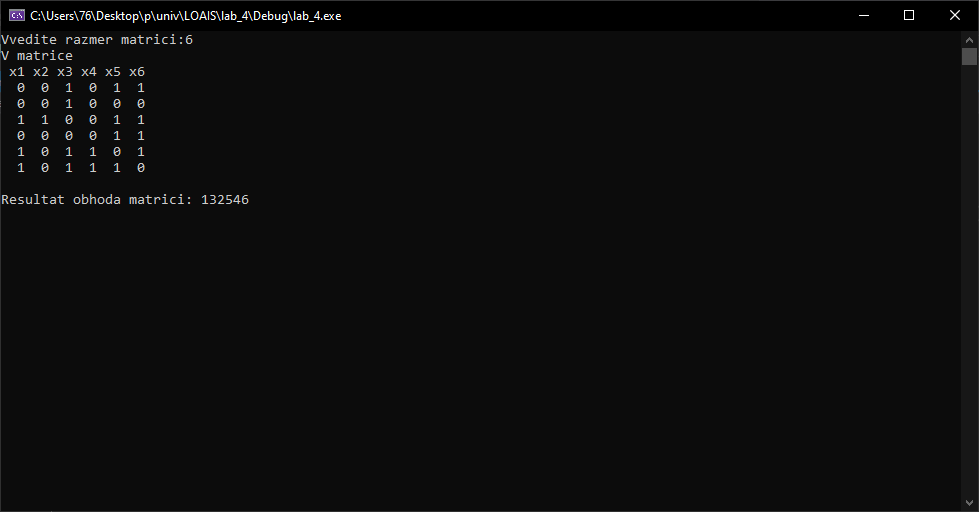
2. Для сгенерированного графа осуществите процедуру обхода в

глубину, реализованную в соответствии с приведенным выше описанием.

**Теоретические сведения**

Одним из способов обхода графов является поиск в глубину. Идея такого обхода состоит в том, чтобы начав обход из какой-либо вершины всегда переходить по первой встречающейся в процессе обхода связи в следующую вершину, пока существует такая возможность.

# Результат работы программы



**Рисунок 1 – Результат работы программы**

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <conio.h>

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <stdbool.h>

void free(int\*\* M1, bool\* M2, int n)

{

int i, j;

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

{

free(M1[i]);

}

free(M1);

free(M2);

}

void print(int\*\* M1, int n)

{

int i, j;

printf("V matrice\n");

for (i = 0; i < n; i++) printf(" x%d", i + 1);

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

{

printf("\n");

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

{

printf("%\*d", 3, M1[j][i]);

}

}

}

void hod(int\*\* M1, bool\* M2, int n, int i)

{

int j;

M2[i] = true;

printf("%d", i + 1);

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

{

if (M1[i][j] == 1 && M2[j] == false) hod(M1, M2, n, j);

}

}

void main()

{

int\*\* M1;

int i, j, n, z = 1;

bool\* M2;

system("cls");

printf("Vvedite razmer matrici:");

scanf("%d", &n);

M1 = (int\*\*)malloc(n \* sizeof(int\*));

M2 = (bool\*)malloc(n \* sizeof(bool));

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

{

M2[i] = false;

}

srand(time(NULL));

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

{

M1[i] = (int\*)malloc(n \* sizeof(int));

M1[i][i] = 0;

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

{

if (rand() % 100 > 50) M1[i][j] = 0; else M1[i][j] = 1;

}

z++;

}

z = 1;

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

{

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

{

M1[j][i] = M1[i][j];

}

z++;

}

print(M1, n);

printf("\n\nResultat obhoda matrici");

for (i = 0; i < n; i++) if (M2[i] == false) { printf(": "); hod(M1, M2, n, i); }

for (i = 0; i < n; i++) M2[i] = false;

free(M1, M2, n);

\_getch();

}