

1. Дан неориентированный граф $G = (V, E)$, веса рёбер которого не обязательно различны...

а) Если к каждому ребру графа прибавить вес w , то каждое минимальное остовное дерево G перейдёт в минимальное остовное дерево модифицированного графа.

Да, это верное утверждение. (Вроде как даже рассматривали подобный случай на семинаре, но все равно распишу доказательство).

Вспомним алгоритм Крускала построения минимального остовного дерева, который разбирался на семинаре: сначала мы сортируем рёбра. Очевидно, что при добавлении к каждому ребру значение w порядок рёбер после сортировки не изменится. Значит, алгоритм в результате выдаст такое же минимальное остовное дерево, но его вес будет больше на $(n - 1) \cdot w$, где n – количество вершин в графе.

б) Если самое лёгкое ребро графа G уникально, то оно входит в любое минимальное остовное дерево.

Да, это верное утверждение.

Вновь вспомним алгоритм Крускала. После сортировки рёбер по весу самое легкое ребро будет самым легким (вот это поворот), следовательно, с него мы и начнем построение остова. Раз уж с него мы начинаем построение остова, то оно в любом случае будет принадлежать этому остову, то есть какой бы минимальный остов мы ни взяли, первое ребро гарантированно будет лежать в нем.

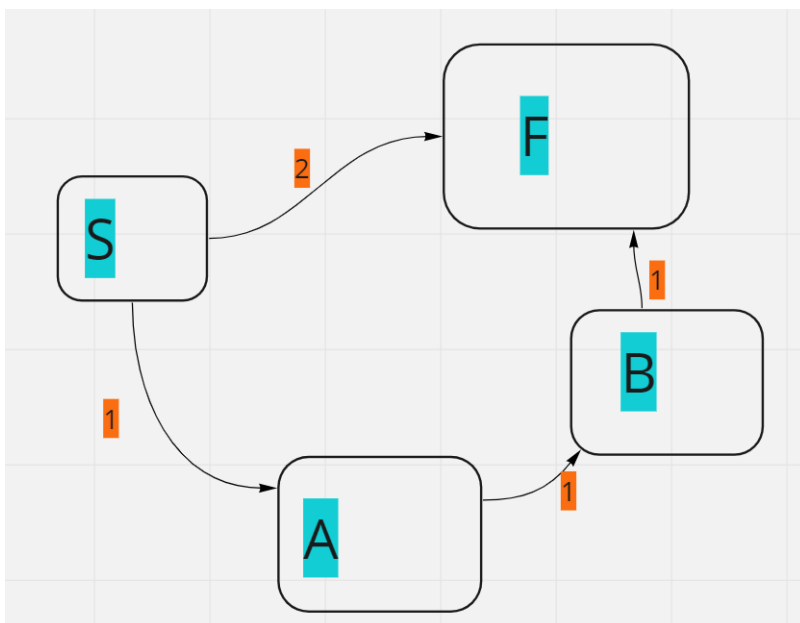
в) Если ребро e входит в некоторое минимальное остовное дерево, то оно является самым лёгким ребром из пересекающих некоторый разрез.

Да, это верное утверждение. Докажем это сильное утверждение от противного.

Предположим, что для каждого из разрезов, которые пересекает ребро e , можно найти более легкое ребро из этого разреза. Но в таком случае при любом разрезе графа выгоднее вместо ребра e взять то самое более легкое ребро, откуда следует, что e не будет лежать в остове, что противоречит условию. Значит, предположение неверно, а утверждение верно.

г) Кратчайший путь между двумя вершинами является частью некоторого минимального остовного дерева.

Это неправда. Контрпример мы рассматривали на семинаре, но я буду оригинальным и придумаю свой.



Контрпример

Здесь множество минимальных остовных состоит из ребер $\{sa, ab, bf\}$. Однако, кратчайший путь от S до F состоит из множества ребер $\{sf\}$, некоторые ребра которого не присутствуют ни в одном минимальном остовном дереве. Ура. Мы все доказали.

Определение. Граф, который получается из графа G удалением некоторых вершин и рёбер, называют (*рёберным*) *подграфом* графа G . В случае, если при изготовлении подграфа, рёбра удалялись только вместе с удалением вершин, подграф называют *индуцированным*.

2. Пусть T — минимальное остовное дерево графа G , а $H \dots$

Предположим, что найдется ребро (u, v) , которое входит в T и H , но не входит ни в какое минимальное остовное дерево H . Это значит, что существует ребро, соединяющее разрез $(A, B) : u \in A, v \in B$, такое, что его вес меньше веса (u, v) . Но в таком случае в графе G появляется цикл, проходящий по этим двум ребрам. Очевидно, что для построения минимального остова нам необходимо будет взять все ребра, кроме самого тяжелого.

Если из указанных выше ребер (u, v) — самое тяжелое, то убираем его. Но оно лежало в остова по предположение, значит, предположение неверно и первый случай доказан.

Если оно не самое тяжелое, то найдется ребро, лежащее в этом цикле, которое тяжелее него. Но в таком случае мы не можем убрать его при построении минимального остова H , т.к. во всех циклах найдется ребро, тяжелее (u, v) .

Получается, что (u, v) в таком случае входит в минимальное остовное дерево графа H , что тоже противоречит предположению.

Оба случая пришли к противоречию, значит, утверждение доказано.

3. Рассмотрим алгоритм Union-Find без улучшения со сжатием путей¹. Приведите последовательность из m операций Union и Find над множеством из n элементов, которая потребует времени $\Omega(m \log n)$.

¹При вызове Find(x) все предки x вместе с x становятся детьми корня.

Предположим, что число вершин – степень двойки. Тогда при построении на множестве вершин бинарного дерева мы можем рекурсивно объединять семьи на каждом уровне по парам.

Далее запускаем Find для каждой вершины.

Количество поисков совпадает с поделенным на 2 количеством вершин и равно $\frac{n}{2} = m$. Количество объединений равно m . Каждый поиск работает за $\log n$.

В результате у нас получается оценка $\theta(T) = \theta(m \log n) \Rightarrow \Omega(T) = \Omega(m \log n)$, что нам и было необходимо.

4. На вход задачи подаётся неориентированный взвешенный граф $G(V, E)$...

1) Строим минимальное остовное дерево для подграфа $V \setminus U$ за время $E \log V$. Если такое дерево построить не удалось, то, следовательно, деревьев, необходимых нам по условию нет.

2) Берем по очереди каждый лист и ищем для него самое легкое по весу ребро из тех, что пересекают разрез U и $V \setminus U$. Суммарная сложность обхода всех листьев – $O(E)$.

3) Найденное ребро добавляем к остову.

В итоге сложность алгоритма равна сложности алгоритма построения минимального остова, т.е. $O(E \log V)$.