Тураев Тимур, 504 (SE)
23-25.11.2013

11.1. Построить массив частот по дереву такой, что дерево по массиву частот будет совпадать с исходным с точностью до поворота.

Можно сразу сказать, когда массив частот построить нельзя: когда дерево не бинарное и когда оно не полное.

Если дерево не бинарное, все просто – бинарное (построенное алгоритмом Хаффмана) не может быть равно небинарному.

Можно доказать, что дерево, не являющееся полным не может соответствовать оптимальному коду Хаффмана: по той простой причине, что если у узла всего один потомок, то все его поддерево можно поднять на 1 уровень наверх, сократив тем самым коды символов на 1 в этом поддереве. Значит, это дерево было не оптимальным.

В остальных случаях массив строится так: сначала за один проход DFS определяем глубину всех узлов в дереве, причем для удобства можно считать ,что самый глубокий узел лежит на глубине 1, следующий за ним – 2 и так далее. Далее, каждому листу на уровне i ставим в соответствие частоту 2^i. *Под частотой тут понимается то количество раз, который этот символ встретился в строке. Перевод в обычные частоты очевиден: делим «частоту» на длину строки.*

Построение дерева кодов Хаффмана по такому массиву частот будет в точности описывать наше дерево: сначала 2 самых редких символа объединятся в дерево на 2-х листах, затем из очереди частот выберутся еще 2 самых редких частоты и на них построится дерево и так далее.

11.2. Perfect matching за линию.

Можно заметить, что все листья обязаны входить в совершенное паросочетание. Значит, можно пометить все такие вершины, затем удалить их из графа и свести задачу к предыдущей. Если на каком-либо шаге мы попытаемся удалить вместе с помеченной вершиной ее непомеченных детей, то можно сразу сказать, что паросочетаний нет.

Еще можно заметить, что если совершенное паросочетание существует, то оно единственно (работает только в деревьях). Действительно: листья выбираются единственным образом, удаляются тоже: задача сводится к аналогичной.

Значит, задачу можно решать «ленивой динамикой» (не уверен, что это действительно ленивая динамика, но вообще очень похоже): сделаем DFS, попросим у текущей вершины найти все паросочетания в ее поддеревьях. Далее, после того как DFS из текущей вершины отработал, посмотрим на ее дочерние вершины: если есть хотя бы одна непомеченная дочерняя вершина, то помечаем ее и ее родителя (текущую): таким образом сформировали паросочетание. Если есть еще непомеченные, можем сразу выйти и сказать, что совершенного паросочетания нет.

Если после всех итераций DFS остались непомеченные вершины (а это может быть, вообще говоря, только корень дерева), то совершенного паросочетания нет.

11.3. Дан граф G, его MST T и подграф H. Доказать, что пересечение T и H содержится в некотором MST графа H.

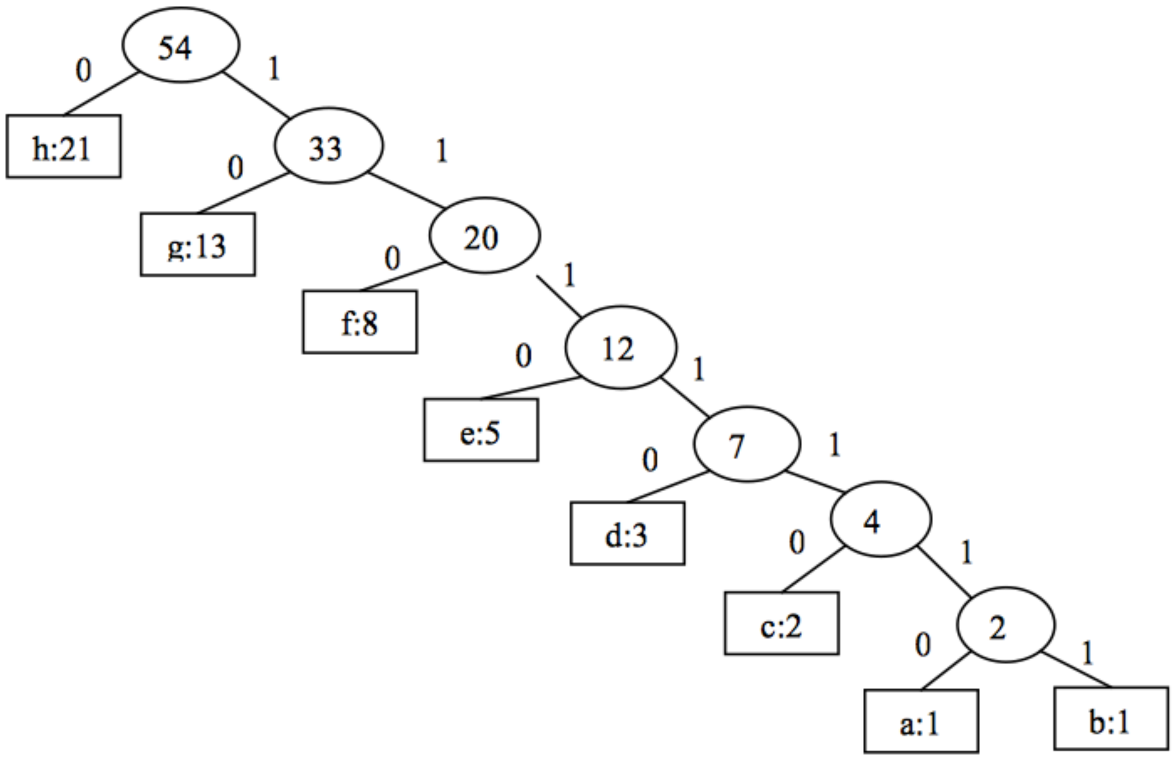
Пока нет.

11.4. Самое длинное кодовое слово на алфавите из n символов.

Самое длинное слово может быть длиной n-1: когда дерево представляет собой одну ветку длиной n-1. Такое может произойти когда частоты символов представляют собой последовательные числа Фибоначчи.

Доказать, что именно n-1 является самым длинным из возможных кодов тоже несложно: так как дерево кодов Хаффмана полное бинарное (см. Задачу 1), то чтобы оно было максимальным по глубине (именно глубина узла определяет длину кодового слова), необходимо, чтобы это дерево представляло собой одну длинную ветку, где у каждого узла 2 дочерних: лист и «ветка» длиной на единицу меньше.

Рассмотрим на примере: a:1, b:1, c:2, d:3, e:5, f:8, g:13, h:21:



Заметим, что значение в узле дерева есть сумма первых k чисел Фибоначчи.
Также известна формула:

$$\sum_{i=1}^{k} F_i = F_{k+2} - 1$$

Откуда следует, что сумма первых k чисел Фибоначчи строго меньше k+2-го числа и не меньше k+1-го. Отсюда следует, что дерево будет строиться именно так (к поддереву, содержащему первые k чисел Фибоначчи прикрепится k+1 число и так далее).

11.5. На листочке.