Training Problems #3 Solutions

1.

Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\lg n)$.

Solution:

We guess that the solution is $T(n) = O(lg\ n)$. We must prove that $T(n) \leq c lg\ n$ for appropriate $c > 0$. We start by assuming that this bound holds for $\lceil n/2 \rceil$, that is that $T(\lceil n/2 \rceil) \leq c lg(\lceil n/2 \rceil)$. Substituting into the recurrence yields

$$\begin{aligned} T(n) &\leq c lg(\lceil n/2 \rceil) + 1 \\ &\leq c lg(n/2) + 1 \\ &= c lg n - c lg 2 + 1 \\ &= c lg n - c + 1 \\ &= c lg n && \text{if } c \geq 1 \end{aligned}$$

the last step holds as long as $c \geq 1$. For the base case, it suffices* to show that $T(2) \leq c lg 2$ for some $c \geq 1$. $T(2) = T(1) + 1$ or $T(2) = 2$ assuming $T(1) = 1$. Thus $T(2) \leq c lg 2$ if $c \geq 2$.

2. Using substitution, find the Big-O time complexity of the below function:

$$T(n) = \begin{cases} 3T(n-1), if\ n > 0 \\ 1 \qquad\qquad\quad , otherwise \end{cases}$$

Solution:

T(n) = 3T(n-1)

T(n) = 3(3T(n-2)) = 3^2T(n-2)

T(n) = 3^2(3T(n-3))

.

.

T(n) = 3^nT(n-n) = 3^nT(0) = 3^n

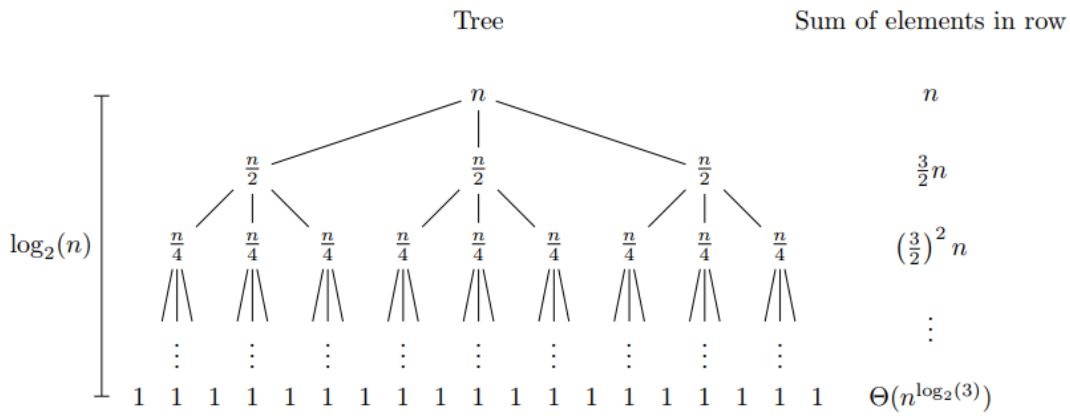
This shows that the complexity of this function is O(3^n). For the boundary condition, we set T(0) = 1, and so T(0) = 1 ≤ c × 3^0, therefore 1 ≤ c .Thus, we can choose n_0 = 0 and c = 1.

3.

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/2 \rfloor) + n$. Use the substitution method to verify your answer.

Solution:

We construct the tree as described in the book, summing all the elements of each row. We make the assumption that n is a power of 2 and we ignore the floor function.



The last row has $(3/2)^{\log_2(n)} n$ elements, but

$$\left(\tfrac{3}{2}\right)^{\log_2(n)} n = n^{\log_2(3/2)} n = n^{\log_2(3) - \log_2(2) + 1} = n^{\log_2(3)},$$

so we get the given result.

Taking the sum of the elements of each row, we get

$$\begin{aligned}
T(n) &= n + \frac{3}{2}n + \left(\frac{3}{2}\right)^2 n + \cdots + \left(\frac{3}{2}\right)^{\log_2(n)-1} n + \Theta(n^{\log_2(3)}) \\
&= \sum_{i=0}^{\log_2(n)-1} \left(\frac{3}{2}\right)^i n + \Theta(n^{\log_2(3)}) \\
&= n \cdot \frac{(3/2)^{\log_2(n)} - 1}{3/2 - 1} + \Theta(n^{\log_2(3)}) \\
&= 2n\left(\frac{3^{\log_2(n)}}{2^{\log_2(n)}} - 1\right) + \Theta(n^{\log_2(3)}) \\
&= \frac{2n \cdot n^{\log_2(3)}}{n} - 2n + \Theta(n^{\log_2(3)}) \\
&= n^{\log_2(3)} - 2n + \Theta(n^{\log_2(3)}) \\
&= O(n^{\log_2(3)}),
\end{aligned}$$

where in the third line we used the formula for a finite geometric series (equation (A.5) on page 1147). This is our asymptotic upper bound on the recurrence $T(n)$. ■

4.

Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n/2) + n^2$. Use the substitution method to verify your answer.

Solution:

This is the tree corresponding to the recurrence:

$$\begin{array}{c|cc} & \text{Tree} & \text{Sum of elements in row} \\ & n^2 & n^2 \\ & | & \\ & \frac{n^2}{4} & \left(\frac{n}{2}\right)^2 \\ \log_2(n) & | & \\ & \frac{n^2}{16} & \left(\frac{n}{4}\right)^2 \\ & | & \vdots \\ & \vdots & \\ & 1 & \Theta(1) \end{array}$$

The last line, by the pattern of the sum of elements in each row, should have $\frac{n^2}{2^{2\log_2(n)}} = \frac{n^2}{n^2} = 1$ element. Taking the sum of the elements of each row, we get

$$\begin{aligned} T(n) &= n^2 + \left(\frac{n}{2}\right)^2 + \left(\frac{n}{4}\right)^2 + \cdots + \left(\frac{n}{2^{\log_2(n)-1}}\right)^2 + \Theta(1) \\ &= \sum_{i=0}^{\log_2(n)-1} \frac{n^2}{2^{2i}} + \Theta(1) \\ &= n^2 \cdot \frac{(1/4)^{\log_2(n)} - 1}{1/4 - 1} + \Theta(1) \\ &= -3n^2 \left(\frac{1}{n^2} - 1\right) + \Theta(1) \\ &= 3n^2 - 3 + \Theta(1) \\ &= O(n^2). \end{aligned}$$