Домашнее Задание по алгоритмам №11

Павливский Сергей Алексеевич , 873

21.04.2019

Задача 1.

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:

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

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

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

Опредедение. 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.

Решение

Задача 2.

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.

Решение

Рассмотрим произвольное минимальное остовное дерево U подграфа H . Рассмотрим первое ребро (u1, v1) , которое содержится как в T , так и в H . Рассмотрим разрез графа G , который получается следующим образом : мы убираем ребро (u1, v1) , тогда дерево T распадается на компоненты связности , которые и будут определять разрец . Заметим , что это ребро (u1, v1) является минимальным в данном разрезе . Теперь рассмотрим этот разрец для подграфа H . Заметим , что при данном разрезе есть ребро (x, y) входящее в U и соединящее вершины из разных компонент для подграфа H , причем оно минимального веса , т.к. иначе минимальное остовное дерево для графа H можно получить удалением ребра (x, y) и добавлением ребра (u1, v1). Таким образом , мы получили , что ребра (x, y) и (u1, v1) являются минимальными в данном разрезе. Тогда мы можем заменить (x, y) на (u1, v1), получив новое минимальное остовное дерево для подграфа H, но уже содержащее ребро (u1, v1). Проделав данный алгоритм для всех ребер (ui, vi), получим миниимальное остовное дерево , содержащее все ребра (ui, vi), в подграфе H.

Задача 3.

Рассмотрим алгоритм Union-Find без улучшения со сжатием путей. Приведите последовательность из m операций Union и Find над множеством из n элементов, которая потребует времени Ω(m log n).

Решение

Задача 4.

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.

Решение

Модифицируем алгоритм Крускала следующим образом . Для каждой вершины из U заведем счетчик , указывающий на степень этой вершиеы в текущем положении . На ётапе проверки концов ребра будем также смотреть следующее : если ровно один из концов реюра лежит в U , то для этой вершины проверяем ее степень , если она еще равна 0 , то ребро можно добавить , и счетчик для данной вершины делаем равным 1 , если нет , то пропускаем данное ребро . Если ребро соединяло две вершины из U , то данное ребро добавлять нельзя . Заметим , что это не влияет на алгоритм ,т.к. выполняется константное количес во операций . В конце мы определяем , если получилось одно множество , то мы собрали нужный граф , все вершины из U будут висячими , т.к. их степени будут равны 1 . Если же мы получили больше одного множества , то выдаем ответ "таких деревьев нет". Заметим , что если такое "остовное"дерево существует , то алгоритм найдет его в силу корректности алгоритма Крускала . Ассимптотика алгоритма Крускала равна O(E $logV$), что и требовалось .