1. Запустим алгоритм Дейкстры поиска кратчайшего пути из вершины a, исключая из пути ребро с отрицательным весом. Фиксируем расстояние до вершины b. Далее запускаем два поиска в глубину: сначала от вершины a до U, а затем от вершины V до b. И вычитаем из суммы длин этих двух путей вес ребра (U, V).

Сложность алгоритма Дейкстры поиска кратчайшего пути и поиска в глубину равна $O((|V| + |E|) * \log |V|)$.

2 Рассмотрим каждый пункт:

а) Это утверждение верно. Докажем это: по определению остовное дерево должно включать в себя все вершины графа. Соответсвенно, должно и вершины, которые соединены минимальным уникальным ребром. Тогда, когда бы ни были включены любая из этих вершин, на следующем же этапе алгоритма, вторая точка будет включена в остовное дерево как вершина с наименьшим расстоянием.

б) Это утверждение верно. Докажем это: изобразим разрез графа, доли которого соединяет ребро e. Если есть какое-то еще ребро e', вес которого меньше, чем e, то образуется цикл, из которого очеивдно нужно выбрать ребро с минимальным весом, то есть e', но тогда получается противореие. Это значит, что ребро e - ребро минимального веса из пересекающих некоторый разрез.

в) Это утверждение неверно. Приведем контрпример: представим следующую картину: слева у нас вершина A, справа - вершина B, между которыми мы ищем кратчайший путь. По "верхней"ветке у нас ровно одна точка, такая что $\rho(A, X) = 1$, $\rho(X, B) = 2$, а на "нижней"ветке у нас 3 промежуточных точки X_1, X_2, X_3, таких что расстояние между ними равно 1, а также $\rho(A, X_1) = 1$, $\rho(X_3, B) = 1$. Тогда минимальное остовное дерево не будет совпадать с кратчашим путем из A в B.

3

4. Представим граф как АВЛ-дерево. Тогда, его глубина равна $\log n$. Чтобы получить нижнюю оценку выполняем m раз операцию $Find$ (операция $Union = 2 * Find$) и получаем необходимую асимптотику.

5. С помощью алгоритма Крускала, рассмотренного на лекции, строим минимальное остовное дерево, предварительно объединя все вершины городской сети в одну вершину. Таким способом мы получим доступ ко всем вершинам за минимальное число соеднений и с минимальной стоимостью.

Сложность алгоритма Крускала равна $O(|E| * (\log |V| + \log |E|))$.