1 Задание 11

1.1 Задача 1

а) После добавления ко всем ребрам w вес любого остовного дерева станет $s + (n-1)w$, где s - вес этого дерева до модификации. (В каждом остовном дереве n вершин и $n-1$ ребро. Поэтому после модификации минимальное остовное дерево останется минимальным, так как ко всем остовным деревьям прибавили одну и ту же константу.

Ответ: правда

б) Допустим противное и минимальное ребро не входит в минимальное остовное дерево. Тогда добавим его - получим цикл, в котором есть это ребро a, выкинув любое другое ребро из этого цикла, получим остовное дерево меньшего веса - противоречие.

Ответ: правда

в) Допустим противное, ребро e не является самым легким ребром пересекающий некоторый разрез, тогда выкинем его и добавим ребро минимального веса, пересекающее этот разрез, тогда получится остовное дерево меньшего веса - противоречие. (Как будет выглядеть это на картинке: удалили ребро из исходного дерева - стало две компоненты связанности - взяли ребро наименьшего веса, соединяющее их - получили остовное дерево меньшего веса.)

Ответ: правда

г) Приведем контр-пример, рассмотрим граф на 3 вершинах: $|AB| = 2, |AC| = 2, |BC| = 3$. Тогда минимальное остовное дерево будет: $B - A - C$. Кратчайший путь между B и C - ребро $|BC|$ и оно не лежит в минимальном остовном дереве.

Ответ: неправда

Задача 2

Воспользуемся алгоритмом Крускала для графа G - получим минимальное остовное дерево для графа G. Потом воспользуемся алгоритмом Крускала для подграфа H, причем отсортируем ребра так, чтобы ребра в отсортированном массиве образовывали подпоследовательность отсортированного массива (нужно для того, чтобы мы в массиве для H брали ребра(которые есть) в той же последовательности, как и в G). Заметим, что если ребро e было добавлено в первом случае, то оно будет добавлено и во втором. Допустим противное, в первом случае оно добавлено, а во втором нет. Это означает, что во втором случае мы пытаемся соединить вершины из одной компоненты связанности, но это значит, что и в первом случае мы бы пытались соединить вершины из одной компоненты связанности, так как в первом случае есть те же самые ребра до e, добавляющие эти вершины в одну компоненту связанности - противоречие. Значит ребра, входящие, как в T, так и в H, входят в некоторое минимальное остовное дерево подграфа H. (Заметим, что все минимальные остовные деревья графа H получаются перестановкой ребер одинакового веса в отсортированном массиве ребер - значит наше доказательство верно для любого остовного дерева графа H.)

1.2 Задача 3

Разделим множества на пары: четная вершина и нечетная - объединим их, так, чтобы нечетная вершина была корнем. Опять разделим на пары и т. д. (каждый раз подаем в $Union$ корни множеств. Получим бинарное дерево, тогда время затраченное на объединение: $\frac{n}{2} + \frac{n}{4} + ... + \frac{n}{2^{log_2 n}} = \Theta(n)$. Затем для четных элементов будем вызывать $Find$. Так как они лежат в листьях, то время каждого запроса $\Theta(logn)$, а время всех запросов $\Theta((m-n)logn)$. Тогда суммарное время работы $\Theta(n)+\Theta((m-n)logn) = \Theta(n+(m-n)logn) = \Theta(mlogn) = \Omega(mlogn)$, так как $m > n$.

1.3 Задача 4

(Мой алгоритм не обрабатывает граф на 2 вершинах, но если он был связанным, то остовным деревом он будет сам, если не был связным, то таких деревьев нет.) Заведем двумерный массив размера V на 2. Если i вершина принадлежит U, то в массив на i месте первой строчки записываем 1, иначе 0, вторую строчку заполняем нулями. Запускаем алгоритм Крускала, но с некоторым уточнением: каждый раз когда рассматриваем ребро, смотрим:

1. Если две вершины принадлежат U, то переходим к следующему ребру;
2. Если только одна вершина принадлежит U, то смотрим на значение во второй строчке для этой вершины, если там стоит 0, то пользуемся алгоритмом Крускала и меняем 0 на 1, если 1, то переходим к следующему ребру;
3. Если ни одна вершина не принадлежит U, то пользуемся алгоритмом Крускала.

Корректность: следует из корректности алгоритма Крускала, только мы запретили ребра из U в U и больше одного ребра в вершину из U, понятно, что надо брать минимальное по весу.

Асимптотика: Время работы алгоритма Крускала: $O(|E|log|V|)$, а время сравнений $O(|V|)$, то есть время сравнений $O(|E|log|V|)$.