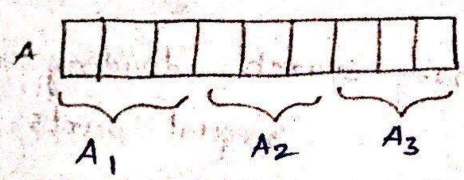
CS 721: Advanced Algorithms & Analysis

Homework-1

1) a) Binary search algorithm

divided into 3 subsets



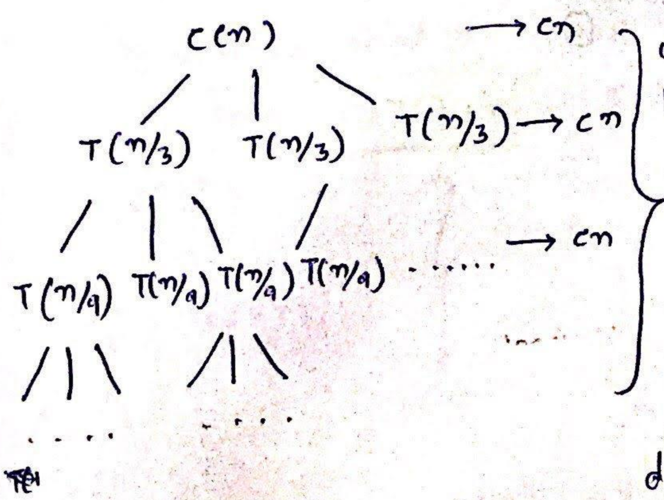
Consider Array 'A' divided into

3 equal subsets A_1, A_2, A_3

∴ Recurrence relation is given as

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

where 2 is no. of comparisions to determine the subset in which search element can be found.



depth ⇒ $\frac{n}{3^k} = 1$

$3^k = n$

$k = \log_3^n$

depth = $\log_3^n$

$T(n) \Rightarrow T(1) + \log_3^n$

$= \Theta(\log_3^n)$

$$\boxed{T(n) = \Theta(\log_3^n)}$$

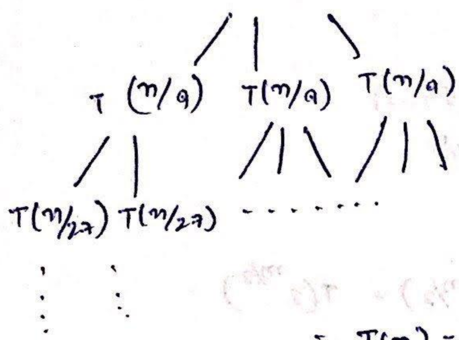
b) Array A is divided into 2 subsets

$A_1 = n/3 \quad A_2 = 2n/3$

(i) Search in smaller subset i.e array with size $n/3$

∴ the recurrence relation is given as

$T(n) = T(n/3) + 1$ where '1' is no. of comparisions



depth $\Rightarrow \frac{n}{3^K} = 1$

$n = 3^K$

$K = \log_3 n$

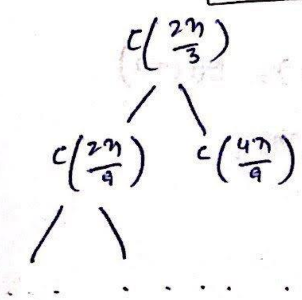
$\therefore T(n) = T(1) + \log_3 n = \theta(\log_3 n)$

$$\boxed{T(n) = \theta(\log_3 n)}$$

(ii) Search in larger subset i.e subset with size $2n/3$

Recurrence relation is given as

$$\boxed{T(n) = T(2n/3) + 1}$$ ['1' is no. of comparisions]



depth $\Rightarrow \frac{n}{(3/2)^K} = 1$

$n = \left(\frac{3}{2}\right)^K$

$K = \log_{3/2} n$

$T(n) = T(1) + \log_{3/2} n$

$$\boxed{T(n) = \theta(\log_{3/2} n)}$$

2) $T(n) = T(n/k) + (k-1)$

a) let $k = \sqrt{n}$

$T(n) = T(n/\sqrt{n}) + (\sqrt{n}-1) = T(\sqrt{n}) + (\sqrt{n}-1)$

let $\boxed{n = 2^m}$ $\Rightarrow$ $\cancel{\log_2 n = m}$ $\sqrt{n} = 2^{m/2}$

now $T(2^m) = T(2^{m/2}) + (2^{m/2}-1)$

let $S(m) = T(2^m)$ then $S(m/2) = T(2^{m/2})$

$S(m) = S(m/2) + (2^{m/2}-1)$ //

$a = 1, b = 2, f(m) = 2^{m/2} - 1$

$m^{\log_b a} = m^{\log_2 1} = 1$

$\lim_{m\to\infty} \frac{f(m)}{m^{\log_b a}} = \frac{2^{m/2}-1}{1} = \infty$

$\therefore$ Case-3 of Master's method

$f(m) = \Omega(m^{\log_b a + \varepsilon})$ then $T(n) = \Theta(f(n))$

$\Rightarrow S(m) = \Theta(f(m)) = \Theta(2^{m/2}-1) \approx \Theta(2^{m/2})$

but $\sqrt{n} = 2^{m/2}$

$S(m) = T(2^m) \Rightarrow \Theta(2^{m/2})$

$T(n) \Rightarrow \Theta\sqrt{n}$ //

b) $T(n) = T(n/\log n) + (\log n - 1)$

here $a = 1$, $b = \log n$

we cannot use master's method because b is not a constant. $[\, T(n) = aT(n/b) + f(n) \,]$ where $a \geq 1$, $b > 1$

2) c) $T_1(n) = \left(\frac{n}{2}\right) + (\log n)$

let $n = 2^m \Rightarrow \frac{n}{2} = \frac{2^m}{2} = 2^{m-1}$

$m = \log_2 n$

$T_1(2^m) = (2^{m-1}) + m$

let $S(m) = T_1(2^m)$

$S(m-1) = T_1(2^{m-1})$

Now, $T_1(2^m) = S(m) = (2^{m-1}) + m$

$S(m) = S(m-1) + m$

$= S(m-2) + (m-1) + m$

$= S(m-3) + (m-2) + (m-1) + m$

......

$= S(0) + 1 + 2 + 3 + \cdots + m$

$= S(0) + \theta(m^2)$

Assuming $S(0) = 1$

$= \theta(m^2)$

$\therefore T_1(n) = \theta(\log^2 n)$ //

2)d) $T_2(n) = \sqrt{n}\left(\frac{n}{\sqrt{n}}\right) + \log n - 1$

$T(n) \geq T_2(n)$

$\therefore T(n) = \Omega(T_2(n))$

$T_2(n) = \sqrt{n} + \log n - 1$

let $\boxed{n = 2^m} \Rightarrow \sqrt{n} = n^{1/2} = 2^{m/2}$

$\log_2 n = m$

$T_2(2^m) = 2^{m/2} + (m-1)$

let $S(m) = T_2(2^m)$

$S(m/2) = T_2(2^{m/2})$

$S(m) = S(m/2) + (m-1)$

$a = 1, \quad b = 2, \quad f(m) = m - 1$

$m^{\log_b a} = m^{\log_2 1} = m^0 = 1$

now, $\lim_{m\to\infty} \frac{f(m)}{m^{\log_b a}} = \lim_{m\to\infty} \frac{m-1}{1} = \infty \Rightarrow \Omega(m^{\log_b a + \varepsilon})$

$\therefore$ Case - 3 of Master's Method

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

$S(m) = \Theta(f(m)) = \Theta(m-1) \approx \Theta(m)$

$= \Theta(\log_2 n)$

$\therefore T_2(n) = \Theta(\log_2 n)$ //

3) a) $f(n) = O((f(n))^2)$

let $f(n) = \frac{1}{n}$

$\therefore \lim_{n\to\infty} \frac{\frac{1}{n}}{\left(\frac{1}{n}\right)^2} = \lim_{n\to\infty} \frac{n^2}{n} = \infty$

because $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq 0$

$\therefore$ False //

b) $f(n) = \Theta(f(\frac{n}{2}))$

$\therefore$ It has to be $f(n) = O\, f(\frac{n}{2})$ & $f(n) = \Omega\, f(\frac{n}{2})$

let $f(n) = 2^n$ $\therefore f(\frac{n}{2}) = \sqrt{2^n}$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{2^n}{\sqrt{2^n}} = \sqrt{2^n} = \infty$

$\therefore$ $f(n)$ is not $\Theta\, f(n/2)$ because $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq c > 0$

$\therefore$ False //

c) $f(n) = \Omega(\sqrt{f(n)})$

let $f(n) = \frac{1}{n}$, $\sqrt{f(n)} = \frac{1}{\sqrt{n}}$

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{1/n}{1/\sqrt{n}} = \frac{1}{n} \times \frac{\sqrt{n}}{1} = \lim_{n\to\infty} \frac{1}{\sqrt{n}} = 0$

$\therefore$ False //

because $\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq \infty$

3) d) $\max(f(n), g(n)) = \Theta(f(n)+g(n))$

i.e
we know that

$$\max(f(n), g(n)) \le f(n)+g(n) \le 2\max(f(n), g(n))$$

$\forall\ f(n), g(n)$ are non -ve

if $f(n)+g(n) \ge \max(f(n), g(n))$

$\therefore \max(f(n), g(n)) = O(f(n)+g(n))$

if $f(n)+g(n) \le 2\max(f(n), g(n))$

$\therefore \max(f(n), g(n)) = \Omega(f(n)+g(n))$

$\therefore$ If it is O and $\Omega \Rightarrow$ it is Θ $\therefore$ True //

let $f(n) = 10n$, $g(n) = n^2$

$\cancel{\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{10n}{n^2}}$ let $n = 5$ $f(n) = 50$, $g(n) = 25$

$\max(f, g) \le f+g \le 2\max(f, g)$

$\max(50, 25) \le 50+25 \le 2\max(50, 25)$

$50 \le 75 \le 100$

$\therefore \max(f(n), g(n)) = \Theta(f(n)+g(n))$

4) $T(n) = 4T(n/2) + n$

need to prove : $T(n) = O(n^2)$

i.e $T(n) \leq cn^2$

Assume $T(n/2) \leq c(n/2)^2$

By substituting,

$$T(n) = 4T(n/2) + n$$
$$\leq 4c\left(\frac{n}{2}\right)^2 + n$$
$$\leq cn^2 + n$$

we need to prove $T(n) \leq cn^2$ not $T(n) \leq cn^2 + n$

$\therefore$ assume $T(n) \leq O(n^2 - n) \approx O(n^2)$

$$T(n) \leq c(n^2 - n)$$

$$T(n) = 4T(n/2) + n$$
$$\leq 4\left[c\left[\left(\frac{n^2}{8}\right) - \left(\frac{n}{2}\right)\right]\right] + n$$
$$\leq 4\left[c\left[\frac{n^2}{4} - \frac{n}{2}\right]\right] + n$$
$$\leq cn^2 - 2cn + n$$
$$\leq cn^2 - cn - cn + n$$
$$\leq c(n^2 - n) - \underbrace{n(c-1)}_{\geq 0}$$

$$\therefore c - 1 \geq 0$$
$$c \geq 1$$

$\therefore \boxed{T(n) = O(n^2)}$