COMP 3804 — Solutions Tutorial January 19

Problem 5: Solve the following recurrence using the *unfolding method* that we have seen in class. Give the final answer using Big-O notation.

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ n + 5 \cdot T(n/7) & \text{if } n \geq 7. \end{cases}$$

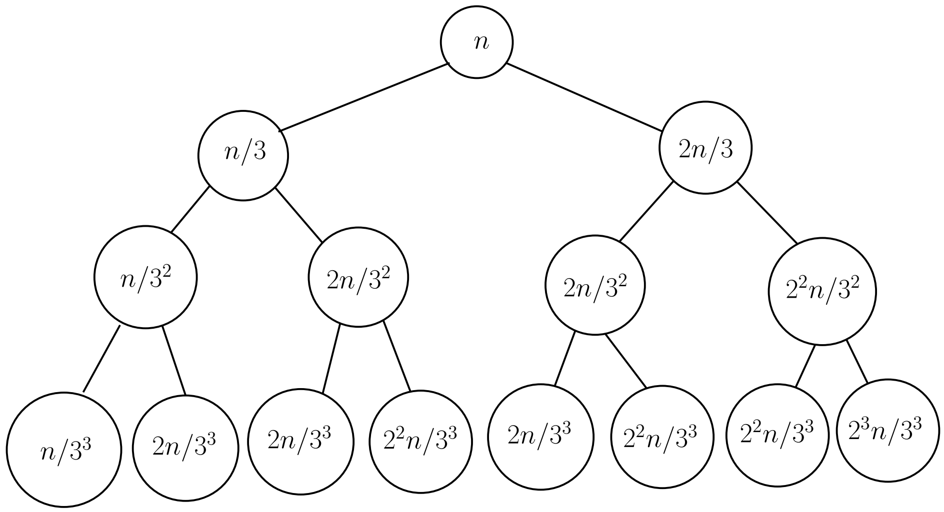
You may assume that n is a power of 7.
Solution: We write $n = 7^k$. Unfolding gives

$$\begin{aligned}
T(n) &= n + 5 \cdot T(n/7) \\
&= n + 5\left(n/7 + 5 \cdot T(n/7^2)\right) \\
&= (1 + 5/7)\, n + 5^2 \cdot T(n/7^2) \\
&= (1 + 5/7)\, n + 5^2\left(n/7^2 + 5 \cdot T(n/7^3)\right) \\
&= \left(1 + 5/7 + (5/7)^2\right) n + 5^3 \cdot T(n/7^3) \\
&= \left(1 + 5/7 + (5/7)^2\right) n + 5^3\left(n/7^3 + 5 \cdot T(n/7^4)\right) \\
&= \left(1 + 5/7 + (5/7)^2 + (5/7)^3\right) n + 5^4 \cdot T(n/7^4) \\
&\;\;\vdots \\
&= \left(1 + 5/7 + (5/7)^2 + \cdots + (5/7)^{k-1}\right) n + 5^k \cdot T(n/7^k) \\
&= \sum_{i=0}^{k-1} (5/7)^i n + 5^k \cdot T(1) \\
&= \sum_{i=0}^{k-1} (5/7)^i n + 5^k \\
&\leq \sum_{i=0}^{\infty} (5/7)^i n + 7^k \\
&= \frac{1}{1 - 5/7}\, n + n \\
&= \frac{9}{2}\, n \\
&= O(n).
\end{aligned}$$

Problem 6: The function $T(n)$ is recursively defined as follows:

$$T(n) = \begin{cases} 1 & \text{if } 1 \leq n \leq 2, \\ n + T(n/3) + T(2n/3) & \text{if } n \geq 3. \end{cases}$$

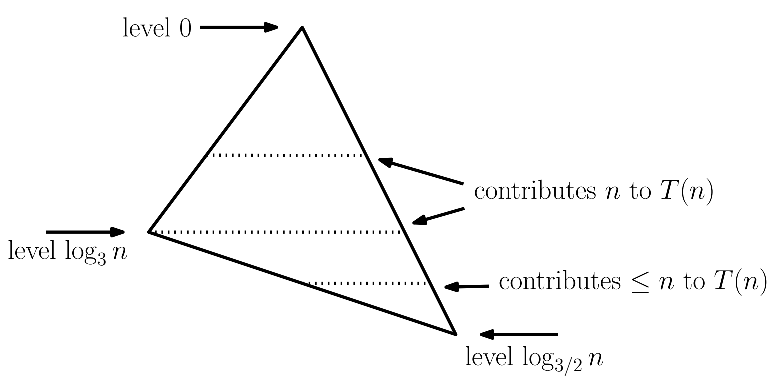
Use the *recursion tree method* that we have seen in class to prove that $T(n) = \Theta(n \log n)$.
Solution: If we draw a few levels of the recursion tree, we get the following:



From this figure, we see that, *if a level is full*, it contributes exactly n to the function $T(n)$. However, since the tree is not perfectly balanced, it is not the case that every level is full.

The path in the recursion tree that starts at the root and always moves to the left child has length $\log_3 n$ (maybe plus or minus 1). Each level, from the root down to the level of the leftmost leaf, is full and contributes n to $T(n)$. Thus,

$$T(n) \geq n \log_3 n = \Omega(n \log n).$$



The path in the recursion tree that starts at the root and always moves to the right child has length $\log_{3/2} n$ (maybe plus or minus 1). Each level, from the root down to the level of the rightmost leaf, contributes at most n to $T(n)$. Thus,

$$T(n) \leq n \log_{3/2} n = O(n \log n).$$