Summary on Lecture 16, February 12, 2016

Weighted Trees

A weighted tree is a finite rooted tree (T, v) in which each leaf is assigned a weight (i.e. a non-negative number) of this leaf.

Let T have t leaves whose weights are $w_1, w_2, \ldots, w_t$. We loose no generality if we assume that

$$0 \le w_1 \le w_2 \le \cdots \le w_t.$$

We will label the leaves by their weights, and will refer to a leaf by its weight. Let $\ell_1, \ell_2, \ldots, \ell_t$ be the corresponding levels of the leaves. Then the weight $W(T)$ of the tree T is defined as the sum:

$$W(T) = \sum_{i=1}^{t} w_i \ell_i.$$

Example. (a) The six leaves of the weighted tree in Fig. 1(a) have weights 2, 4, 6, 7, 7, and 9. Thus $w_1 = 2$, $w_2 = 4$, $w_3 = 6$, $w_4 = 7$, $w_5 = 7$, and $w_6 = 9$.

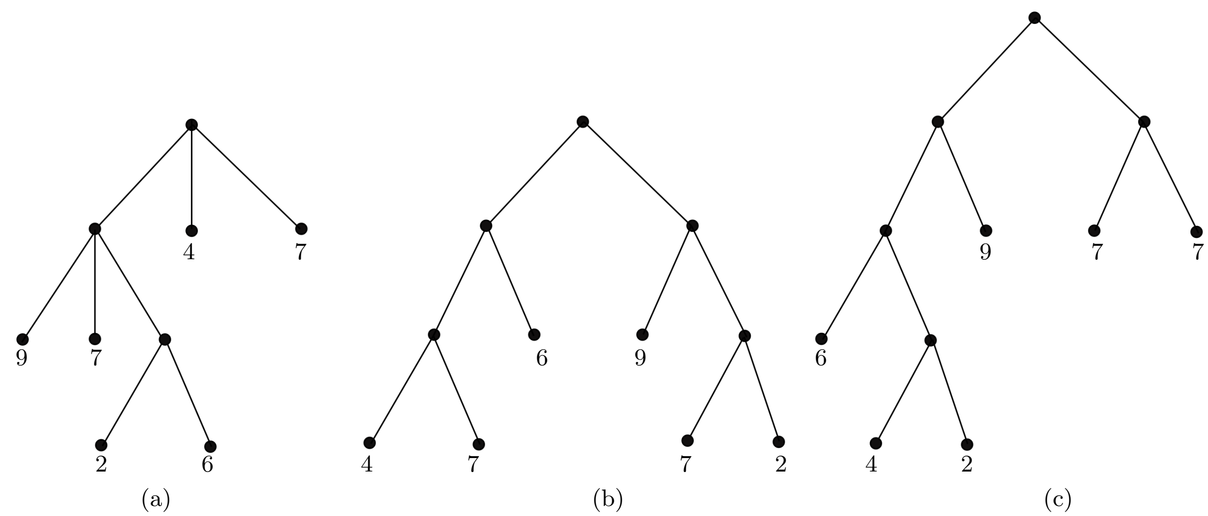


Fig. 1.

There are two leaves labeled 7, and it does not matter which we regard as w_4 and which we regard as w_5. For definiteness, we let w_4 represent the leaf labeled 7 at level 2. Then the level numbers are $\ell_1 = 3$, $\ell_2 = 1$, $\ell_3 = 3$, $\ell_4 = 2$, $\ell_5 = 1$, and $\ell_6 = 2$. Hence

$$W(T) = \sum_{i=1}^{6} w_i \cdot \ell_i = 2 \cdot 3 + 4 \cdot 1 + 6 \cdot 3 + 7 \cdot 2 + 7 \cdot 1 + 9 \cdot 2 = 67.$$

(b) The same six weights can be placed on a binary tree, as in Fig. 1(b) for instance. Now the level numbers are $\ell_1 = 3$, $\ell_2 = 3$, $\ell_3 = 2$, $\ell_4 = \ell_5 = 3$, and $\ell_6 = 2$, so

$$W(T) = \sum_{i=1}^{6} w_i \cdot \ell_i = 2 \cdot 3 + 4 \cdot 3 + 6 \cdot 2 + 7 \cdot 3 + 7 \cdot 3 + 9 \cdot 2 = 90.$$

(c) Fig. 1(c) shows another binary tree with these weights. Its weight is

$$W(T) = \sum_{i=1}^{6} w_i \cdot \ell_i = 2 \cdot 4 + 4 \cdot 4 + 6 \cdot 3 + 7 \cdot 2 + 7 \cdot 2 + 9 \cdot 2 = 88.$$

The total weight is less than that in part (b), because the heavier leaves are near the root and the lighter ones are farther away.

Remark. We are often interested in **binary trees with minimum weight.** If we omit the **binary requirement** and allow many weights near the root, as in part (a), then we can get the lowest possible weight by placing all the weights on leaves at level 1 so that $W(T) = \sum_{i=1}^{6} 1 \cdot w_i$. For weights 2, 4, 6, 7, 7, and 9, this gives a tree of weight 35. Such weighted trees **are not interesting for us**, since they won't help us solve any interesting problems.

Merge and Sort. Consider a collection of sorted lists, say $L_1, L_2, \ldots, L_n$. For example, each list L_i could be an alphabetically sorted mailing list of clients or a pile of exam papers arranged in increasing order of grades. To illustrate the ideas involved, we suppose that each list is a set of real numbers arranged by the usual order "$\le$. Also we suppose that we can merge lists together **only two at a time** to produce new lists. Our problem is to determine how to merge the n lists most efficiently to produce a single sorted list. Two lists are merged by comparing the first numbers of both lists and selecting the smaller of the two (either one if they are equal). The selected number is removed and becomes the first member of the merged list, and the process is repeated for the two lists that remain. The next number selected is placed second on the merged list, and so on. The process ends when one of the remaining lists is empty.

Let $|L|$ be the length of the list L. We notice that to merge two lists L and L', it takes at most $|L| + |L'| - 1$ comparisons. Indeed, let $L = \{4, 8, 9\}$ and $L' = \{3, 6, 10, 11\}$. We compare first two numbers and select the smaller of the two: here we get 3 and put it on new list $L \cup L' := \{3\}$, and then we compare new remaining lists $L = \{4, 8, 9\}$ and $L' = \{6, 10, 11\}$. Then we choose 4, and redefine $L \cup L' := \{3, 4\}$, $L = \{8, 9\}$ and $L' = \{6, 10, 11\}$, and so on. Thus here we will need at most $3 + 4 - 1$ comparisons (we do not need to compare the last number).

We observe that a merging of n lists in pairs involves $n - 1$ merges. Suppose, for example, that we have five lists L_1, L_2, L_3, L_4, and L_5 with $|L_1| = 15$, $|L_2| = 22$, $|L_3| = 31$, $|L_4| = 34$, and $|L_5| = 42$ and suppose that they are merged as it is shown in Fig. 2 (a):

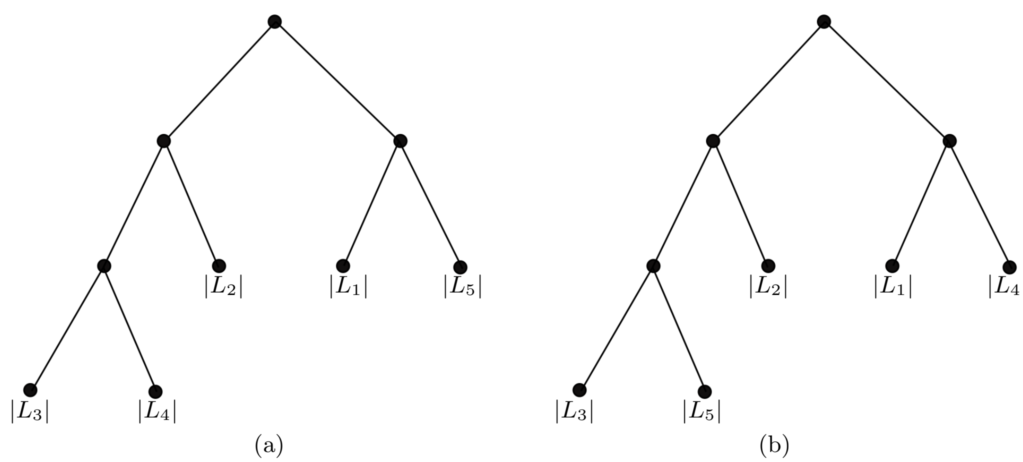


Fig. 2.

Then we find a total number of camparisons to create a final list $L = L_1 \cup L_2 \cup L_3 \cup L_4 \cup L_5$:

$$W(T) - 4 = 2 \cdot |L_1| + 2 \cdot |L_2| + 3 \cdot |L_3| + 3 \cdot |L_4| + 2 \cdot |L_1| - 4 = 2 \cdot 15 + 2 \cdot 22 + 3 \cdot 31 + 3 \cdot 34 + 2 \cdot 42 - 4 = 349.$$

On the other hand, if we use the merging scheme given by Fig. 2 (b), we get the following total:

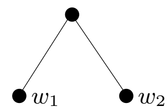
$$W(T) - 4 = 2 \cdot |L_1| + 2 \cdot |L_2| + 3 \cdot |L_3| + 3 \cdot |L_4| + 2 \cdot |L_1| - 4 = 2 \cdot 15 + 2 \cdot 22 + 3 \cdot 31 + 2 \cdot 34 + 3 \cdot 42 - 4 = 363.$$

Thus if we use a binary weighted tree T with n leaves and weights $|L_1|, \ldots, |L_n|$, the we need at most $W(T) - (n - 1)$ comparisons.

Let $L = (w_1, \ldots, w_t)$ be a list of weights. We say that a binary weighted tree T is **optimal for the weights** $L = (w_1, \ldots, w_t)$ if $W(T) \leq W(T')$ for any weighted tree T' with the same weights $L = (w_1, \ldots, w_t)$.

Here is the algorithm to find an optimal tree for a given list of weights:

Huffman$(L = \{w_1, w_2, \ldots, w_k\})$:
{`Input: A list of weights:` $L = \{w_1, w_2, \ldots, w_k\}$, $k \geq 2$}
{`Output: an optimal tree` $T(L)$}
`if` $k = 2$ `then`
  `return the tree`



`else`
  `Choose two smallest weights` u `and` v `of` L.
  `Make a list` L' `by removing the elements` u `and` v `and adding the element` $u + v$.
  `Let` $T(L') :=$**Huffman**(L').
  `Form a tree` $T(L)$ `from` $T(L')$ `by replacing a leaf of weight` $u + v$
  `by a subtree with two leaves of weights` u `and` v.

  `return` $T(L)$.

Now we returen to the example above to merge the lists L_1, L_2, L_3, L_4, and L_5 with $|L_1| = 15$, $|L_2| = 22$, $|L_3| = 31$, $|L_4| = 34$, and $|L_5| = 42$. We run the algorithm **Huffman**$(L = \{15, 22, 31, 34, 42\})$ and we get the following weighted tree:

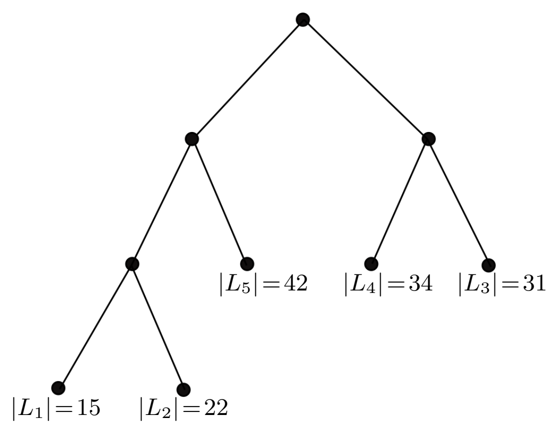


Fig. 3.

We get the the following total number of comparisons:

$W(T) - 4 = 3 \cdot |L_1| + 3 \cdot |L_2| + 2 \cdot |L_3| + 2 \cdot |L_4| + 2 \cdot |L_1| - 4 = 3 \cdot 15 + 3 \cdot 22 + 2 \cdot 31 + 2 \cdot 34 + 2 \cdot 42 - 4 = 321.$