Algorithms: Assignment #3

Tanvi Magdum
magdum.t@northeastern.edu

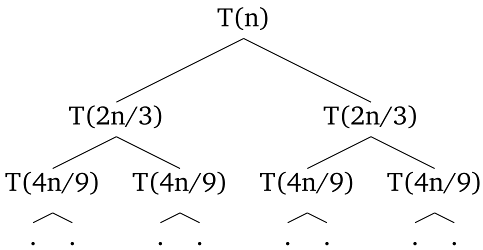
January 30, 2023

Answers

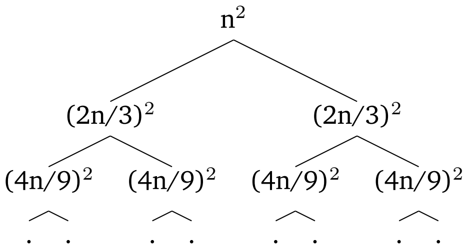
1 Problem 1 Solution :

$T(n) = 2T(\frac{2}{3}n) + n^2$

Using recursion tree method,



Recursion tree after adding the cost of corresponding levels,



We continue expanding each node in the tree by breaking it into its constituent parts as determined by the recurrence. Each level has two time more nodes than level above, so the number of levels at i would be 2^i.

The cost of each node at level i can be represented as $(\frac{2}{3})^{2i}n^2$. Thus the sub-branches hit n=1 when $(\frac{2}{3})^{2i}n^2 = 1$, or $i = \log_{3/2} n$. Thus, the height of the tree is $\log_{3/2} n$.

So, the cost of first level would be n^2, second level would be $\frac{8}{9}n^2$, third level $\frac{64}{81}n^2$ and so on. The bottom level at depth $\log_{3/2} n$ has $2^{\log_{3/2} n} = n^{\log_{3/2} 2}$ nodes, each contributing to cost T(1) for a total cost of $n^{\log_{3/2} 2}T(1)$, which is $\Theta(n^{\log_{3/2} 2})$, assuming T(1) is a constant.

So, the total cost for entire tree would be -

$T(n) = n^2 + \frac{8}{9}n^2 + (\frac{8}{9})^2n^2 + ... + (\frac{8}{9})^{\log_{3/2} n-1}n^2 + \Theta(n^{\log_{3/2} 2})$

$T(n) = \sum_{i=0}^{\log_{3/2} n-1}(\frac{8}{9})^i n^2 + \Theta(n^{\log_{3/2} 2})$

Let us use infinite decreasing geometric series as upper bound. Hence,

$T(n) < \sum_{i=0}^{\infty}(\frac{8}{9})^i n^2 + \Theta(n^{\log_{3/2} 2})$

$T(n) = \frac{1}{1-(8/9)} n^2 + \Theta(n^{\log_{3/2} 2})$

$T(n) = 9n^2 + \Theta(n^{\log_{3/2} 2})$

$T(n) = O(n^2)$

Thus, the coefficients of n^2 form decreasing geometric series . The sum of these coefficients is bounded by an upper bound of constant 9. The cost of root dominates the cost of total tree here i.e. n^2.

Using Master Theorem,

Comparing the given equation with $T(n) = aT(n/b) + f(n)$,

For this recurrence we have, a=2, b=3/2 and $f(n) = n^2$

We have $n^{\log_b a} = n^{\log_{3/2} 2} = n^{1.7095} = \Theta(n^{\log_{3/2} 2}) = \Theta(n^{1.7095})$

Since, $f(n) = \Omega(n^{\log_{3/2} 2+\epsilon}) = \Omega(n^{1.7095+\epsilon})$, where $\epsilon = 0.2905$,

we can apply Case 3 of Master Theorem and conclude that the solution is $T(n) = \Theta(n^2)$.

2 Problem 2 Solution :

$T(n) = 3T(\frac{n}{2}) + \frac{n}{\log n}$

Using Master Theorem,

Comparing the given equation with $T(n) = aT(n/b) + f(n)$,

For this recurrence we have, a=3, b=2 and $f(n) = \frac{n}{\log n}$

Now we check which of the three cases of the Master Theorem applies.

We have $n^{\log_b a} = n^{\log_2 3} = n^{1.58} = \Theta(n^{\log_2 3}) = \Theta(n^{1.58})$

$n^{1.58}$ will always be greater than $\frac{n}{\log n}$, since the power of former term is greater than the latter one. So f(n) is asymptotically as well as polynomially smaller than $n^{\log_b a}$.

The height of the tree in this case is $n^{\log_2 3}$ and we incurred that the work done at leaves is polynomially more. So leaves are the dominant part, and our result becomes the work done at leaves (Case 1).

Case 1 of Master theorem says, if the cost of solving the sub-problems at each level increases by a certain factor, the value of f(n) will become polynomially smaller than $n \log_b a$. Thus, the time complexity is oppressed by the cost of the last level i.e $n \log_b a$

$\therefore f(n) = O(n^{\log_{3/2} 2-\epsilon})$

Hence according to Case 1, $T(n) = \Theta(n^{\log_{3/2} 2})$

The asymptotic upper bound would be $O(n^{\log_{3/2} 2})$ and lower bound would be $\Omega(n^{\log_{3/2} 2})$ for the given function.

3 Problem 3 Solution :

3.1 Problem 3a Solution -

$T(n) = \sqrt{n}T(\sqrt{n}) + n$

We cannot apply Master theorem here directly because $a = \sqrt{n}$. Thus, a is not constant. We use another method to solve this recurrence.

Changing variables,

Suppose, $n = 2^k$

$\therefore T(2^k) = 2^{k/2}T(2^{k/2}) + 2^k$

Divide by 2^k on both sides,

$\frac{T(2^k)}{2^k} = \frac{2^{k/2}T(2^{k/2})}{2^k} + \frac{2^k}{2^k}$

$\frac{T(2^k)}{2^k} = \frac{T(2^{k/2})}{2^{k/2}} + 1$

Now let $\frac{T(2^k)}{2^k} = S(k)$

$\therefore S(k) = S(k/2) + 1$

Apply Master theorem to this function,

Comparing this equation with $T(n) = aT(n/b) + f(n)$,

For this recurrence we have, a=1, b=2 and $f(k) = 1$

Now we check which of the three cases of the Master Theorem applies.

We have $k^{\log_b a} = k^{\log_2 1} = k^0 = 1$

Thus, according to Case 2 of Master theorem, if the cost of solving the sub-problem at each level is nearly equal, then the value of f(k) will be $k^{\log_b a}$.

$\therefore f(k) = \Theta(k^{\log_b a})$

The height of this tree is $k^{\log_2 1}$ or 1. If the work done at leaves and root is asymptotically the same, then our result becomes height multiplied by work done at any level (Case 2).

Thus, the time complexity will be f(k) times the total number of levels ie. $k^{\log_b a} * \log k$

So, $S(k) = k^{\log_b a}.\log k$ according to Case 2 of Master theorem

$\therefore S(k) = \log k$, since the height of tree $k^{\log_b a} = 1$

Re-substitute $S(k) = \frac{T(2^k)}{2^k}$,

$\frac{T(2^k)}{2^k} = \log k$

$T(2^k) = 2^k \log k$

Now re-substitute $2^k = n$,

$T(n) = n \log \log n$

$\therefore T(n) = \Theta(n \log \log n)$

Thus the asymptotic upper bound for original recurrence is $O(n \log \log n)$ and lower bound will be $\Omega(n \log \log n)$.

3.2 Problem 3b Solution -

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

Using, substitution or induction method and expanding the terms,

We need to find value of $T(n-1)$ first

So, $T(n-1) = 3T(n-1-1) = 3T(n-2)$

Now substitute this value in original function and expand T(n-2) and so on..

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

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

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

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

$T(n) = 3^4T(n-4)$

.

.

$T(n) = 3^kT(n-k)$

Consider that we have reached the end of the tree, i.e. (n-k)=0. It means n=k.

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

Assume at n=0, program does nothing, so we can consider T(0) as some constant, suppose $T(0) = 1$

$T(n) = 3^n$

$\therefore T(n) = \Theta(3^n)$

4 Problem 4 Solution :

Please start with the 'README.txt' file first. Please refer to 'Main.java' for source code in Java for Q4. Input format is provided through 'input.txt' file. Screenshots of outputs for 2 sample inputs are provided as output1 and output2. Test cases are provided in 'test_cases.txt'.