CSE331: Introduction to Algorithms
Notes on Lecture 8: Solving Recurrences II

Antoine Vigneron

July 23, 2021

1 The master method: Example 6

As we saw in class, the master theorem does not cover the following recurrence relation:

$$T(n) = 2T(n/2) + n \log n.$$

We now show how to solve it using the recursion tree method. To simplify the proof, we assume that n is a power of 2.

Let h denote the height of the recursion tree. When we go down one level in the tree, the number of nodes increases by a factor 2, and the size of the nodes decreases by a factor 2. (See Figure 1.) So at depth i, the tree has 2^i nodes and each node has size $n/2^i$. Therefore, the total cost at depth i is

$$2^i \cdot \frac{n}{2^i} \log \frac{n}{2^i} = n \log \frac{n}{2^i} = n(\log(n) - i).$$

whenever $0 \leqslant i < h$. At depth h, there are n leaves, so the cost is $nT(1)$. As the leaves have size 1, we have $2^h = n$ and thus $h = \log n$.

So the total cost is given by:

$$\begin{aligned}
T(n) &= nT(1) + \sum_{i=0}^{\log(n)-1} n(\log(n) - i) \\
&= nT(1) + n \log^2 n - n \sum_{i=0}^{\log(n)-1} i \\
&= nT(1) + n \log^2 n - n \frac{\log(n)(\log(n) - 1)}{2} \\
&= \frac{n}{2}(\log^2 n + \log n + 2T(1)).
\end{aligned}$$

As $T(1) = o(\log^2 n)$ and $\log n = o(\log^2 n)$, we have $\log^2 n + \log n + 2T(1) = \Theta(\log^2 n)$. It follows that $T(n) = \Theta(n \log^2 n)$.

Alternate proof. Another way of proving that $T(n) = \Omega(n \log^2 n)$ is the following. Consider the top half of the recursion tree, that is, levels 0 to $h/2 = \log(n)/2$. The total cost at each levels $0 \leqslant i \leqslant \log(n)/2$ is $n(\log(n) - i)$, which is at least $n \log(n)/2$. So the total cost of the top half of the recursion tree is at least $(1 + \log(n)/2)n \log(n)/2 = \Omega(n \log^2 n)$.

Conversely, the cost at each level is at most $n \log n$, and there are $1 + \log n$ levels, so the total cost is $T(n) = O(n \log^2 n)$. As we just proved that $T(n) = \Omega(n \log^2 n)$, it means that $T(n) = \Theta(n \log^2 n)$.

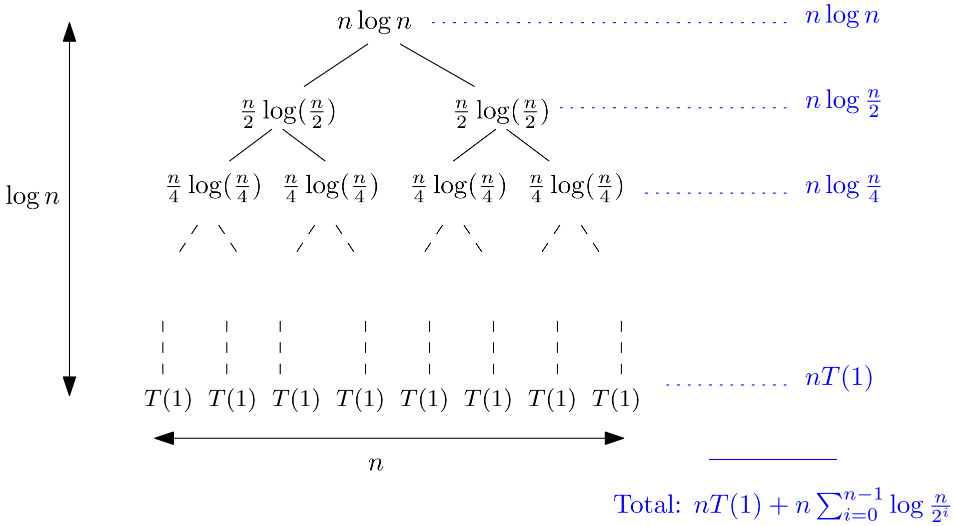

Figure 1: Recursion tree for $T(n) = 2T(n/2) + n \log n$.