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 20вв1:***

*Филиппов В.А.*

***Приняли:***

*Юрова О. В.*

*Акифьев И. 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 Queue представляет обычную очередь, работающую по алгоритму FIFO ("первый вошел - первый вышел").

У класса Queue можно 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

**Практическая часть:**

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <locale>

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <iostream>

#include <queue>

using namespace std;

int NUM[5]; // вершины графа

void BFS(int v, int\*\* matrix, int size) {

queue<int> q;

NUM[v] = 1;

q.push(v); // помещаем в очередь вершины

while (!q.empty()) { // пока очередь не пуста

v = q.front(); // извлекаем вершину

q.pop();

printf("%4i", v);

for (int i = 0; i < size; i++) { // проверяем для нее все смежные вершины

if (matrix[v][i] == 1 && NUM[i] == 0) { // если вершина смежная и не обнаружена

q.push(i); // добавляем ее в очередь

NUM[i] = 1; // отмечаем вершину как обнаруженную

}

}

}

}

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "RUS");

int v, n;

printf("Введите размерность матрицы: ");

scanf("%d", &n);

int\*\* M;

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

for (int i = 0; i < n; i++) // цикл по строкам

{

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

}

//сгенерирование матрицы

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");

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

scanf("%d", &v);

printf("\n");

printf("\n""Результат: ");

while (NUM[v] != 1)

{

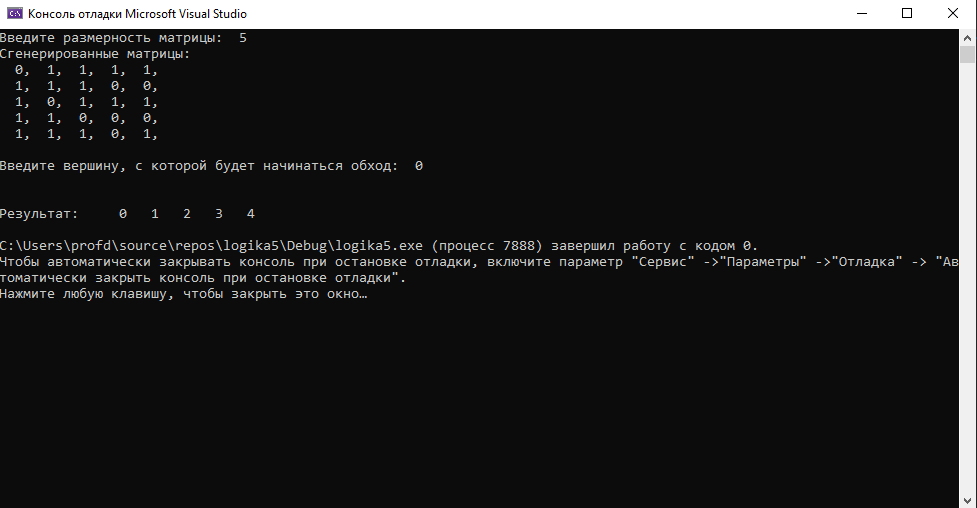
BFS(v, M, n);

}

printf("\n");

}

**Результат работы программы:**



**Вывод:** Я научился выполнять обход графа в ширину.