EECS 3101 - B Tutorial 3 Notes

Shahin Kamali

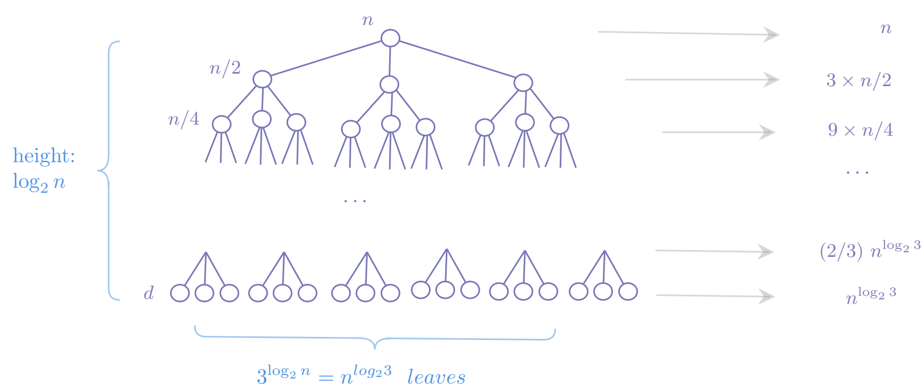
York University

September 2022

1. Draw the recursion tree and analyze it to determine the asymptotic complexity of the following recursive function:

$$f(n) = \begin{cases} 1 & \text{if } n = 1 \\ 3f(n/2) + n \end{cases}$$

Answer: The recursion tree is depicted below:



Note that it is a "leaf-hevay" tree, that is, the majority of work is done at the leaves. You can see that the amount of work at the leaf level is $n^{\log_2 3}$ which is asymptotically larger than the amount of work at the root. For the time complexity, we can write (summing the work on each level in a bottom-up approach):

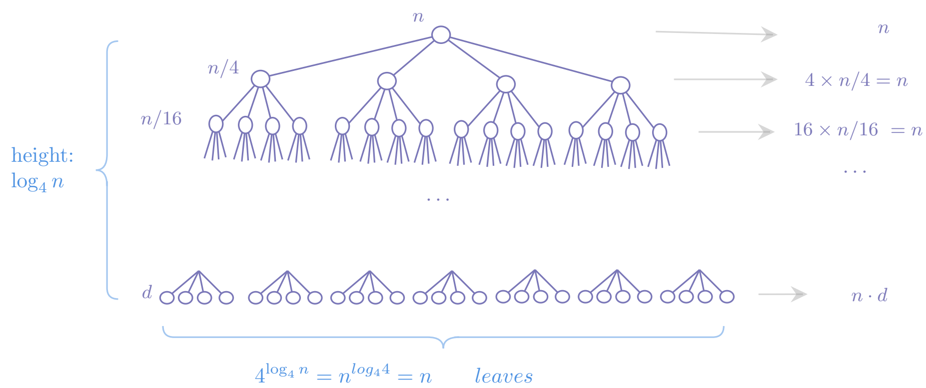
$$\begin{aligned} f(n) &= n^{\log_2 3}d + 2/3n^{\log_2 3} + \ldots + 9n/4 + 3n/2 + n \\ &\leq n^{\log_2 3}(1 + 2/3 + 4/9 + \ldots) \\ &= 3n^{\log_2 3} \end{aligned}$$

Therefore, the value of $f(n)$ is at least $n^{\log_2 3}$ and at most $3n^{\log_2 3}$. Thus $f(n) = \Theta(n^{\log_2 3})$. Note that, when the tree is "leaf-heavy" like this case, we are in Case 1 of the Master theorem.

2. Draw the recursion tree and analyze it to determine the asymptotic complexity of the following recursive function:

$$f(n) = \begin{cases} d & \text{if } n = 1 \\ 4f(n/4) + n \end{cases}$$

Answer: The recursion tree is depicted below:



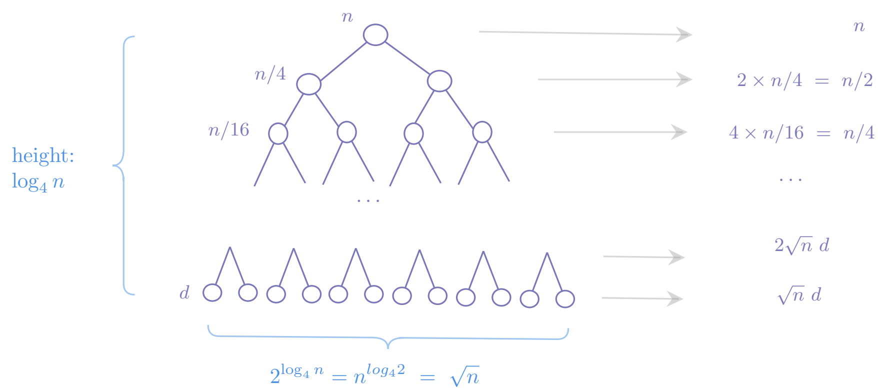
Note that it is a "balanced" tree, that is, the amoun of work done is the same at all levels of the tree, that is n at all levels. Given that there are $\log_4 n$ levels in the tree, we can write $f(n) = \log_4 n \times n = (\log_2 n / \log_2 4) \times n = \Theta(n \log n)$.

Note that, when the tree is "balanced" like this case, we are in Case 2 of the Master theorem and the time complexity is $\Theta(f(n) \times \log n)$, here, $\Theta(n \log n)$.

3. Draw the recursion tree and analyze it to determine the asymptotic complexity of the following recursive function:

$$f(n) = \begin{cases} d & \text{if } n = 1 \\ 2f(n/4) + n \end{cases}$$

Answer: The recursion tree is depicted below:



Note that it is a "root-heavy" tree, that is, the majority of work is done at the root level. You can see that the amount of work at the root level is n which is asymptotically larger than the amount of work at the leaves ($\sqrt{n}$). For the time complexity, we can write (summing the work on each level in a top-down approach):

$$\begin{aligned} f(n) &= n + n/2 + n/4 + \ldots + 2\sqrt{n} + \sqrt{n} \\ &\leq n(1 + 1/2 + 1/4 + \ldots) \\ &= 2n \end{aligned}$$

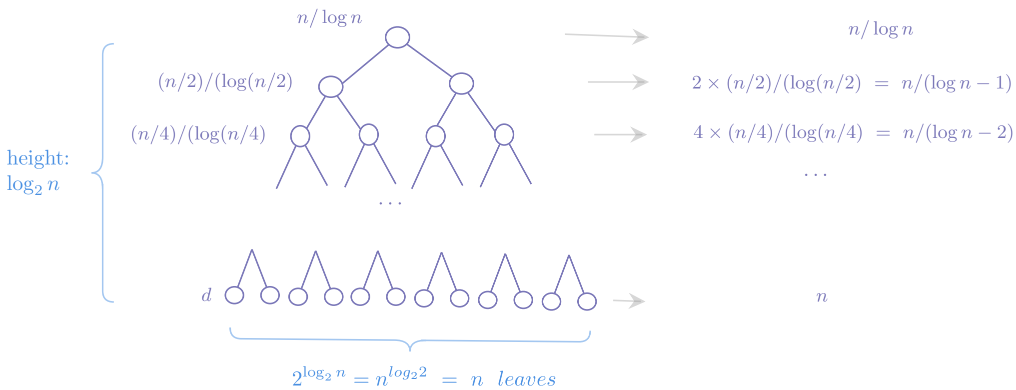
Therefore, the value of $f(n)$ is at least n (the work done at the root) and at most $2n$. Thus $f(n) = \Theta(n)$.

Note that, when the tree is "leaf-heavy" like this case, we are in Case 3 of the Master theorem.

4. Draw the recursion tree and analyze it to determine the asymptotic complexity of the following recursive function:

$$f(n) = \begin{cases} d & \text{if } n = 1 \\ 2f(n/2) + n/\log n \end{cases}$$

Answer: The recursion tree is depicted below:



Note that the amount of work is done at the root level. But we cannot apply the Master theorem in this case; here we have $n^{\log_b a} = n$ and $f(n) = n/\log n$. Now, although $f(n) = O(n)$, we cannot state that $f(n) \in O(n^{1-\epsilon})$ for any positive ϵ. So, we cannot apply Master theorem here.

For the time complexity, we can write:

$$\begin{aligned} f(n) &= \frac{n}{\log n} + \frac{n}{\log n - 1} + \ldots + \frac{n}{3} + \frac{n}{2} + n \\ &= n(1 + 1/2 + 1/3 + \ldots + 1/\log n) \\ &= \Theta(n \log \log n) \end{aligned}$$

Note that $1 + 1/2 + 1/3 + \ldots + 1/x$ is the Harmonic series which is asymptotically equal to $\log(x)$; here $x = \log n$.