Федеральное государственное автономное образовательное учреждение

высшего образования

«Нижегородский государственный университет

им. Н.И. Лобачевского»

Институт информационных технологий математики и механики

Отчёт по лабораторной работе

**«Дороги: алгоритм Дейкстры»**

Выполнила:

студентка гр. 0823-2

Рожнова М.А.

Проверил:

ассистент каф. МОСТ института ИТММ

Сиднев А.А.

Нижний Новгород

2016 г.

Оглавление

[Введение 2](#_Toc453370579)

[Постановка задачи 3](#_Toc453370580)

[Руководство программиста 4](#_Toc453370581)

[Общая структура проекта 4](#_Toc453370582)

[Описание структур данных 4](#_Toc453370583)

[Описание алгоритмов 5](#_Toc453370584)

[Заключение 6](#_Toc453370585)

# Введение

Графы формально описывают множество близких ситуаций. Самым привычным примером служит карта автодорог, на которой изображены перекрестки и связывающие их дороги. Перекрестки являются вершинами графа, а дороги — его ребрами. Графы могут быть ориентированы (подобно улицам с односторонним движением) или взвешены, когда каждой дороге приписана стоимость путешествия по ней (если, например, дороги платные).

С формальной точки зрения граф представляет собой упорядоченную пару множеств первое из которых состоит из вершин (узлов) графа, а второе — из его ребер. Ребро связывает между собой две вершины. При работе с графами часто решается вопрос, как проложить путь из ребер от одной вершины графа к другой. В этом случае говорят о движении по ребру, что означает переход из вершины графа в другую вершину , связанную с ней ребром .

# Постановка задачи

Реализовав алгоритм Дейкстры и приоритетную очередь с использованием структур данных d-куча и АВЛ-дерево, необходимо решить следующую задачу.

Дорожная сеть города представляет собой N перекрёстков, соединённых M дорогами с двусторонним движением. Стартовый перекрёсток (начало маршрута) - 1, финальный перекрёсток - N. По заданной карте города вычислите длину кратчайшим по суммарной длине маршрута и количество дорог, которое удалось посмотреть команде. Из всех таких маршрутов выбрать тот, который включает в себя наибольшее количество дорог.

**Входные данные:**

Первая строка содержит два целых числа N и M — количество перекрёстков в городе и количество дорог соответственно (2 ≤ N ≤ 1000, 1 ≤ M ≤ 2⋅10^5).

Каждая из следующих M строк содержит три целых числа ai, bi и ci (1 ≤ ai, bi ≤ N, ai ≠ bi, 1 ≤ ci ≤ 1000) — номера перекрёстков, соединяемых i-й дорогой, и длину этой дороги.

**Вывод:**

Два целых числа P и Q — длину кратчайшего пути от старта до финиша и максимальное число дорог, которые может включать в себя кратчайший путь.

# Руководство программиста

## Общая структура проекта

* **Class** – директория для размещения файлов, относящихся к реализации d-кучи, и АВЛ-дерева.
* **Sample** – директория для размещения файла, содержащего реализацию алгоритма Дейкстры.
* **отчет** – директория для хранения отчёта по лабораторной работе.

## Описание структур данных

**Приоритетная****очередь** — это абстрактный тип данных, предназначенный для представления взвешенных множеств. Множество называется взвешенным, если каждому его элементу однозначно соответствует число, называемое ключом или весом. Основными операциями над приоритетной очередью являются следующие операции:

* **ВСТАВИТЬ** в множество новый элемент со своим ключом.
* **ИЗВЛЕЧЬ** из множества элемент с максимальным ключом.

Приоритетная очередь может быть реализована на основе различных структур данных, например, в данной работе на основе структур данных d-куча и АВЛ-дерево.

**Представление приоритетной очереди с помощью d-кучи**

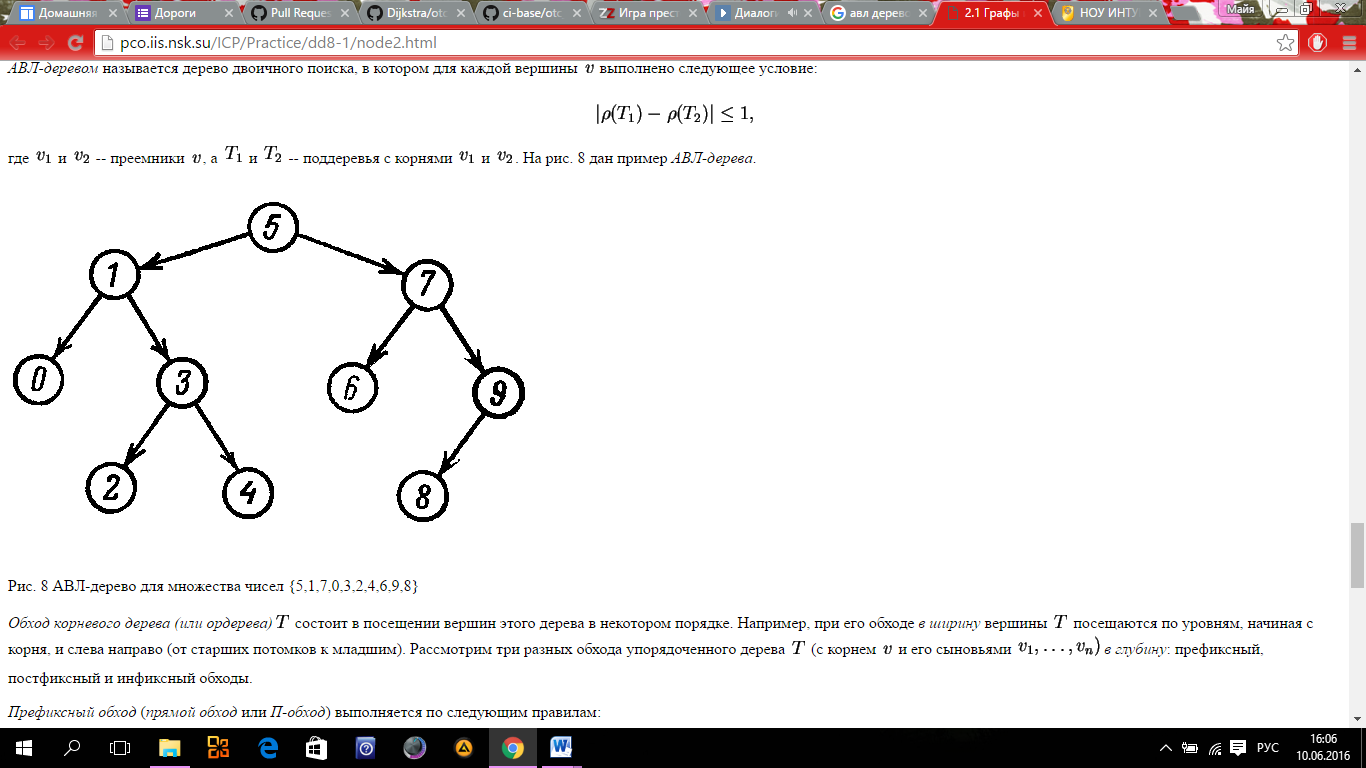
Представление приоритетной очереди с помощью -кучи основано на использовании так называемых завершенных -арных деревьев .

Завершенное -арное дерево — это корневое дерево со следующими свойствами:

* Каждый внутренний узел (то есть узел, не являющийся листом дерева), за исключением, быть может, только одного, имеет ровно потомков. Один узел-исключение может иметь от до потомков.
* Если — глубина дерева, то для любого такое дерево имеет ровно узлов глубины .
* Количество узлов глубины в дереве глубины может варьироваться от до .

Узлы завершенного -арного дерева принято нумеровать следующим образом: корень получает номер , потомки узла с номером получают номера. Такая нумерация удобна тем, что позволяет разместить узлы дерева в массиве в порядке возрастания их номеров, при этом позиции потомков любого узла в массиве легко вычисляются по позиции самого узла. Так же легко по позиции узла вычислить позицию его родителя. Так, для узла, расположенного в позиции , родительский узел располагается в позиции , где — операция деления нацело.

**АВЛ-дерево** – это двоичное дерево поиска, для каждой вершины которого выполняется требование: высота левого и правого поддеревьев различаются не более, чем на единицу. При операциях добавления и удаления может произойти нарушение сбалансированности дерева. В этом случае потребуются некоторые преобразования, не нарушающие упорядоченности дерева и способствующие лучшей сбалансированности - повороты вокруг тех или иных узлов дерева.



## Описание алгоритмов

**Алгоритм Дейкстры** - алгоритм на графах. Находит кратчайшие пути от одной из вершин графа до всех остальных.

Дан взвешенный ориентированный граф без дуг отрицательного веса. Найти кратчайшие пути от некоторой вершины графа до всех остальных вершин этого графа.

Каждой вершине из сопоставим метку — минимальное известное расстояние от этой вершины до . Алгоритм работает пошагово — на каждом шаге он «посещает» одну вершину и пытается уменьшать метки. Работа алгоритма завершается, когда все вершины посещены.

**Инициализация.** Метка самой вершины a полагается равной , метки остальных вершин — бесконечности. Это отражает то, что расстояния от до других вершин пока неизвестны. Все вершины графа помечаются как непосещённые.

**Шаг алгоритма.** Если все вершины посещены, алгоритм завершается. В противном случае, из ещё не посещённых вершин выбирается вершина , имеющая минимальную метку. Мы рассматриваем всевозможные маршруты, в которых является предпоследним пунктом. Вершины, в которые ведут рёбра из , назовём соседями этой вершины. Для каждого соседа вершины , кроме отмеченных как посещённые, рассмотрим новую длину пути, равную сумме значений текущей метки и длины ребра, соединяющего с этим соседом. Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину как посещённую и повторим шаг алгоритма.

# Заключение

В ходе работы была полностью реализована поставленная задача. Была написана программа, вычисляющая длину кратчайшего по суммарной длине маршрута и количество дорог, из которых состоит этот маршрут. Очередь с приоритетами реализована с помощью структур данных d-куча и АВЛ-дерево. Были получены результаты работы программы для четырех контрольных тестов. Разработано консольное приложение, демонстрирующее результат работы алгоритма для двух вариантов реализации приоритетной очереди.

Результаты для контрольных тестов в виде пары чисел (P, Q) — длина кратчайшего пути от старта до финиша и максимальное число дорог, которые может включать в себя кратчайший путь.

016 - (3,3)

017 - (799200, 999)

018 - (109,99)

019 - (392883,999)