Sri Sai Anusha Gandu

16230560

Design and Analysis of Algorithms.

Assignment #1.

1. Calculate T(n) and O(n) for the following algorithms (Quick Sort) for the average and worst case

```
void quick_sort (int first, int last, std::vector<int>& arr) {
if (last - first > 1) {
    int pivot = partition (first, last, arr);
    quick_sort (first, pivot, arr);
    quick_sort (pivot +1, last, arr);
}
}

int partition (int first, int last, std::vector<int>& arr) {
int up = first +1;
int down = last -1;
do{
    while ((up != last -1) && arr[first] >= arr[up]) {
    ++up;
}
 while (arr[first] < arr[down]) {
  --down;
 }
 if (up< down ) {
```

```
    swap (arr[up], arr[down]);
  }
} while (up < down);
swap (arr[first], arr[down]);
return down;
}
```

Sol. **Average case:**

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

n - - - - - - - $T(n)$

n - - - - - $T(\frac{n}{4})$ $\quad$ $T(\frac{3n}{4})$. $\quad$ Height = h $\quad$ - - - - $(\frac{3}{4})^1$

n - - - $T(\frac{n}{16})$ $\quad$ $T(\frac{3n}{16})$ $\quad$ $T(\frac{3n}{16})$ $\quad$ $T(\frac{9n}{16})$ - - - - $(\frac{3}{4})^2$

$$T(\frac{9n}{16}) = T(\frac{3n}{4})(\frac{3}{4}) = T(n)(\frac{3}{4})^2$$

$\vdots$

$$T(1) = \cancel{T(\frac{3n}{4})} T(\frac{3}{4})^k (n)$$

$$\Rightarrow (\frac{3}{4})^k n = 1$$

$$n = (\frac{4}{3})^k$$

$$\log n = \log (\frac{4}{3})^k = k \log (\frac{4}{3})$$

$$k = \frac{\log n}{\log (4/3)}$$

$$k = \log_{4/3}(n) = \text{Height}$$

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

→ For the first case, n comparisons are required.

→ For the Cecond case, n comparisons are required.

→ This carries on till the height of the tree ($\log_{4/3}(n)$)

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

If the split is made for pivot = 50%.

$$\boxed{T(n) = 2T(n/2) + n.}$$

Height $k = \log_2 n$.

$$T(n)$$

$$T(\tfrac{n}{2}) \quad T(\tfrac{n}{2})$$

$$T(\tfrac{n}{4}) \quad T(\tfrac{n}{4}) \quad T(\tfrac{n}{4}) \quad T(\tfrac{n}{4})$$

→ In this case,

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

Worst case:

→ In this scenario, the two sets of split is 0 elements and $(n-1)$ elements

n	n
$0 \quad n-1$	$n-1$
$0 \quad n-2$	$n-2$
$0 \quad n-3$	$n-3$
$\vdots$	
2	2
$0 \quad 1$	0

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

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

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

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

$$\boxed{O(n) = n^2} \text{ — } O(n^2)$$

8. Use substitution, summation or recursion tree method to solve the following recurrence relations

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

$T(1) = \Theta(1)$

Sol.

The best case would be in the first attempt.

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

The worst case is demonstrated as shown below.

$\boxed{\text{Height} = \log n}$

$T(n)$ —— $n\log n$

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

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

—— $n\log\frac{n}{2^h}$

$$n\log\frac{n}{2} = n(\log n - \log 2) = n\log n - n$$

$$n\log\frac{n}{4} = n(\log n - \log 4) = n\log n - 2n$$

$$\vdots$$

$$n\log\frac{n}{2^h} = n(\log n - \log 2^h) = n\log n - hn$$

$$T(n) = (n\log n - n) + (n\log n - 2n) + \dots + (n\log n - hn)$$

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

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

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

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

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

$$\boxed{O(n) = n(\log n)^2} \text{ — } O(n\log^2 n)$$

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

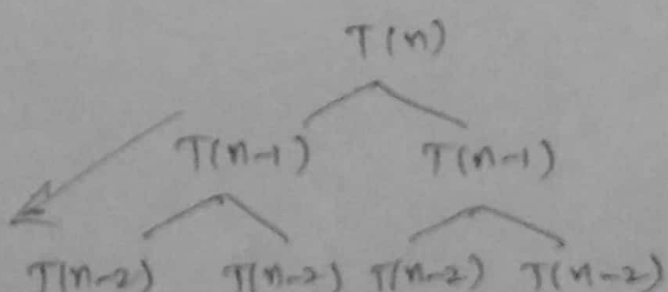
$$\therefore \boxed{\Theta(n) = n\log n} \text{ — } \Theta(n\log n)$$

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

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

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

$T(0) = 8$.



Sol:

Height $= k$

$n - k = 1 \Rightarrow k = n-1$

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

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

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

$$\vdots$$

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

$$T(n) = 5^n + 2(5^{n-1}) + 2^2(5^{n-2}) + 2^3(5^{n-3})$$

$$+ \cdots + 2^k(5^{n-k}) + T(0)$$

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

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

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

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

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

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

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

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

$O(5^n)$

3. Use master's theorem for each $T(n)$ of the following recurrence relations.

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

Sol:

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

$\log_b a = \log_2 9 = 3.17$

Case-1.

$f(n) = n^3 \log n = O(n^{3.17-\epsilon})$ for $\epsilon > 0$

$= O(n^{3.17-0.17})$ $\epsilon = 0.17$

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

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

$\boxed{T(n) = O(n^{3.17})}$

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

Sol:

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

$\log_b a = \log_3 9 = 2.$

Case-1:

$f(n) = n^2 = O(n^{2-\epsilon})$ for $\epsilon > 0$

False.

Case - 2:

$f(n) = n^3 = \Theta(n^3)$ True.

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

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

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

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

$\log_b a = \log_2 6 = 2.58$

Case - 1

$f(n) = n^3 = O(n^{2.58 - \epsilon})$ for $\epsilon > 0$.

False

Case - 2:

$f(n) = n^3 = \Theta(n^{2.58})$

False

Case - 3:

$f(n) = n^3 = \Omega(n^{2.58 + \epsilon})$ for $\epsilon > 0$

$= \Omega(n^{2.58 + 0.42})$ $\epsilon = 0.42$

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

True

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

$$6\left(\frac{n}{2}\right)^3 \leq c \cdot n^3$$

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

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

$$\frac{3}{4} n^3 \leq c \cdot n^3$$

for $c = \frac{4}{5}$, $\frac{3}{4} n^3 \leq c \cdot n^3$ True

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

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