Студент: Маргарита Чудова
Группа: 2
Дата: 27 апреля 2022 г.

---

1. (a) Пойдем от противного — допустим, что это не дерево. Дерево по определению — это связный граф без циклов. Дерево Штейнера должно быть связным по определению, значит с этой частью все хорошо. Тогда допустим, что в дереве Штейнера могут быть циклы.
Допустим, существет цикл. Возьмем в этом цикле ребро с самым большим весом и выкинем его из дерева Штейнера. Мы не испортили определение — поскольку граф неориентированный, мы все еще из любой вершины этого цикла можем добраться в любую другую, при этом, поскольку ребра имеют неотрицательный вес, у нового графа вес будет меньше, чем у старого. Поступим так со всеми циклами, получим дерево.

   (b) Асимптотика и неотрицательные ребра намекают нам на алгоримтм Дейкстры, основанный на куче. Возьмем и по очереди запустим Дейкстру из каждой терминальной вершины, для того, чтобы мы смогли восстановить путь, для каждой вершины во время прохода Дейкстрой будем записывать в массив предшествующую ей вершину в оптимальном пути (как много раз мы делали в ДП для восстановления ответа).
Первая мысль, которая приходит в голову — возьмем два наименьших кратчайших пути между этими вершинами, это и будет наше дерево. Но это может не работать. У нас может быть такая ситуация:

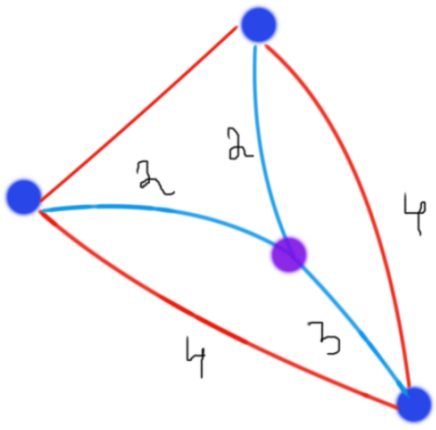

Рис. 1: Рисунок-пояснение. На Tikz времени так и не появилось :D

Пусть синие вершины — терминалы. Красные пути — это наименьшие пути из одного терминаал в другой. Видим, что через фиолетовую вершину можно построить дерево с меньшим весом за счет того, что часть пути как бы схлопывается. Такая развилка может быть только одна, поскольку вершин всего 3. Развилка может совпалать с терминалом, тогда просто будет путь а-ля красные ребра.
Будем искать вершинки такого формата. Для этого пройдемся по всем вершинкам и при обходе вершины будем считать величину

$$d[u] = dt_1[u] + dt_2[u] + dt_3[u]$$

Здесь dt_i — это насчитанное Дейкстрой расстояние от u до соответствующей вершины терминала. Из таких $d[u]$ берем минимум, восстанавливает пути от u до $t_1,\ t_2,\ t_3$ — это и есть наше дерево Штейнера.
На Дейкстру ушло $\mathcal{O}(E \log V)$, на обход $\mathcal{O}(V)$ — наверное, мы не считаем, что наш граф суперразреженный и обход нам сильно сломает асимптотику, это было бы немного странно. Но если нам это не нравится, можно сказать, что мы не просто обходим все вершины, а запускаемся dfs-ом только из терминалов. Тогда пройдем только по ребрам.

(c) В этом пункте ситуация похожая, но поскольку вершин 4, здесь уже может быть две развилки (как и в предыдущей части, вершины развилки могут совпадать с терминалами или друг с другом).

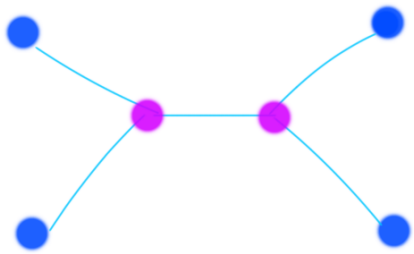

Рис. 2: Рисунок-пояснение к пункту (c)

Нам нужно посчитать все кратчайшие пути между всеми вершинами, чтобы в дальшейшем работать с развилками и терминалами (если в предыдущей части нам было достаточно сосчитать расстояния от всех вершин до терминалов, то здесь нет из-за того, что нужно знать расстояние между развилками).
Для этих целей подходит алгоритм Флойда-Уоршелла, его асимптотика как раз $\mathcal{O}(V^3)$. Зная все эти пути, можем перебрать все пары потенциальных вершин-развилок, посчитать сумму весов кратчайших путей до терминалов и между самими вершинами, и аналогично предыдущему пункту выбрать минимум и восстановить ответ. На это все у нас уйдет $\mathcal{O}(V^2)$, так что итоговая асимптотика будет $\mathcal{O}(V^3)$.

2. Пусть d — исходная матрица расстояний. Рассмотрим некоторые вершины $u,\ v$ в графе, пусть (a, b) — это ребро, вес которого мы меняем с $w_{(a,b)}$ на $w'_{(a,b)}$. Попробуем разбить путь между ними, проходящий через искомое ребро, на сумму трех частей — от u до a, от a до b (просто наше ребро) и от b до v. Понятно, что раз мы хотим рассматривать кратчайший путь, отдельные его части тоже должны быть кратчайшими, берем их из d.
Может быть два варианта. Первый: между u и v не существует пути через (a, b), и тогда одна из всех трех частей пути равна бесконечности, а в d ничего не меняется. Тогда для того, чтобы понять, нужно ли что-то менять в паре $u,\ v$, нужно проверить условие

$$d[u][a] + w'_{(a,b)} + d[b][v] < \infty$$

Второй: путь существует и тогда он потенциально может быть меньше, чем исходный.
Тогда можно пройтись по всем парам вершин $u,\ v$ и для второго случая пересчитать d следующим образом:

$$d = \min(d[u][v], d[u][a] + w'_{(a,b)} + d[b][v])$$

Это займет у нас как раз $\mathcal{O}(V^2)$ времени.

3. (a) Рассмотрим некий путь $v_1, v_2, \ldots, v_n$. Его изначальный вес равен

$$W = \omega(v_1, v_2) + \ldots + \omega(v_{n-1}, v_n)$$

После того, как мы заменили веса, суммарный вес равен

$$W' = \omega(v_1, v_2) + \phi(v_1) - \phi(v_2) + \omega(v_2, v_3) + \phi(v_2) - \phi(v_3) + \ldots + \phi(v_{n-1}) - \phi(v_n) + \omega(v_{n-1}, v_n) =$$

$$= \omega(v_1, v_2) + \ldots + \omega(v_{n-1}, v_n) + \phi(v_1) - \phi(v_n)$$

Из этой формулы видно, что для любых путей из v_1 в v_n вес изменится на одинаковую величину $\phi(v_1) - \phi(v_n)$, значит кратчайший путь из v_1 в v_n остался кратчайшим.

(b) С таким потенциалом получаем

$$\omega'(v, u) = \omega(v, u) + \rho(s, v) - \rho(s, u)$$

$\omega(v, u)$ больше или равно растоянию между вершинами $v,\ u$ — $\rho(u, v)$. Тогда

$$\omega'(v, u) \geq \rho(v, u) + \rho(s, v) - \rho(s, u) = 0$$

(c) В предыдущем пункте мы предьявили потенциал Джонсона, который сохраняет кратчайшие пути и превращает веса в неотрицательные. Пусть s — некоторая вершина, посчитаем все расстояния от нее за $\mathcal{O}(VE)$. Потом применим Джонсона — возьмем $\phi(v) = \rho(s, v)$, пересчитаем веса и применим алгоритм Дейкстры к каждой вершине, чтобы для каждой вершины насчитать расстояния до других. Веса нужно будет пересчитывать в исходные:

$$\omega(v, u) = \omega'(v, u) - \rho(s, v) + \rho(s, u)$$

Таким образом, запускаем Дейкстру V раз, получаем время $\mathcal{O}(VE \log V)$