Десятое задание по алгоритмам

Ситдиков Руслан

23 апреля 2018 г.

Задача 1

Пункт а)

Это верно. Пусть s-сумма весов рёбер остовного дерева. Так как колличество рёбер в остовном дереве фиксированное, то s для любого остовного дерева возрастёт на фиксированную величину k. Значит если было верно, что $s \leq g$, где g-сумма весов любого другого дерева, то $s + k \leq g + k$, следовательно минимальное остовное дерево осталось минимальным остовным.

Пункт б)

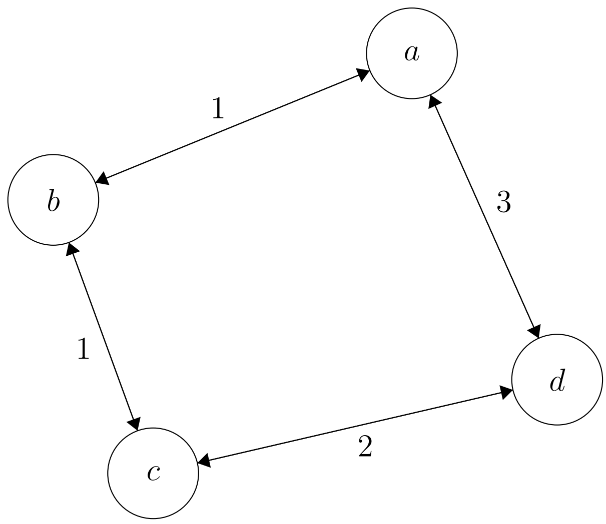
Если самое легкое ребро соединяет две вершины, степень одной из которых равна 1, или же если это ребро является единственным ребром между некоторыми двумя компонентами связности, то случай тривиален. Рассмотрим случай, который не относится к указанным. Пусть несколько ребер, включая данное самое легкое, соединяют две компоненты связности, между которыми уже построено минимальное остовное дерево. Тогда очевидно, что выбрав из них только самое легкое, мы получим минимальное остновное дерево уже для двух этих компонент. При этом из уникальности самой легкой вершины следует, что оно всегда будет входить в наше минимальное остовное дерево.

Пункт в)

Поделим исходный граф на два подграфа, которые соединены некоторыми ребрам. Тогда очевидно, что выбрав самое легкое ребро в этом разрезе, мы получим минимальное остовное дерево для всего графа. Что и требовалось доказать.

Пункт г)

Это не верно. Контрпример:



Минимальное остовное дерево не содержит ребро ad.

Задача 2

Пусть T — минимальное остовное дерево графа G, H — связный подграф G, а H_T - его минимальный остов.

Если ребро (u, v) находится, как в T, так и в H, то это одно из минимальных ребер, соединяющих две вершины в подграфе. Тогда оно обязательно находится в каком-то минимальном остове из всего мн-ва остовов. Тогда начнем строить H_T, просто взяв рёбра, входящие как в T, так и в H. У нас получится какой-то лес компонент связности из неориентированного графа. Достроим его до остова соединив компоненты связности минимальным ребром (u, v), принадлежащим подграфу T, таким, что u и v пренадлежат разным компонентам связности. Полученный граф по построению является минимальным остовом.

Задача 3

Первые $n-1$ операцию проведем $Union$ так, чтобы в результате получить сбалансированное дерево. Оставшиеся операции $Find$ будут требовать $\Omega(\log n)$ для не менее чем половины элементов. В результате получим асимптотику $\Omega((m + 1 - n) \log n)$, которая при достаточно большом m переходит в искомое $\Omega(mlogn)$.

Задача 4

Воспользуемся модернизированным алгоритмом Кускала. Для каждой вершины $u \in U$ выбрем соседа, до которого наименьшее расстояние и соеденим эти вершины ребром. Мы это можем с увереностью сделать. Покажем это: Допустим после этой операции получилось, что одна из вешин оказалась недостижима, то есть её никак не связать с другими, если оставлять все вершины u листьями. Но если бы мы выбрали ребро, которое соединяется с этой вершиной, то тогда оказалось бы, что мы не можем достигнуть вершину, до которой минимальное расстояние. Значит в конце алгоритма нужно проверить остались ли вершины не соединённые рёбрами, если да, то такое дерево построить невозможно - это критерий существования такого дерева. Первый шаг можно сделать за O(|V|). Теперь воспользуемся алгоритмом Крускала, но не будем брать рёбра, которые соединяются с терминальными вершинами. Если эти компоненты окажутся связными, то мы построим минимльное требуемое дерево. Это верно так алгоритм Крускала добавляет на каждом шагу безопасное ребро с минимальным весом(Следствие 23.2 Корман 3-е Издание). Если число итераций цикла в алгоритме меньше чем|V|-1, то такого дерева не существует. Итого время работы: $O(|V| + |E| \log |V|) = O(|E| \log |V|)$.