EL9343 Homework 02

Spring 2022

Name: Sagar Patel
NETID: sp5894

Question 01

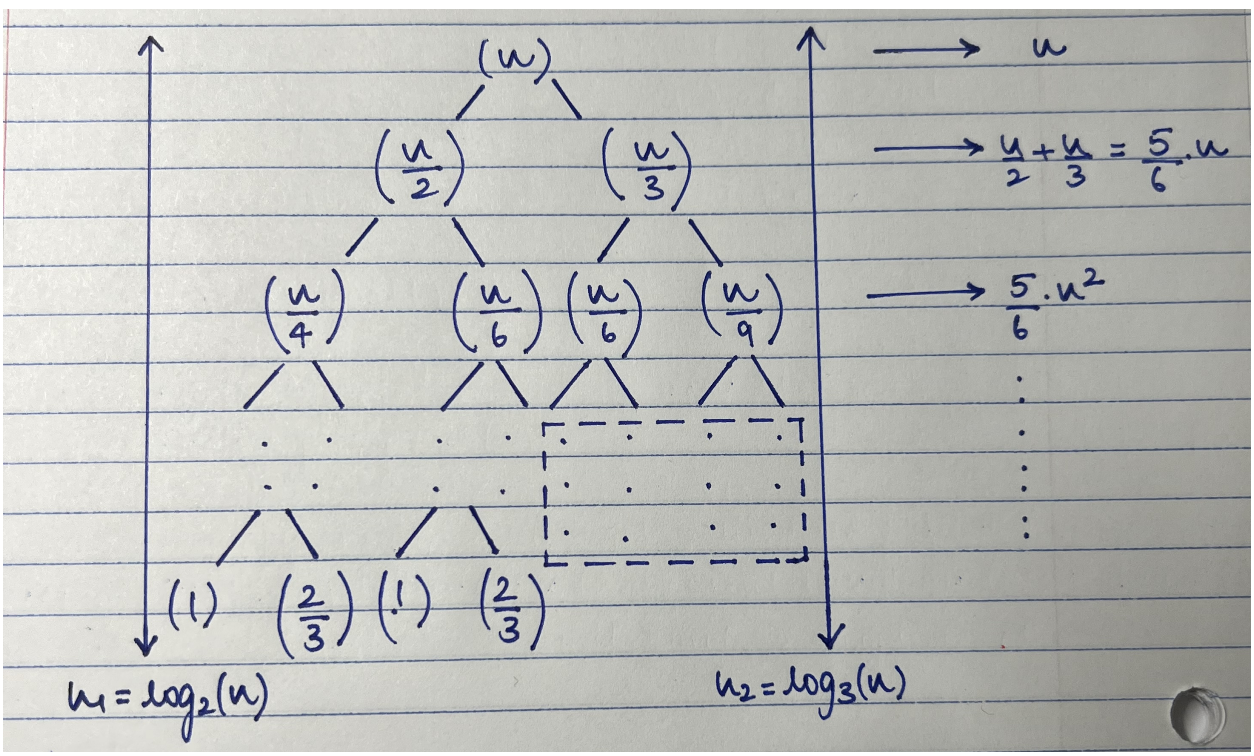
First use the iteration method to solve the recurrence, draw the recursion tree to analyze.

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

Then use the substitution method to verify your solution.

Solution

The running time can be analyzed by drawing the recursion tree —



As we can observe, the right child of the node in this tree has a smaller problem size and that indicates there are some empty nodes in this tree (as shown). We can compute the height of the left side h_1 and right side h_2 of this tree to get the upper bound and the lower bound.

∴ We have,

$c_1 \leq T(n) \leq c_2 \Rightarrow n + \frac{5}{6}.n + ... + (\frac{5}{6})^{log_3 n}.n \leq T(n) \leq n + \frac{5}{6}.n + (\frac{5}{6})^{log_2 n}.n$

Note that → $(a^{log_b n} = n^{log_b a})$ **and that is because** $\Rightarrow \{a^{log_b n} = a^{log_b a^{log_a n}} = a^{log_a n . log_b a} = n^{log_b a}\}$

$c_1 = n + \frac{5}{6}.n + ... + (\frac{5}{6})^{log_3 n}.n \Rightarrow n.(1 + \frac{5}{6} + ... + (\frac{5}{6})^{log_3 n}) \Rightarrow n.\frac{1-(\frac{5}{6})^{log_3 n+1}}{1-(\frac{5}{6})} \Rightarrow 6n$

$.(1 - \frac{5}{6}.n^{log_3 \frac{5}{6}}) \Rightarrow \Theta(n)$

$c_2 = n + \frac{5}{6}.n + ... + (\frac{5}{6})^{log_2 n}.n \Rightarrow n.(1 + \frac{5}{6} + ... + (\frac{5}{6})^{log_2 n}) \Rightarrow n.\frac{1-(\frac{5}{6})^{log_2 n+1}}{1-(\frac{5}{6})} \Rightarrow 6n$

$.(1 - \frac{5}{6}.n^{log_2 \frac{5}{6}}) \Rightarrow \Theta(n)$

$\therefore$ We can say that $T(n) = \Theta(n)$

The upper bound:
--- Induction Hypothesis: $T(k) \leq d.k$ for all $k < n$
$\therefore T(n) = T(\frac{n}{2}) + T(\frac{n}{3}) + n \Rightarrow T(n) \leq (d.\frac{n}{2} + d.\frac{n}{3} + n) \Rightarrow T(n) \leq (d.\frac{5}{6}.n + n)$
In order to satisfy the condition of having $T(n) \leq d.n$, we can set –
$(d.\frac{5}{6}.n + n) \leq d.n \Rightarrow n \leq \frac{d}{6}.n \Rightarrow d \geq 6$
Should we set $d \geq 6$, $T(n) \leq d.n$ and that means $T(n) = O(n)$

The lower bound:
--- Induction Hypothesis: $T(k) \geq c.k$ for all $k < n$
$\therefore T(n) = T(\frac{n}{2}) + T(\frac{n}{3}) + n \Rightarrow T(n) \geq (c.\frac{n}{2} + c.\frac{n}{3} + n) \Rightarrow T(n) \geq (c.\frac{5}{6}.n + n)$
In order to satisfy the condition of having $T(n) \geq c.n$, we can set –
$(c.\frac{5}{6}.n + n) \geq c.n \Rightarrow n \geq \frac{c}{6}.n \Rightarrow c \leq 6$
Should we set $c \leq 6$, $T(n) \geq c.n$ and that means $T(n) = \Omega(n)$

Since, it is implied that $T(n) = \Theta(n)$, we know that the constant efficient is 6.

Question 02

Use the substitution method to prove that $T(n) = 2T.(\frac{n}{2}) + cn.log_2 n$ is $O(n(log_2 n)^2)$

Solution

For $n = 1$, $T(1) = 1$ which is greater than 0 as $d.1.log_2 1 = 0$, so we set the boundary condition as 2 so that our base will be $T(2) \leq d.2.(log_2 2)^2$
--- Induction Hypothesis: $T(k) \leq dk.(log_2 k)^2$ if we have $k < n$ for all values
$T(n) = (2T(\frac{n}{2}) + cn.log_2 n) \Rightarrow T(n) \leq (2d.\frac{n}{2}(log_2 \frac{n}{2})^2 + cn.log_2 n)$
$\Rightarrow T(n) = dn(log_2 n - 1)^2 + cn.log_2 n$
$\Rightarrow T(n) = dn(log_2 n)^2 - 2dn(log_2 n) + dn + cn.log_2 n$

Should we set $(-2dn.log_2 n + dn + cn.log_2 n \leq 0)$, then the statement is smaller than $dn(log_2 n)^2$

$\Rightarrow (2.log_2n - 1).nd \geq cn.log_2n \Rightarrow (2.log_2n - 1).d \geq c.log_2n$
$\Rightarrow d \geq \frac{c.log_2n}{2.log_2n-1} = c.\frac{1}{2-\frac{1}{log_2n}}$ and that's a decrease from +2 to ∞
Now, if we set $d \geq c$, $c.\frac{1}{2-\frac{1}{log_2n}} \leq c.\frac{1}{2-\frac{1}{log_22}} = c$ and we can also see that $T(n) \leq dn.(log_2n)^2$

$\therefore T(n) = O(n(log_2n)^2)$

Question 03

Solving the recurrence:

$$T(n) = 3T(\sqrt[3]{n}) + log_2n$$

(Hint: Making change of variable)

Solution

If we have $m = log_2n$, the equation will be —
$\Rightarrow T(2^m) = 3T.(2^{\frac{m}{3}}) + m$
Replacing $T(2^m)$ with $P(m)$, the equation will be —
$\Rightarrow P(m) = 3P(\frac{m}{3}) + m$
We can make use of the master method to the get the solution of $P(m)$ as $P(m) = \Theta(m.logm)$ and we substitute it back to the result.
$T(n) = T(2^m) \Rightarrow P(m) = \Theta(m.logm)$
$\Rightarrow \Theta(m.logm) = \Theta(logn * log(logn))$

Question 04

You have three algorithms to a problem and you do not know their efficiency, but fortunately, you find the recurrence formulas for each solution, which are shown as follows:

A. $T(n) = 3T(\frac{n}{3}) + \Theta(n)$
B. $T(n) = 2T(\frac{9n}{10}) + \Theta(n)$
C. $T(n) = 3T(\frac{n}{3}) + \Theta(n^2)$

Please give the running time of each algorithm (In Θ notation), and which of your algorithms is the fastest (You probably can do this without a calculator)?

Solution

--- Algorithm A
We know the following —
$f(n) = \Theta(n), a = 3, b = 3, d = 1, log_ba = log_33 = 1 = d$
$\therefore T(n) = \Theta(n^{log_ba} * logn) = \Theta(n.logn)$

--- Algorithm B

We know the following –

$f(n) = \Theta(n)$, $a = 2$, $b = \frac{10}{9}$, $d = 1$, $log_b a = log_{\frac{10}{9}} 2 > 1 = d$

$\therefore T(n) = \Theta(n^{log_b a}) = \Theta(n^{log_{\frac{10}{9}} 2})$

--- Algorithm C

We know the following –

$f(n) = \Theta(n^2)$, $a = 3$, $b = 3$, $d = 2$, $log_b a = log_3 3 = 1 < 2 = d$

$\therefore T(n) = \Theta(f(n)) = \Theta(n^2)$

Upon visual inspection, it is safe to state that the **algorithm A** is the fastest.

Question 05

Can the master theorem be applied to recurrence of $T(n) = T(\frac{n}{2} + n^2 . lgn)$? Why does it work or not? Provide the asymptotic upper bound for this recurrence.

Solution

We know the following –

$a = 1$, $b = 2$, $n = log_b a = 0 < n^2 \Rightarrow f(n) = \Theta(n^2 . lgn)$ by satisfying the 3rd case of Master Theorem.

Should $a = 4$, $b = 2$, the solution is given as shown –

$t(N) = T(\frac{n}{2}) + n^2 . lgn$

$\therefore f(n) = n^2 . lgn$, $a = 1$, $b = 2$

$\Rightarrow n^{log_b a} = n^{log_2 1} = n^0 = 1$

Case 1 and Case 2 do not seem to be applicable for this.

Therefore, for case 3: $f(n) = \Omega(n^{log_b a + \epsilon})$ holds

$\Rightarrow c. f(n) - a. f(\frac{n}{b}) = c. n^2 . lgn - (\frac{n}{2})^2 lg(\frac{n}{2})$

$\Rightarrow c. n^2 . lgn - \frac{n^2}{4}(lgn - lg2)$

The condition $(c - \frac{1}{4}). n^2 . lgn + \frac{n^2}{4} . lg2 \geq 0$ needs to be applied to make it positive for all $n \geq N_0$, where $N_0 \in N$

Now, we get – $c \geq \frac{1}{4} - \frac{1}{4} . \frac{lg2}{lgn}$

$\therefore$ We can make $c < 1$ satisfy $c \geq \frac{1}{4} - \frac{1}{4} . \frac{lg2}{lgn}$

That is for all the values of $n \geq N_0$, where $N_0 \in N$

$\therefore$ The master theorem can be applied, and –

$T(n) = \Theta(n^2 . lgn) = O(n^2 . lgn)$