Пензенский государственный университет

Кафедра "Вычислительной техники"

**Отчёт**

По лабораторной №5

по дисциплине "Л и ОА в ИТ"

на тему "Обход графа в ширину"

***Выполнили студенты группы 20ВВ4:***

*Демаев А.И.*

*Шалеев Я.А.*

*Макеев Д.И*

***Приняли:***

*Акифьев И.В.*

*Юрова О. В.*

Пенза 2021

# Цель работы: выполнить обход графа в ширину.

**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 ширину,

реализованную в соответствии с приведенным выше описанием. При

реализации алгоритма в качестве очереди используйте класс queue из

стандартной библиотеки С++.

**Теоретическая часть:**

Обход графа в ширину – еще один распространенный способ обхода

графов.

Основная идея такого обхода состоит в том, чтобы посещать вершины

по уровням удаленности от исходной вершины. Удалённость в данном

случае понимается как количество ребер, по которым необходимо прейти до

достижения вершины.

Класс Queueпредставляет обычную очередь, работающую по алгоритму FIFO ("первый вошел - первый вышел").

У класса Queueможно отметить следующие методы:

* Dequeue: извлекает и возвращает первый элемент очереди
* Enqueue: добавляет элемент в конец очереди
* Peek: просто возвращает первый элемент из начала очереди без его удаления

**Практическаячасть:**

**Листинг:**

#define\_CRT\_SECURE\_NO\_WARNINGS

#include"locale"

#include"stdio.h"

#include"conio.h"

#include<locale.h>

#include<stdlib.h>

#include<malloc.h>

#include<iostream>

usingnamespace std;

#include<queue>;

voidBFS(intv, int\*\* matrix, intsize, int\* NUM)

{

int st = 0;

st = v;

queue<int> Q;

NUM[st] = 1;

Q.push(st);

while(!Q.empty()) {

st = Q.front();

Q.pop();

printf("%d", st);

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

{

if (matrix[st][i] == 1 &&NUM[i] == 0)

{

Q.push(i);

NUM[i] = 1;

}

}

}

}

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "RUS");

int v, n;

printf("Введите размерность матрицы: ");

scanf("%d", &n);

int\*\* M;

int\* NUM;

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

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

{

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

}

//int M[6][6];

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

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

{

NUM[i] = 0;

}

printf("Сгенерированные матрицы:");

printf("\n");

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

{

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

{

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

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

}

printf("\n");

}

printf("\n\n");

printf("Введите вершину, с которой будет начинаться обход: ");

scanf("%d", &v);

printf("\n");

printf("Результат: ");

while (NUM[v] != 1)

{

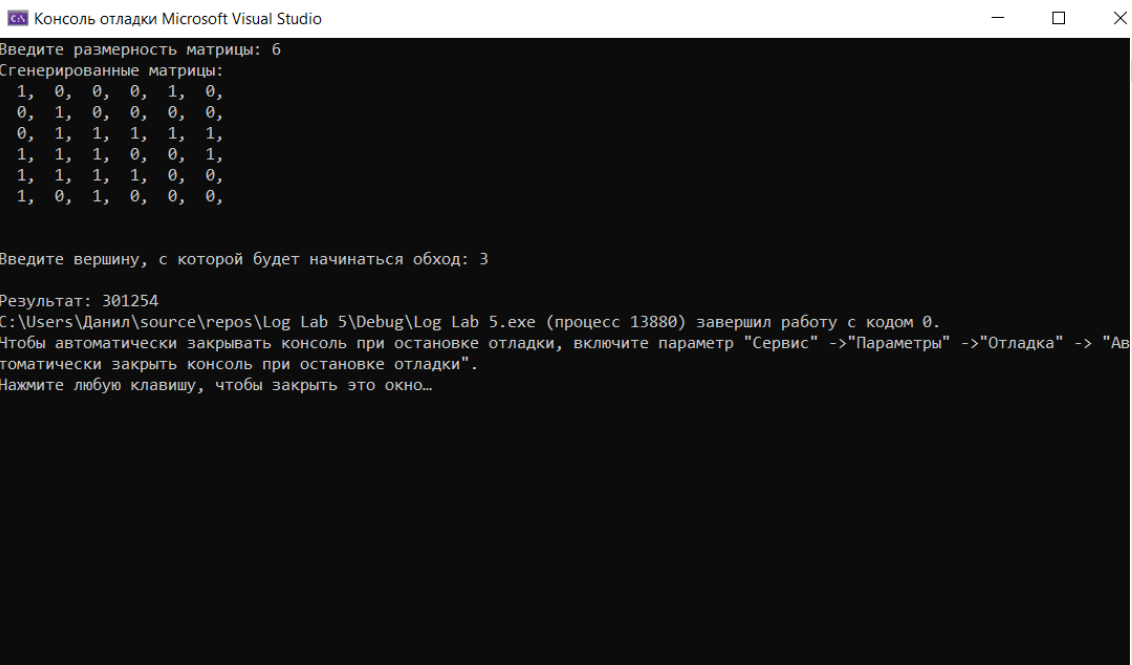
BFS(v, M, n, NUM);

}

void free(int\* NUM);

}

**Результат работы программы:**

****

**Вывод:**Научились выполнять обход графа в ширину.