Федеральное государственное бюджетное образовательное учреждение высшего образования «Пензенский Государственный университет»

Кафедра «Вычислительная техника»

ОТЧЕТ

**По лабораторной работе №8**

по курсу «Логика и основы алгоритмизации в ИЗ»

на тему «Обход графа в ширину»

Выполнили: ст. гр. 21ВВ1

Сурков А.А

Хорошильцев Е.А

Кривенков И.В

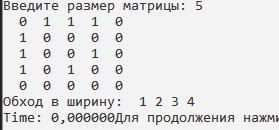
Приняли: Юрова О.В.

Акифьев И.В.

2022

**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++.



**Вывод:**

В ходе данной лабораторной работы мы научились пользоваться алгоритмом обхода графа в ширину.

**Листинг**

**Задание 1:**

#include "StdAfx.h"

#include <iostream>

#include <queue>

#include "time.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

//const int size = 7;

queue<int> Queue;

int size;

printf("Введите размер матрицы: ");

scanf("%d", &size);

int\*\* mas = (int\*\*)malloc(sizeof(int\*) \* size);

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

mas[i] = (int\*)malloc(sizeof(int) \* size);

}

srand(time(NULL));

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

for (int j = 0; j < size; j++) {

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

}

}

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

for (int j = 0; j < size; j++) {

if (i == j) {

mas[i][j] = 0;

}

if (mas[i][j] = mas[j][i]) {

mas[i][j] = mas[j][i];

}

}

}

for (size\_t i = 0; i < size; ++i) {

for (size\_t j = 0; j < size; ++j)

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

printf("\n");

}

int \*nodes = new int [size];

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

nodes[i] = 0;

int start = clock();

Queue.push(0);

printf("Обход в ширину: ");

while (!Queue.empty())

{

int node = Queue.front();

Queue.pop();

nodes[node] = 2;

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

{

if (mas[node][j] == 1 && nodes[j] == 0)

{ // если вершина смежная и не обнаружена

Queue.push(j);

nodes[j] = 1;

}

}

printf(" ");

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

}

int end = clock();

double time = (end-start) / 1000.0;

printf("\nTime: %lf", time);

cin.get();

return 0;

}