Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Л и ОА в ИЗ»

на тему «Обход графа в ширину»

**Выполнили:**

студенты группы 19ВВ2

Сидоров Н. Р.

Карамышев А. А.

Горбунов Д.А.

**Приняли:**

Юрова О. В.

Митрохин М. А.

Пенза 2020

**Название:**

Обход графа в ширину

**Цель работы:**

Научиться применять алгоритм рекурсивного обхода графа в ширину.

**Лабораторные задания:**

### ***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++.

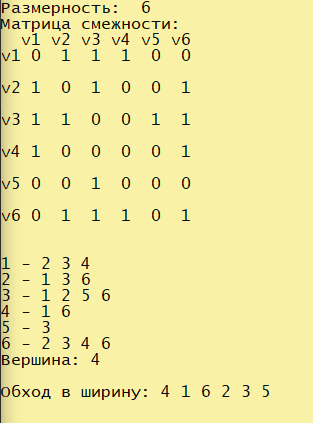
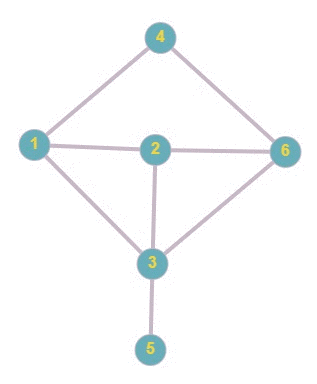
**Описание метода решения задач:**

Реализация состоит из подготовительной части, в которой все вершины помечаются как не поcещённые и осуществляется запуск процедуры обхода для вершин графа.

В самой процедуре обхода сначала создается пустая очередь, в которую помещается исходная вершина, из которой начат обход.

Далее итерационно, пока очередь не опустеет, из нее извлекается первый элемент, который становится текущей вершиной. Затем в цикле просматривается **v**-я строка матрицы смежности графа G(v,i). Как только алгоритм встречает смежную с **v** не посещенную вершину, эта вершина помещается в очередь и помечается как посещенная. После просмотра строки матрицы смежности алгоритм делает следующую итерацию цикла или заканчивает работу, если очередь пуста.

**Результат работы программы:**





Начнем обход с вершины 4. У этой вершины есть две смежные: 1 и 6. Выбираем 1 (4, 1), затем возвращаемся обратно в 4 и выбираем 6 (4, 1, 6), затем опять возвращаемся в 4, но так как смежных вершин больше нет, идем опять в 1 и ищем вершины там. Есть вершина 2 (4, 1, 6, 2), затем возвращаемся опять в вершину 1, и находим вершину 3 (4, 1, 6, 2, 3), опять возвращаемся в вершину 1, но так как смежных вершин больше нет, то переходим в 3-ю, а у 3-ей есть только одна не посещённая и смежная вершина с номером 5 (4, 1, 6, 2, 3, 5), возвращаемся в вершину 3, но там вершин больше нет и обход можно считать завершенным.



Исходя из результатов работы программы и ручного подсчета можно сделать вывод, что обход в том и другом случае совпадает и программа работает правильно.

**Вывод:**

Научились применять алгоритм обхода графа в ширину на примере матричного представления графа, а также визуализировать его в виде обыкновенного графа.

Листинг 1:

#include <time.h>

#include <iostream>

#include <stack>

#include <queue>

#include <conio.h>

using namespace std;

queue <int> Q;

int i, j, m, n, v;

bool\* vis = new bool[m];

int\*\* graph;

void BFS(int s) {

int i;

Q.push(s);

vis[s] = 1;

while (!Q.empty()) {

s = Q.front();

Q.pop();

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

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

if (graph[s][i] != 0 && vis[i] == 0) {

Q.push(i);

vis[i] = 1;

}

}

}

}

void DFS\_main()

{

setlocale(LC\_ALL, "Rus");

printf("Размерность: ");

scanf\_s("%d", &m);

n = m;

graph = new int\* [m];

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

graph[i] = new int[m];

}

hah:

int start;

printf("Матрица смежности: \n");

srand(time(NULL));

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

for (j = i; j < m; ++j)

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

printf(" ");

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

{

printf("v%d ", j + 1);

}

printf("\n");

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

{

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

vis[i] = false;

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

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

printf("\n\n");

}

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

{

printf("\n%d - ", i + 1);

for (j = 0; j < m; j++) {

if (graph[i][j] == 1) printf("%d ", j + 1);

}

}

printf("\nВершина: ");

scanf\_s("%d", &start);

if (start == 777) {

system("CLS");

goto hah;

}

while (start > m) {

printf("\nТакой вершины нет!\n");

printf("\nВершина: ");

scanf\_s("%d", &start);

}

printf("\nОбход в ширину: ");

BFS(start-1);

\_getch();

}

int Menu(void)

{

system("CLS");

int c = 0;

setlocale(LC\_ALL, "Rus");

while (c < '0' || c > '1')

{

printf(

"::=================================::\n"

"|| 1 : Обход в ширину ||\n"

"|| 0 : Выход ||\n"

"::=================================::\n"

"> ");

c = \_getch();

system("cls");

}

return c;

}

void main(void)

{

int Select;

while ((Select = Menu()) != '0' && Select != 27)

switch (Select)

{

case '1':

DFS\_main();

break;

}

}