CIS 675 Homework 1

Design and Analysis of Algorithms

Saket Kiran Thombre

SU ID: 899913802

NetID: sthombre

Email: sthombre@syr.edu

Collaborators for Q1, Q2 and Q3:

1. Saurav Shashank Narvekar

   NetID: sanarvek

2. Chandan Pothumarthi

   NetID: cpothuma

Problem 1: Master Theorem (Point 3)

Consider the following recurrence and prove that master theorem case 3 is not applicable for this recurrence. Give detailed explanation. If the explanation is not detailed, you will not get full point even if the answer is true.

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

Solution: In Condition 1

a = 1, b = 2, f(n) = n(3-sin(n)), First we must prove that f(n) = Ω $(n^{\log_b a+\mathcal{E}})$

Now checking for $n^{\log_b a}$

$n^{\log_2 1}$ = n^0 = 1, Therefore f(n) = Ω $(n^{\mathcal{E}})$.

Since f(n) = n(3-sin(n)), the value of sin always oscillates between 1 and -1. Therefore we can say that n(3-1) < f(n) < n (3-(-1)) which is n(2) < f(n) < n(4).

Therefore, first condition is satisfied for 0 < $\mathcal{E}$ < 1.

In Condition 2,

We need to prove that a * f ()$\frac{n}{b}$ < c * f(n) where c < 1, such that condition holds true for all values of n, where n > 0.

Therefore, by further solving it,

a * f $\left(\frac{n}{b}\right)$ < c * f(n) = 1 * $\frac{n}{2}\left(3 - \sin(\frac{n}{2})\right)$ < c * (3n – n sin(n))

a * f $\left(\frac{n}{b}\right)$ < c * f(n) = $\frac{3n}{2}$ - $\frac{n}{2}$ sin $(\frac{n}{2})$ < c * (3n – n sin(n)) since value of c < 1

Therefore a * f($\frac{n}{b}$) > c * f(n)

This contradiction happens since this is a regulatory violation. This happens when the work done on the lower leaf node might be more than the root node.

Thus, Masters Theorem Case 3 does not work in this case.

Problem 2: Master theorem 3, applies or not? (Point 3)

Following recurrence: T(n) = 2T ($\frac{n}{2}$) + f(n) in which

$$f(n) = \begin{cases} n^3 \; if \; [\log(n)] \; is \; even \\ \quad n^2 \; otherwise \end{cases}$$

Show that f(n) = Ω ($n^{\log_b(a)+\mathcal{E}}$). Explain why the third case of masters theorem stated above does not apply. Prove that T(n) = Θ (n^3) for your recurrence.

Hint: You can use guess and check method. You can consider the base cases T (1) = C1 and T (2) = C2.

Solution: Given equation T(n) = 2T ($\frac{n}{2}$) + f(n), a =2, b = 2, $n^{\log_b a}$ = $n^{\log_2 2}$ = n Thus f(n) = Ω ($n^{1+\mathcal{E}}$)

Master theorem Case 3 has the following condition for all values of n.

a * f ($\frac{n}{b}$) < c * f(n) where c <1. Now let's consider value of n as 8,

2* f ($\frac{8}{2}$) $\leq$ c*f (8)

2*f (4) $\leq$ c*f (8)

2* $4^3 \leq$ c*8^2

Therefore 128 $\leq$ c*64 where c < 1.

The condition is no longer applicable. Hence, we can say that Masters theorem cannot be applied to this.

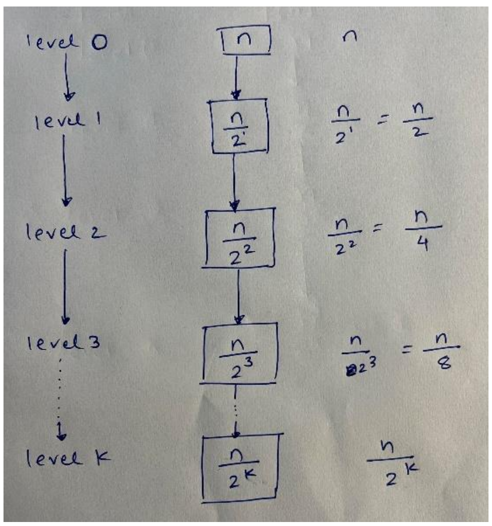
Problem 3: Recurrence: Expansion or Tree approach (Point 4)

Solve the following recurrence and prove that T(n) = Θ(n). (Need to mathematically explain all the steps)
T(n) = T(n/2) + n, where n is a power of 2. That means, let n = 2 k for integer k > 0.

You can only use expansion or tree approach. You cannot use induction or master theorem here.

Solution: T(n) = T $(\frac{n}{2})$ + n, a = 1, b =2, fn = (n)

Solving the equation by tree method. Since the value of a = 1, the tree will be divided similarly.



Now aggregating the terms, we get the equation

T(n) = $\left[n + \frac{n}{2} + \frac{n}{2^2} + \frac{n}{2^3} + \cdots + \frac{n}{2^k}\right]$

Therefore, solving further we get

T(n) = $n + n\left[\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \cdots + \frac{1}{2^k}\right]$

The Geometric Progression of finite series formula is,

$\frac{a(1-r^n)}{(1-r)}$, a = $\frac{1}{2}$, r = $\frac{1}{2}$, n = k

Where, a = first term, r = common ratio.

Solving using the geometric progression, we get

T(n) = n +n $\left[\frac{1}{2}\left[\frac{1-\left(\frac{1^k}{2}\right)}{1-\frac{1}{2}}\right]\right]$

T(n) = n + n $\left[\frac{1}{2}\left[1 - \left(\frac{1^k}{2}\right)\right]\right]$

T(n) = n + n $\left[1 - \frac{1^k}{2}\right]$

Since n = 2k

T(n) = n+(n-1)

T(n) = 2n-1

Since in asymptotic notations, we only consider the dominant terms for Big-oh Notation and do not consider the constants. Therefore **T(n) = O(n)**.

For Lower bound, we should not consider other terms other than n. Therefore, it is **T(n) = $\Omega(n)$.**

Thus, it is proved that **T(n) = Θ(n)**.