Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительной техники»

**Отчет**

По лабораторной работе №5

По курсу «Логика и основы алгоритмизации в инженерных задачах»

На тему «Обход графа в ширину»

Выполнили студенты гр.20ВВ4

Филиппов О.Р.

Кузнецов Н.С.

Проверили:

Юрова О.В

Акифьев И.В.

Пенза, 2021

**Цель работы:** реализовать алгоритм обхода графа в ширину.

**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++.

**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.

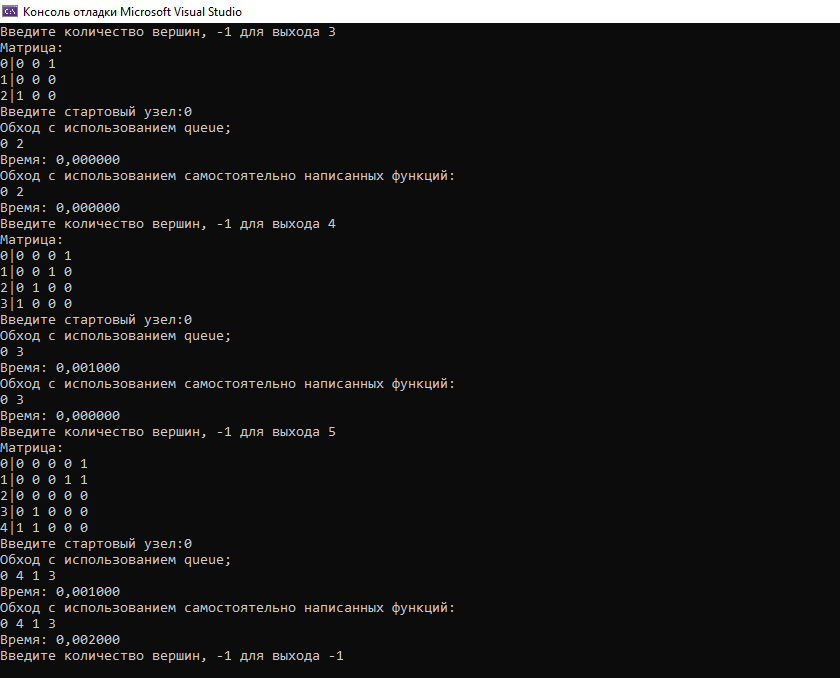
**Листинг:**

В репозитории.

**Пояснение к программе:**

Программа формирует матрицу и массив, в котором будут храниться посещённые вершины. Функция BFS() - стандартный алгоритм обхода в ширину, использующий очередь (класс queue). Функция BFS() - также стандартный алгоритм, но использующий функции работы с очередью, написанные самостоятельно. Функция createList() подготавливает указатели, используемые в очереди, для добавления новых элементов. Функция pop() заносит элемент в очередь, функция push() - извлекает элемент из очереди (при этом извлечённый элемент удаляется). Программа может создавать матрицы разных порядков и измеряет время, затраченное на выполнение обхода.

**Результат работы:**



**Вывод:** мы изучили алгоритм обхода графа в ширину, реализовали его с использованием класса queue и самостоятельно написанных функций.