DESIGN AND ANALYSIS

OF

ALGORITHMS

Assignment #1 (spring 2018)

Pragathi Thammaneni

16230695

Sridevi Mallipudi

16251191

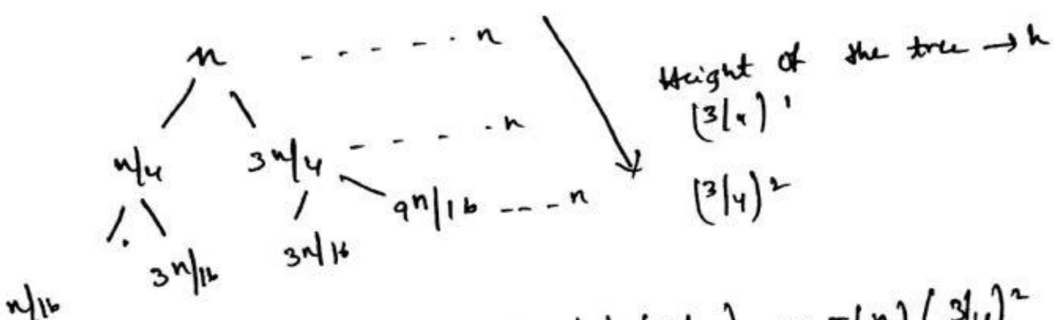
Design And Analysis of Algorithms.

<u>Assignment - 1</u>

1) Calculate $T(n)$ and $O(n)$ for the (Quick sort) - Average & worst case.

Answer: Average case for Quick sort:

The average gets into 3 to 1 split which is (25% - 75%)



$$T(9n/16) = T(3n/4)(3/4) = T(n)(3/4)^2$$

$$\vdots$$

$$T(1) = T(3/4)^k (n)$$

$$(3/4)^k n = 1$$

$$n = (4/3)^k$$

$$\log n = \log (4/3)^k = k \log(4/3)$$

$$K = \log n / \log(4/3) \Rightarrow \log_{4/3}(n) \Rightarrow \text{Height}$$

$$\boxed{T(n) = T(n/4) + T(3n/4) + n}$$

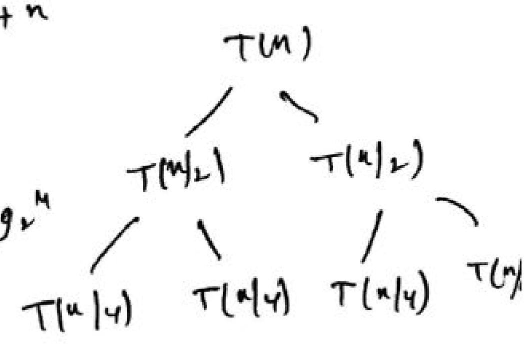
for the 1st case, second case n comparisons are required

Big oh $\boxed{O(n) = n \log_{4/3}(n)}$

If the split is made for 50%. (half)

$$T(n) = 2T(n/2) + n$$

Height $K = \log_2 n$



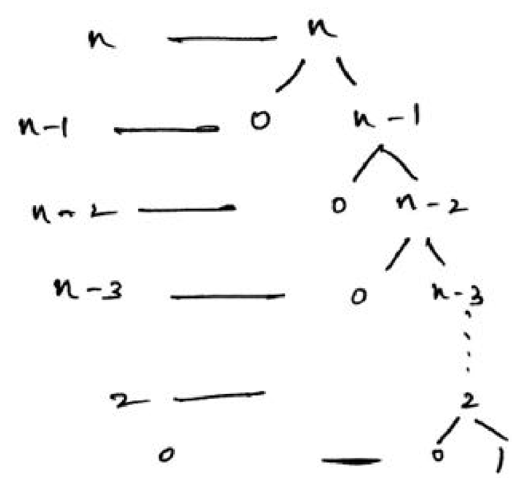
Then for this case the

Average case will be

$$\boxed{O(n) = n\log n} \rightarrow O(n\log n)$$

<u>Worst case:</u>

In this scenario the two sets of split will be into 0 elements and $(n-1)$ elements.



worst case of Quicksort

$$T(n) = n + (n-1) + (n-2) + \cdots + 2 + 0$$

$$T(n) = (1 + 2 + 3 + \cdots + (n-2) + (n-1) + n) - 1$$

$$T(n) = \frac{n(n+1)}{2} - 1$$

$$\boxed{T(n) = \frac{n^2+n}{2} - 1}$$

so the $\boxed{O(n) = n^2}$ — $O(n^2)$.

2) use substitution, summation (or) Recursion tree method to solve

a) $T(n) = 2T(n/2) + n\log n \qquad T(1) = \Theta(1)$

Ans: The best case would be in the first attempt

$$\boxed{\Omega(n) = n\log n}$$

The worst case can be demonstrated as shown below

Height $\Rightarrow \log n$

$T(n)$ ——— $n\log n$

$\frac{n}{2}\log \frac{n}{2} \rightarrow T(n/2)$ $\quad \frac{n}{2}\log \frac{n}{2} \rightarrow T(n/2)$ ——— $n\log n/2$

$\frac{n}{4}\log \frac{n}{4} \rightarrow T(n/4)$ $\quad T(n/4)$ $\quad T(n/4)$ $\quad T(n/4)$ ——— $n\log n/4$

——— $n\log n/2^h$

$$n\log n/2 = n(\log n - \log 2) = n\log n - n$$

$$n\log n/4 = n(\log n - \log 4) = n\log n - 2n$$

$\vdots$

$$n\log n/2^h = n(\log n - \log 2^h) = n\log n - hn \quad (\because h \rightarrow \text{height})$$

$$T(n) = (n\log n - n) + (n\log n - 2n) \text{-----} + (n\log n - hn)$$

$$T(n) = h(n\log h) - n(1+2+3 --- + h) + \Theta(1)$$

$$T(n) = h(nh) - n\left(\frac{h(h+1)}{2}\right) + 0$$

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

$$T(n) = \frac{nh^2 - nh}{2}$$

$$O(n) = nh^2 \Rightarrow \boxed{O(n) = n(\log n)^2} - O(n\log^2 n)$$

Average case $\theta(n)$ is when it satisfies $O(n)$ and $\Omega(n)$

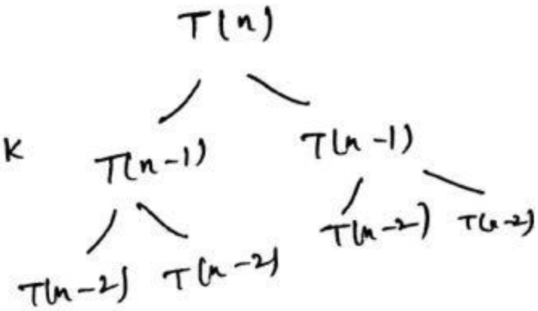
$$\boxed{\therefore\ \theta(n) = n\log n} \;\; \theta(n\log n)$$

This satisfies both $\Omega(n) = n\log n$ and

$$O(n) = n(\log n)^2$$

2 b) $T(n) = 2T(n-1) + 5^n$

$T(0) = 8$



Ans:-

Height = K

$n - K = 1$

$\Rightarrow K = n-1$

$$T(n) = 2T(n-1) + 5^n$$

$$T(n-1) = 2T(n-2) + 5^{n-1}$$

$$T(n-2) = 2T(n-3) + 5^{n-2}$$

$$\vdots$$

$$T(n-K) = 2T(n-K) + 5^{n-K}$$

$$T(n) = 5^n + 2(5^{n-1}) + 2^2(5^{n-2}) + 2^3(5^{n-3}) + \cdots$$
$$\cdots + 2^K(5^{n-K}) + T(0)$$

$$T(n) = 2^0 5^n + 2^1 5^{n-1} + 2^2 5^{n-2} + \cdots + 2^K 5^{n-K} + 8$$

$$T(n) = \sum_{i=0}^{K} 2^i 5^{n-i} + 8$$

$$T(n) = \sum_{i=0}^{n-1} 2^i 5^{n-i} + 8$$

$$T(n) = \sum_{i=0}^{n-1} 5^n (2/5)^i + 8$$

$$T(n) = 5^n \left[(2/5)^0 + (2/5)^1 + (2/5)^2 + \cdots + (2/5)^{n-1} \right] + 8$$

$$T(n) = 5^n \left[\frac{1 - (2/5)^n}{1 - (2/5)} \right] + 8$$

$$T(n) = \frac{5^n - 2^n}{(3/5)} + 8$$

order is $\boxed{O(n) = 5^n} \rightarrow O(5^n)$

3) use Master's method for each $T(n)$ of the following recurrence relation. (Assuming $T(n)$ is constant for $n \leq 4$)

a) $T(n) = 9T(n/2) + n^3 \log n$

Ans:

$a = 9$, $b = 2$, $f(n) = n^3 \log n$

$\log_b a = \log_2 9 = 3.17$

Case 1: $f(n) = n^3 \log n = O(n^{3.17 - \epsilon})$

for $\epsilon > 0 \Rightarrow$

Assume $\epsilon = 0.17$ Case 1:

$f(n) = n^3 \log n = O(n^3)$ (satisfied.)

$T(n) = O(n^{\log_b a})$

$T(n) = \Theta(n^{\log_2 9})$

$T(n) = \Theta(n^{\log_2 9}) \Rightarrow O(n^{3.17})$

3(b) $T(n) = 9T(n/3) + n^2$

Ans: $a = 9,\ b = 3,\ f(n) = n^2$

$\log_b^{a} = \log_3^{9} = 2$

Case 1: $f(n) = n^2 = O(n^{2-\epsilon})$ for $\epsilon > 0$
(Not satisfied)

Case 2: $f(n) \Rightarrow n^2 = \theta(n^2)$ (satisfied)

$T(n) = \theta(n^{\log_b^{a}} \log n)$

So, $T(n) = \theta(n^{\log_3^{9}} \log n)$

$\boxed{T(n) = \theta(n^2 \log n)}$

3(c) $T(n) = 6T(n/2) + n^3$

Ans: $a = 6,\ b = 2,\ f(n) = n^3$

$\log_b^{a} = \log_2^{6} = 2.58$

Case 1: $f(n) = n^3 = O(n^{2.58 - \epsilon})$ for $\epsilon > 0$
Not satisfied

Case 2: $f(n) = n^3 = \theta(n^{2.58})$ Not satisfied.

Case 3: $f(n) = n^3 = \Omega(n^{2.58 + \epsilon})$ for $\epsilon > 0$
Assume ϵ would be $\epsilon = 0.42$

$f(n) = n^3 = \Omega(n^3)$ satisfied.

$$a f(n/b) \leq c f(n)$$

$$6 (n/2)^3 \leq c \cdot n^3$$

$$\frac{6n^3}{2^3} \leq c \cdot n^3$$

$$\frac{6n^3}{8} \leq c \cdot n^3$$

$$3/4 \, n^3 \leq c \cdot n^3$$

for $c = 3/4$, $c < 1$, case 3 satisfies

so $T(n) = \Theta(f(n))$

$\therefore$ $\boxed{T(n) = \Theta(n^3)}$