CSE 6331 HOMEWORK 3

BRENDAN WHITAKER

1. *A function $T(n)$ satisfies:*

$$T(n) = \begin{cases} c, & n \leq 1 \\ 3T(\lfloor n/4 \rfloor) + n, & n > 1 \end{cases},$$

where c is a positive constant. Prove that $T(n)$ is asymptotically nondecreasing.

Proof. We use induction. Let $n = 1$. Then

$$T(n+1) = T(2) = 3T(\lfloor 2/4 \rfloor) + 2 = 3T(0) + 2 = 3c + 2 > c.$$

So $T(1) = c \leq 3c+2 = T(2)$. So the base case holds. Now fix n and assume $T(k) \leq T(k+1), \forall k \leq n$. We wish to show that $T(n+1) \leq T(n+2)$. We know $T(n+2) = 3T(\lfloor (n+2)/4 \rfloor) + n + 2$. And $T(n+1) = 3T(\lfloor (n+1)/4 \rfloor) + n + 1$.
Case 1: $n + 1 \equiv 3 \mod 4$. Then we know $(n+2)/4$ is an integer, so let $l + 1 = (n+2)/4$. Then we have: $\lfloor (n+1)/4 \rfloor = l$, and then $T(n+2) = 3T(l+1) + n + 2$, and $T(n+1) = 3T(l) + n + 1$. Now since $l \leq n$, we know $T(l) \leq T(l+1)$. So we have:

$$T(n+1) = 3T(l) + n + 1 \leq 3T(l+1) + n + 2 = T(n+2).$$

Case 2: $n + 1 \not\equiv 3 \mod 4$. Then $\lfloor (n+1)/4 \rfloor = \lfloor (n+2)/4 \rfloor = l \in \mathbb{Z}$. So we have:

$$T(n+1) = 3T(l) + n + 1 \leq 3T(l) + n + 2 = T(n+2).$$

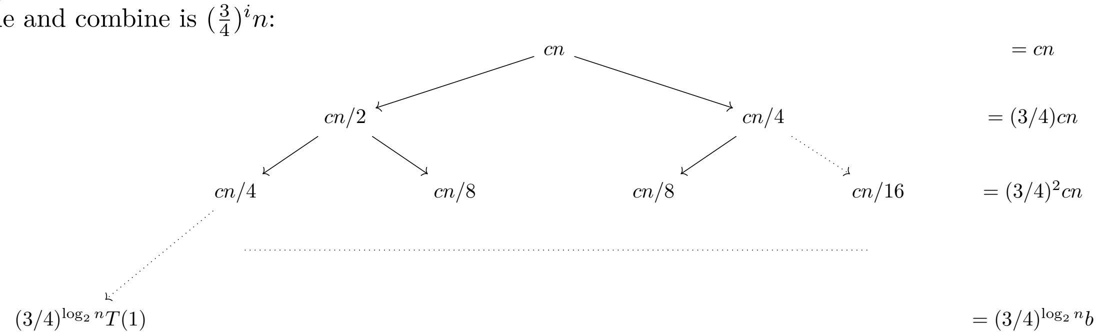
Hence we know $T(n+1) \leq T(n+2)$ in all cases, so by induction this is true for all $n \in \mathbb{N}$. So $T(n)$ is nondecreasing for positive integers. In particular it is asymptotically nondecreasing. □

2. *Determine the tight asymptotic complexity of the following function:*

$$T(n) = \begin{cases} b, & n \leq 3 \\ T(\lfloor n/2 \rfloor) + T(\lfloor n/4 \rfloor) + cn, & n > 3 \end{cases}.$$

We claim $T(n) \in \Theta(n)$.

Proof. Assume n is a power of 4. We use a recursion tree to observe that at each level i, the time to divide and combine is $(\frac{3}{4})^i n$:



Date: SP18.

And since the height of the tree is given by the longest path to a leaf node, which is along the side with all recursive calls to $T(\lfloor n/2 \rfloor)$, we know the height is $\log_2(n)$. So we have:

$$\begin{aligned} T(n) &= cn + \left(\frac{3}{4}\right) cn + \left(\frac{3}{4}\right)^2 cn + \cdots + \left(\frac{3}{4}\right)^{\log_2 n} b \\ &\leq cn \left(1 + \left(\frac{3}{4}\right) + \left(\frac{3}{4}\right)^2 + \cdots \right) \\ &= 4cn \in O(n). \\ T(n) &\geq cn \in \Omega(n). \end{aligned} \tag{1}$$

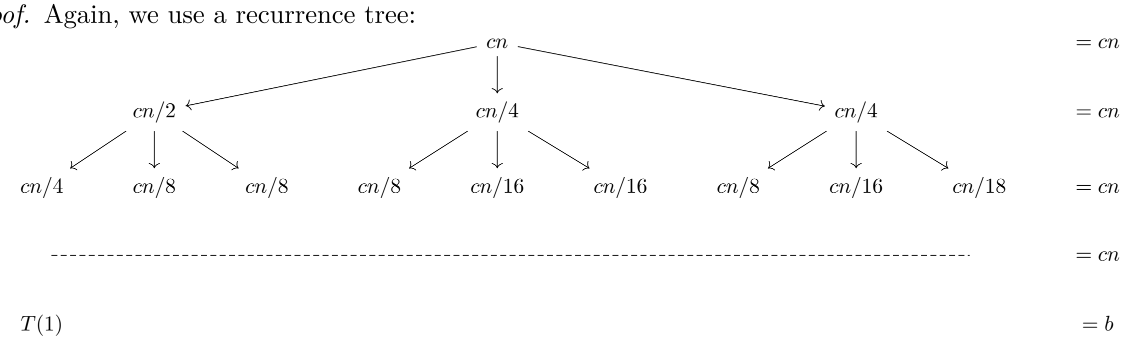
So we know $T(n) \in \Theta(n)$. □

3. *Determine the tight asymptotic complexity of the following function:*

$$T(n) = \begin{cases} b, & n \leq 3 \\ T(\lfloor n/2 \rfloor) + 2T(\lfloor n/4 \rfloor) + cn, & n > 3 \end{cases}.$$

We claim $T(n) \in \Theta(n \log(n))$.

Proof. Again, we use a recurrence tree:



So we have that the time to divide and combine on each level of the tree adds to cn. And the height of the tree is $\log_2 n$ since the longest path to a tree node is along the left hand side, where all the recursive calls are $T(n/2)$. So we have:

$$\begin{aligned} T(n) &= cn + cn + \cdots + b \\ &= \log_2 n(cn) \in \Theta(n \log n). \end{aligned} \tag{2}$$

□

4. *Use the master method to solve the following recurrences.*
   (a) $T(n) = 4T(n/2) + n^2$.
   Note $a = 4, b = 2, f(n) = n^2$, and $n^{\log_2(4)} = n^2 = \Theta(n^2)$. So by the master method, this is **Case 3**, and hence $T(n) \in Theta(n^2 \log n)$.
   (b) $T(n) = 4T(n/2) + n^2 \log^2 n$.
   Note $a = 4, b = 2, f(n) = n^2 \log^2 n$. And $n^{\log_2(4)} = n^2$, and $n^2 \log^2 n \in \Theta(n^2 \log^2 n)$, so this is **Case 4**. Hence $T(n) \in \Theta(n^2 \log^3 n)$.
   (c) $T(n) = 4T(n/2) + n^3$.
   Note $a = 4, b = 2, f(n) = n^3$. And $n^{\log_2(4)} = n^2$. So we know $f(n) = n^3 >> n^2$, and thus this is **Case 2**. So we know $T(n) \in \Theta(n^3)$.
5. *The running time of an algorithm A is described by the recurrence $T(n) = 7T(n/2) + n^2$. A competing algorithm A' has a running time of $T'(n) = aT'(n/4) + n^2$. What is the largest integer value for a such that A' is asymptotically faster than A?*

We apply the master theorem. Note that for $T(n)$, we have $a = 7, b = 2, f(n) = n^2$. And since $n^{\log_2(7)} >> n^2$, we are in **Case 1**, and $T(n) \in \Theta(n^{\log_2(7)})$. For $T'(n)$, we have $a, b = 4, f(n) = n^2$. If we choose $a = 17$, we have $n^{\log_4(17)} >> n^2 = n^{\log_4(16)}$, and since:

$$\log_4(17) < 2.05 < 2.80 < \log_2(7),$$

we know we are in **Case 1** for $T'(n)$ as well, and $T'(n) \in \Theta(n^{\log_4(a)})$. Observe that $\log_4(7^2) = 2\log_4(7) = 2\frac{\log_2(7)}{\log_2(4)} = \log_2(7)$. So when we let $a = 7^2 = 49$, we have $T'(n) = \Theta(T(n))$. So we let $a = 7^2 - 1 = 48$, and then $T'(n) \in \Theta(n^{\log_4(48)})$, so since $n^{\log_4(48)} << n^{\log_2(7)}$, we know 48 is the largest integer a s.t. $T'(n)$ is asymptotically faster than $T(n)$.