Министерство науки Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №8

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему: «Обход графа в ширину»

Выполнили студенты

группы 21ВВ4:

Шутихин А.Э.

Ошкин И.В

Приняли:

Юрова О. В.

Акифьев И.В.

Пенза 2022

**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++.

**3.**\* Реализуйте процедуру обхода в ширину для графа, представленного списками смежности.

**Задание 2\***

1. Для матричной формы представления графов реализуйте алгоритм обхода в ширину с использованием очереди, построенной на основе структуры данных «список», самостоятельно созданной в лабораторной работе № 3.
2. Оцените время работы двух реализаций алгоритмов обхода в ширину (использующего стандартный класс **queue** и использующего очередь, реализованную самостоятельно) для графов разных порядков.

**Листинг:**

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <locale.h>

#include <queue>

using namespace std;

queue<int>q;

void BFS(int s, int n, int\* vis, int\*\* G) {

vis[s] = 1;

q.push(s);

while (!q.empty()) {

s = q.front();

q.pop();

printf("%d ", s);

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

{

if (G[s][i] == 1 && vis[i] == 0)

{

q.push(i);

vis[i] = 1;

}

}

}

}

int main() {

int n,m, i = 0;

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

printf("Введите количество вершин для матрицы: ");

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

int\*\* G = new int\* [n];

int\* vis = new int[n];

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

{

G[i] = new int[n];

vis[i] = 0;

}

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

{

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

{

if (i == j)

{

G[i][j] = 0;

}

else if (i != j)

{

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

G[j][i] = G[i][j];

}

}

}

printf("Матрица смежности:\n");

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

{

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

{

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

}

printf("\n");

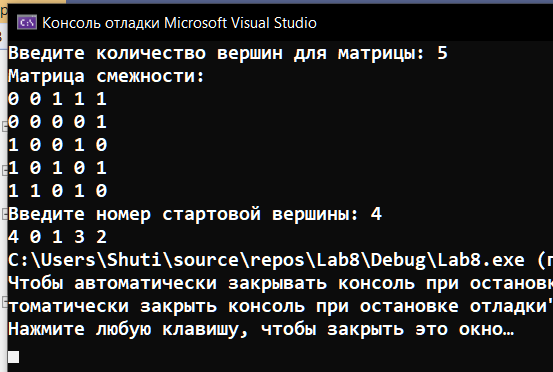
}

printf("Введите номер стартовой вершины: ");

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

BFS(m, n, &vis[0], &(&G)[0][0]);

}



Вывод: Мы научились реализовывать алгоритм прохождения графа в глубину.