Домашняя работа №11

Бредихин Александр

3 мая 2020 г.

Задача 1

Задача: Дан неориентированный граф $G = (V, E)$, веса рёбер которого не обязательно различны. Для каждого из утверждений ниже приведите доказательство, если оно истинно, или постройте контрпример, если оно ложно:

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

   Решение: это утверждение верно:
   если пользуемся корректностью алгоритма Крускалы, то добавление к каждому ребру графа веса w на его работу не повлияет (после сортировки по весам рёбра останутся в том же порядке, что и без добавления веса w). Следовательно алгоритм найдёт то же минимальное остновное дерево, что и без добавления.

   Можно рассуждать от противного: пусть после добавления веса w минимальное остовное дерево изменилось и стало T_2'. Пусть T_1' – дерево из которого оно получилось до добавления весов. В искодном графе T_1 – MST, из него получается T_2 и $w(T_2') < w(T_2)$. Так как к каждому ребру мы прибавляем один и тот же вес и количество рёбер в T_2 и T_2' одинаковое, так как это деревья, то верно такие равенства:
   $w(T_2') = w(T_1') + w \cdot k$
   $w(T_2) = w(T_1) + w \cdot k$, k - количество вершин в дереве
   так как $w(T_1) < w(T_1')$, так как T - MST. То получаем противоречие с тем, что $w(T_2') < w(T_2)$, следовательно наше предположение неверно.

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

Решение: это утверждение верно: это можно сказать сразу из алгоритма Крускала, так как в нём мы сортируем рёбра по весам и берём по порядку, чтобы не образовывалось циклов. Так как ребро уникальное и является самым лёгким, то после сортировки оно окажется первым.
То есть мы будем брать его первым, когда ещё ничего не взяли в MST, следовательно, оно точно попадёт в него, так как при его взятии не смогут образоваться циклы (так как в MST ещё ничего не взято).

Можно рассуждать от противного: пусть $e = (a, b)$ - минимальное уникальное ребро нашего графа. И есть $T - MST$, такое что $e \notin T$. Тогда добавим в T это ребро, появится цикл. Удалим из полученного цикла ребро, которое будет находиться на пути из a в b по T (такой путь точно будет, так как T - дерево). Получим новое остовное дерево, так как связность и то что охватываем все вершины не нарушится. Но его вес будет меньше T, так как $w(e) = min$ т оно уникально. Противоречие.

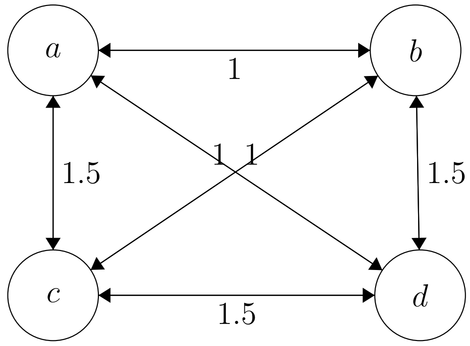
3) Если ребро e входит в некоторое минимальное остовное дерево, то оно является самым лёгким ребром из пересекающих некоторый разрез.

Решение: это утверждение верно:
Рассмотрим некотрый разрез и ребро e которое входит в $T = MST$ нашего графа. От противного: пусть это ребро не минимально и существует ребро k такое, что $w(k) < w(e)$.
Для этого ребра k будет верно то, что k и e лежат на цикле $T \cup k$: так как $T - MST$ содержит все вершины нашего графа и в нём минимальное число рёбер для связности T, если добавляем $k \notin T$, то получается цикл (если этот цикл будет в одном из подмножеств разреза и нет ребра e, то первоначально дерево не было связным, противоречие). Следовательно циклпроходит через k и e.
Из теории графов если удалим любое ребро цикла, то граф на этих вершинах останется связным. Удаляем ребро e, получаем новое остовное дерево (так как в нём нет циклов и оно проходит через все вершины), но вес нового остновного дерева будет меньше, чем

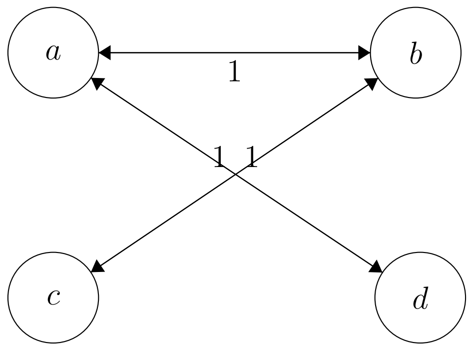
первоначального, так как $w(k) < w(e)$. Получили противоречие, следовательно, ребро e является самым лёгким ребром из пересекающих разрез.

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

Решение: это утверждение неверно, приведём контрпример (из семинара):



Для такого графа MST (с помощью алгоритма Крускала) будет выглядить так:



Минимальным путём между вершинами c и d будет ребро cd (с помощью алгоритма Дейкстры, так как все веса рёбер положительные), которого нет в MST, следовательно, кратчайший путь между двумя вершинами **НЕ**является частью некоторого минимального остовного дерева.

Задача 2

Задача: пусть T — минимальное остовное дерево графа G, а H — связный подграф G. Покажите, что рёбра, входящие как в T, так и в H, входят в некоторое минимальное остовное дерево графа H.

Рассматриваем рёбра E, которые будут пересечением рёбер остовного дерева G и множеством рёбер H. T_H – минимальное остовное дерево для подграфа H, пусть $T_H = E$. А далее будем действовать аналогично алгоритму Крускала: сортируем оставшиеся рёбра. Добавляем рёбра в T_H с минимальным весом чтобы не появлялись циклы. Получим остовное дерево (так как для H - остовное дерево и затем действуем алгоритмом Крускала), докажем, что полученное этими действиями остовное дерево будет минимально для всего графа.

От противного: пусть T'_H – MST (а не T_H). Тогда вместо дерева T_H поставим T'_H получим новое дерево T'. Это дерево, так как в нём нет циклов: по построению в T_H нет циклов и рёбра из H лежащие в T принадлежат различным компонентам связности.
Также T' – остовное дерево, так как оно содержит все рёбра графа G: T'_H - остовное дерево, дальше по построению. Получим остовное дерево, меньше, чем T – противоречие, что $T - MST$, следовательно, наше предположение неверно и T_H – MST подграфа H, которое содержит все E. То что нам и требовалось показать.

Задача 3

Задача: рассмотрим алгоритм Union-Find без улучшения со сжатием путей[1]. Приведите последовательность из m операций Union и Find над множеством из n элементов, которая потребует времени $\Omega(m \log n)$.

С помощью операций Union-Find получим бинарное дерево следующим способом: (б.о.о считаем, что количество вершин это точная степень двойки) сначала объединяем вершины по парам, затем обединяем каждые пары друг с другом и так далее. Заметим, что на kой итерации все корни имеют ранг k а сумма длин всех путей до корня будет равна $\frac{n}{2} \cdot k$. Это можно проверить с помощью индукции:

Б.И. $k = 1$ - верно.
Ш.И. пусть на kой итерации все корни имеют одинаковый ранг, то при

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

следующем попарном объединении множеств ранг будет увеличиваться на 1 у всех получившихся корней (по определению операции Union), следовательно, на $k+1$ом шаге ранг каждого корне равняется $k+1$.
Заметим, что при объединении двух множеств один из корней станет на 1 выше другого, следовательно путь до нового корня от каждой вершины множества, чей корень окажется ниже увеличится на 1. По построению во всех множествах будет одинаковое количество элементов, следовательно, путь увеличится у половины вершин. Следовательно, сумма всех путей станет $\frac{n}{2} \cdot (k+1)$. То есть доказали шаг индукции.

Посчитаем количество Union-Find в таком случае. Их получается $2n$. Всего итераций (так как бинарное дерево) было $k = \log n$. Следовательно, из того что сумма всех путей до корня равна $\frac{n}{2} \cdot k$ все поиски будут выполняться за $n \log n$. Поэтому оценка работы алгоритма (с учётом того, что $m = 2n$) получается: $\Theta(m \log n)$, что и требовалось показать.

Задача 4

Задача: на вход задачи подаётся неориентированный взвешенный граф $G(V, E)$ и подмножество вершин $U \subseteq V$. Необходимо построить остовное дерево, минимальное (по весу) среди деревьев, в которых все вершины U являются листьями (но могут быть и другие листья) или обнаружить, что таких остовных деревьев нет. Постройте алгоритм, который решает задачу за $O(|E| \log |V|)$. Обратите внимание, что искомое дерево может не быть минимальным остовным деревом.

Алгоритм: найдём (например, алгоритмом Крускала или Примы) минимальное остовное дерево на подграфе $V \backslash U$. (если такого нет, то искомого дерева не будет). Затем смотрим на разрез U и $V \backslash U$ и находим для каждой вершины из U минимальное по весу ребро, которое пересекает рассматриваемый разрез и добавляем его к полученному алгоритму дереву.

Корректность: если мы удалим из дерева листы, то оно все равно останется деревом. Нам нужно найти MST с листьями из U, поэтому на подграфе $V \backslash U$ остовное дерево должно быть минимальным (так как оно индуцирует MST, которое нам нужно) (если это не так и есть другое дерево, которое удволетворяет условиям задачи, то если мы уберём из него все листья, то получим дерево на подграфе $V \backslash U$ котрое должно быть минимальным, противоречие) (и оно должно быть, так как иначе при добавлении листьев связность не появится: по определению листа).

Также после нахождения дерева на подграфе $V \setminus U$ присоединяем к нему каждый лист минимальным образом (по минимальному ребру в разрезе). Поэтому получаем минимальное остовное дерево, удволетворяющее условию задачи.

Сложность: ищем минимальное дерево за $O(|E| \log(|V|))$ (сложность алгоритма Примы, считаем, что $|U|$ не влияет на асимптотику). Последующее добавление листьев в худшем случае занимает $O(|E|)$. Суммарно $O(|E| \log(|V|))$