МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Кафедра «Вычислительной техники»

**ОТЧЁТ**

по лабораторной работе №10  
по курсу «Поиск расстояний во взвешенном графе»  
на тему «Обход графа в ширину»

Выполнил:

Студент группы 23ВВВ2

Пичаев И. А.

Чупраков С. В.

Приняли:

Митрохин М. А.  
Юрова О.В.

Пенза 2024

**Цель работы**

Приобрести навыки программирования и работы алгоритмов нахождения расстояния до других вершин графа различными способами.

**Задание**

### **Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного взвешенного графа *G*. Выведите матрицу на экран.
2. Для сгенерированного графа осуществите процедуру поиска расстояний, реализованную в соответствии с приведенным выше описанием. При реализации алгоритма в качестве очереди используйте класс **queue** из стандартной библиотеки С++.

**3.**\* Сгенерируйте (используя генератор случайных чисел) матрицу смежности для ориентированного взвешенного графа *G*. Выведите матрицу на экран и осуществите процедуру поиска расстояний, реализованную в соответствии с приведенным выше описанием.

### 

### **Задание 2**

1. Для каждого из вариантов сгенерированных графов (ориентированного и не ориентированного) определите радиус и диаметр.
2. Определите подмножества периферийных и центральных вершин.

### **Задание 3\***

1. Модернизируйте программу так, чтобы получить возможность запуска программы с параметрами командной строки (см. описание ниже). В качестве параметра должны указываться тип графа (взвешенный или нет) и наличие ориентации его ребер (есть ориентация или нет).

**Ход работы**

### **Задание 1.1-2**

Эта программа реализует задачу поиска расстояний от исходной вершины в неориентированном графе с помощью алгоритма, похожего на BFS (поиск в ширину). Рассмотрим, что она делает и разберем её главную функцию:

В функции BFSD реализуется алгоритм, который вычисляет минимальные расстояния от исходной вершины до всех остальных вершин графа, используя очередь.

Начинается с того, что исходная вершина помещается в очередь с расстоянием 0. Далее происходит обход в ширину (по уровням), обновляются расстояния для всех достижимых вершин.

Ребра, которые существуют между вершинами, учитываются при подсчете расстояний, и обновление расстояний происходит с учетом веса рёбер.

Программа выводит матрицу смежности графа и затем для каждой вершины выводит её расстояние от исходной вершины, либо сообщает, что вершина недоступна (если она не достижима).

### **Задание 1.3**

### Программа отличается от предыдущей способом генерацией графа.

### Программа генерирует случайную матрицу смежности для ориентированного графа с заданным количеством вершин n. В этой матрице:

### Если есть рёбра между вершинами, то в соответствующих ячейках будет стоять случайный вес от 1 до 10.

### **Задание 2.1-2**

Программа реализует задачи работы с ориентированным графом, включая генерацию случайной матрицы смежности, вычисление расстояний от исходной вершины, а также нахождение радиуса, диаметра и центральных/периферийных вершин графа.

Функция findGraphProperties() вычисляет радиус и диаметр графа, а также находит центральные и периферийные вершины.

Диаметр графа — это максимальное расстояние между двумя вершинами в графе.

Радиус графа — это минимальное максимальное расстояние от какой-либо вершины до всех остальных.

Центральные вершины — это вершины, для которых максимальное расстояние до других вершин минимально (то есть они находятся ближе всего к остальным).

Периферийные вершины — это вершины, для которых максимальное расстояние до других вершин максимально (то есть они находятся дальше всех).

### **Задание 3**

В этом коде параметры командной строки передаются через массив argv, и с помощью цикла for они анализируются. Параметры могут быть переданы в любом порядке, как указано в примере.

Используются такие параметры:

-weighted — указывает, что граф взвешенный.

-directed — указывает, что граф ориентированный.

-n — количество вершин в графе.

-start — вершина, от которой нужно вычислять расстояния.

Пример запуска программы:

Program.exe -weighted -directed -n 5 -start 0

В этом случае будет сгенерирован ориентированный взвешенный граф с 5 вершинами, и программа будет вычислять расстояния от вершины 0.

### **Результаты работы программы**

### 

Рисунок 1 — Результаты работы программы 1.1-2

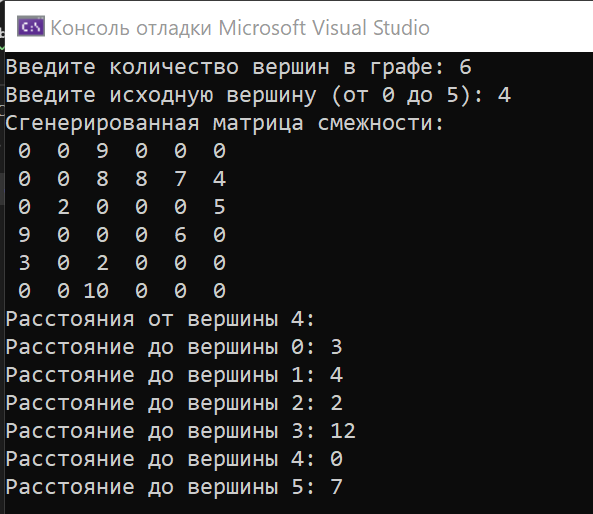


Рисунок 2 — Результаты работы программы 1.3

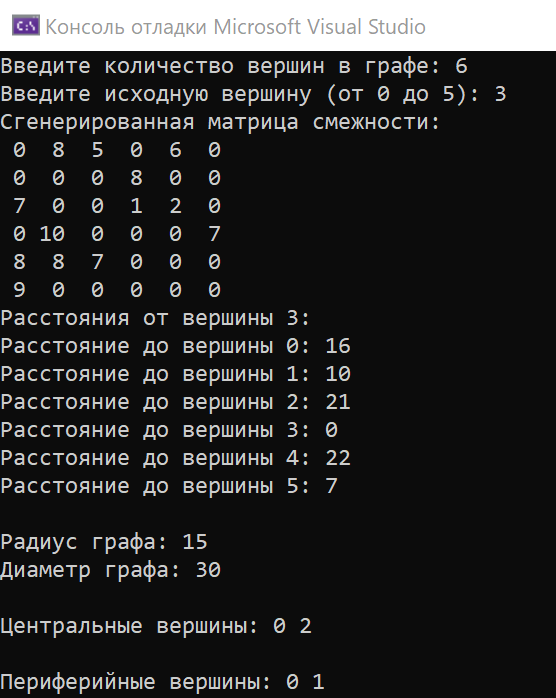


Рисунок 3 — Результаты работы программы 2.1-2

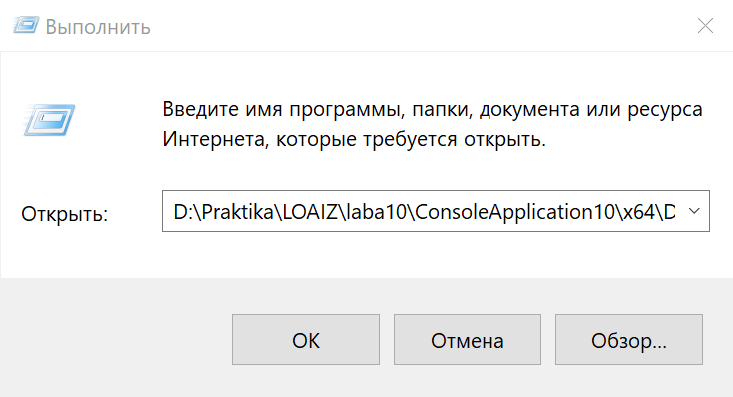


Рисунок 4 — Ввод в командную строку команды для вызова программы 3

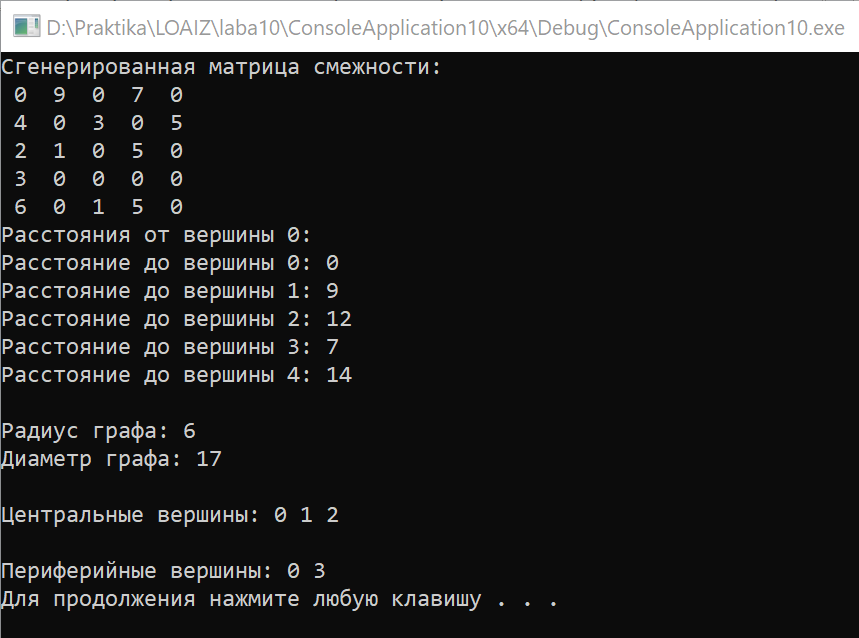


Рисунок 5 — Результаты работы программы 3

**Вывод**

Приобретены навыки программирования и работы алгоритмов нахождения расстояния до других вершин графа различными способами.

.