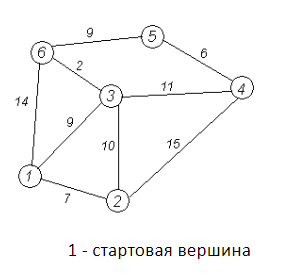
**Билет №6**

**Алгоритм Дейкстры. Улучшение алгоритма для разреженных графов за O(ElogV)**

**Постановка задачи:** Дан ориентированный или неориентированный взвешенный (у каждого ребра есть свой вес) граф с n вершинами и m ребрами. Веса всех ребер неотрицательны. Указана некоторая стартовая вершина s. Нужно найти длины кратчайших путей из вершины s во все остальные вершины графа:



**Описание старого алгоритма:**

1. Сопоставим каждой вершине 2 значения: 1-минимальное известное расстояние от стартовой до текущей (например:1,3 = 9), 2- массив visit[v] = true/false (посетили ли мы эту вершину или нет)
2. Определяем минимальные расстояния от вершины s до остальных: до самой себя = 0, до других = ∞ (на начале). Все вершины графа изначально не посещённые (visit[v] = false)
3. Будем посещать вершину расстояние от s до которой минимально. Если такой вершины нет, что не существует такого пути не равного ∞, то алгоритм завершается и вершина будет не посещена. А если нашлась такая вершина, то она будет посещена.
4. Пытаемся улучшить расстояние для соседних вершин суммой расстояния до выбранной вершины и длины ребра соседа. Те рассмотреть обходные пути к вершине через другие вершины (пример вершина 3). = пытаемся найти посредника
5. Теперь когда мы минимизировали кратчайший путь до какой-то вершины, то возвращаемся к пункту 2.

**Описание улучшенного алгоритма и его временная сложность:**

Предыдущий алгоритм работает за O(V^2+E), а дальнейший алгоритм работает за O(ElogV), алгоритм схож с предыдущим реализацией, но мы фактически убираем цикл, который отвечает за поиск минимума введением какой-нибудь структуры данных, которая способна поддерживать минимум за O(log(n)), на мой взгляд в Java легче всего использовать PriorityQueue (единственное она располагает элементы по убыванию, но мы перехитрим это), тогда время работы алгоритма составит O(V\*log(V)) (время нахождения минимума для каждой вершины)+O(E\*log(V)) (время релаксации(изменения d[to])  = O(V\*log(V)+O(E\*log(V)) = O(E\*log(V))

**Задача:**

**Изображение выглядит как текст

Автоматически созданное описание**

**Решение:** https://pastebin.com/1WVc5tEB