CSE 207 Class Test 2 Answers with Brief Explanation

May 11, 2011

Q1: Find optimal cost of matrix multiplication with dimensions: 5,10,3,12,5,50,6.

Ans: I am not showing the complete calculations here. But the final results are: optiaml cost= $m_{1,6} = 2010$, $k = 2$, and multiplication sequence $((A_1, A_2)((A_3, A_4)(A_5, A_6)))$.

Q2. Find optimal Huffman coding for $A : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21$.

Ans: Combine a and b, then combine this result to c, then combine this result to d, and so on, until combine the result with h. Put 0 in the left and 1 in the right. Traverse each leaf from root and collect the 0's and 1's to make the binary code of the leaf.

Q3. Show a sequence of total m MAKE SET, UNION and FIND operations on n elements with only heuristic used be **union by rank** such that they take $\Omega(m \log n)$ time.

Ans: I am writing the solution without any figure. You must add figures when you answer this type of question.

Remember the definition of **union by rank**: connect two trees such that the root of the smaller node becomes the new root.

At first apply n MAKE SET operations to create n single-note trees as follows:

MAKE SET (1), MAKE SET (2), ... MAKE SET(n)

Then apply $n/2$ UNION operations by taking two subsequent trees for each UNION from the above n trees as follows:

UNION (1,2), UNION (3,4), UNION (5,6), ..., UNION(n-1,n)

This will make $n/2$ trees, each with 2 nodes and height 1. Then apply $n/4$ UNIONs by taking two subsequent trees from the above $n/2$ trees as follows:

UNION (1,4), UNION (5,8), UNION (9,12), ..., UNION(n-3,n)

This will make $n/4$ trees, each with 4 nodes and height 2. Then apply $n/8$ UNIONs by taking two subsequent trees from the above $n/4$ trees as follows:

UNION (1,8), UNION (9,16), UNION (17,24), ..., UNION(n-7,n)

Repeat this procedure until there is only one tree of n nodes. Observe that at every step we reduce the number of tree by half and increase the size of each tree by double. So, after $\log n$ steps we shall end up with only one tree with height $\log n$.

Until now, there are $n + n/2 + n/4 + n/8 + \ldots + 1 = \Theta(n)$ UNION operations.

Then perform the remaining $(m - n)$ FIND operations. Each FIND will take "at least $\log n$"$= \Omega(\log n)$ time in worst case. So total minimum cost: minimum cost of MAKE SET + FIND + UNION = $\Omega(n) + \Omega(n) + \Omega((m - n) \log n) = \Omega(m \log n) + O(n) - \Omega(n \log n)$. Since $m > n$, this becomes $\Omega(m \log n)$.

We assume that n is power of 2. But if that is not the case, then still we are fine, because we are in order notation.