Recurrence Relations

Sorting:- Using Find-Max

Ex:- 1 5 6 4 3 → 1 5 4 3 6

n-1 Comparisons

$T(n)$ = # Comps to Sort array of size n

$= T(n-1) + n-1, \; T(1) = 0$

Substitution method

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

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

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

$\vdots$

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

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

$\frac{n^2}{10} \leq \frac{n^2 - n}{2} \leq 3n^2$

$= \Theta(n^2)$

Best case } For all inputs
Worst case } $T(\cdot) = \Theta(n^2)$.

Can Sorting be done in

$\Theta(n)$, $\Theta(n \log n)$, $\Theta(n\sqrt{n})$

Recurrence Relations

Substitution method:- Change of Variable Technique.

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

$n = 2^m$ $\quad T(2^m) = T(\sqrt{2^m}) + 1$

$$= T(2^{\frac{m}{2}}) + 1$$

$S(m) = T(2^m) \Rightarrow S(m) = S(\frac{m}{2}) + 1$

↳ Recurrence Relation of Binary search

$$= O(\log m)$$

$$S(m) = O(\log m)$$

$$T(n) = T(2^m) = O(\log_2 m)$$

$$= O(\log_2 \log_2 n)$$

If $n = 2^m$ then $m = \log_2 n$.

Recurrence Relations

Substitution Method :- Change of Variable Technique.

$T(n) = T(\sqrt{n}) + n, \quad T(2) = \text{Constant}$

$T(2^m) = T(\sqrt{2^m}) + 2^m$

$S(m) = S(\frac{m}{2}) + 2^m$

Substitution

$= S(\frac{m}{4}) + 2^{\frac{m}{2}} + 2^m$

$= S(\frac{m}{8}) + 2^{\frac{m}{4}} + 2^{\frac{m}{2}} + 2^m$

$\vdots$

$= S(\frac{m}{2^k}) + 2^{\frac{m}{2^{k-1}}} + \cdots + 2^m$

Assume $m = 2^k$

$= S(1) + 2^{\frac{m}{2^{k-1}}} + \cdots + 2^m$

Constant

$2^m + 2^{\frac{m}{2}} + 2^{\frac{m}{4}} + \cdots + 2^2 + 2^1$

$2^m \leq \leq 2 \cdot 2^m$

$= \Theta(2^m)$

$S(m) = \Theta(2^m)$

$T(n) = T(2^m) = \Theta(n)$

Recurrence Tree Method

Binary Search $T(n) = T(\frac{n}{2}) + 1, \; T(1) = 1$

I/P Size Reduction Tree

$n = \frac{n}{2^0}$ Level 0

$\frac{n}{2} = \frac{n}{2^1}$ Level 1

$\frac{n}{4} = \frac{n}{2^2}$ Level 2

$\frac{n}{8} = \frac{n}{2^3}$

$1 = \frac{n}{2^k} = 1$ Level K $(\log_2 n)$

$K = \log_2 n$ $\quad$ # Levels $= \log_2 n + 1$

Computation Tree

1 $\Rightarrow$ 1

$\Rightarrow$ 1

$\Rightarrow$ 1

$\Rightarrow$ 1

$\Rightarrow$ 1

$(\log_2 n + 1) \times 1 = \log_2 n + 1$

$T(n) = 1 + \log_2 n$

Recurrence Tree Method

Ternary Search $T(n) = T(\frac{n}{3}) + 2, \; T(1) = 1$

I/P Size Reduction Tree

$n = \frac{n}{3^0}$ level 0

$\frac{n}{3} = \frac{n}{3^1}$ level 1

$\frac{n}{9} = \frac{n}{3^2}$ level 2

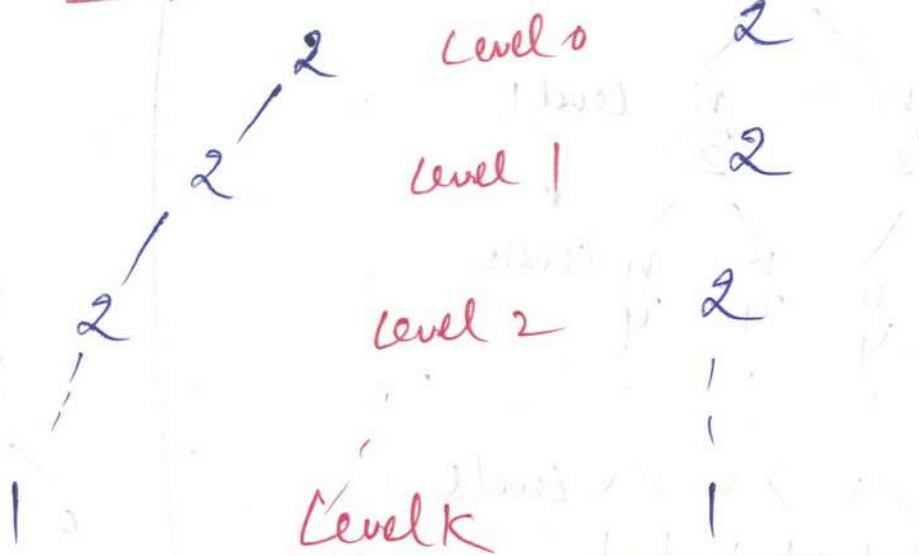
$1 = \frac{n}{3^k}$ Level k

$k = \log_3 n$

#Levels $= k + 1$

$= \log_3 n + 1$

Computation Tree



$(2 * 2 + 2 + \cdots 2) + 1$

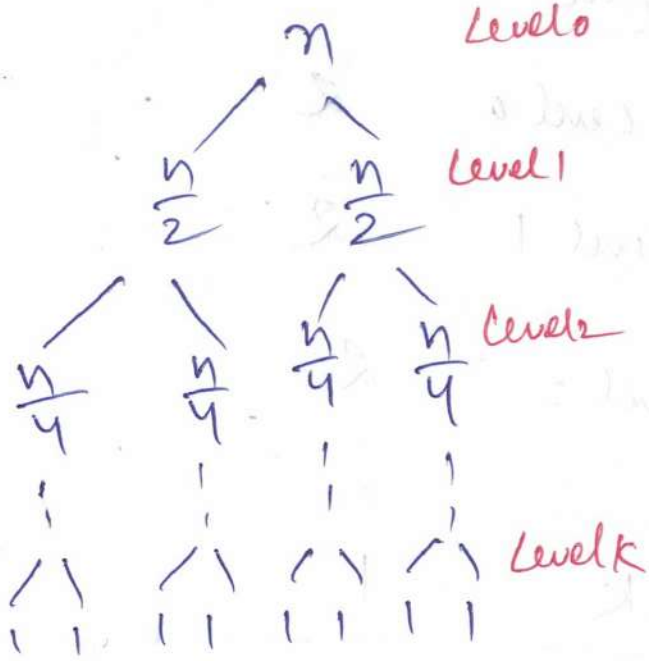
$= 2k + 1 = 2 \log_3 n + 1$

$= O(\log_3 n) = O(\log_2 n)$

$= O(\log_k n)$, k = fixed Integer

Recurrence Tree Method

Find Max:- $T(n) = 2T(\frac{n}{2}) + 1,\ T(1) = 0$ (Divide & Conquer 2-Way)

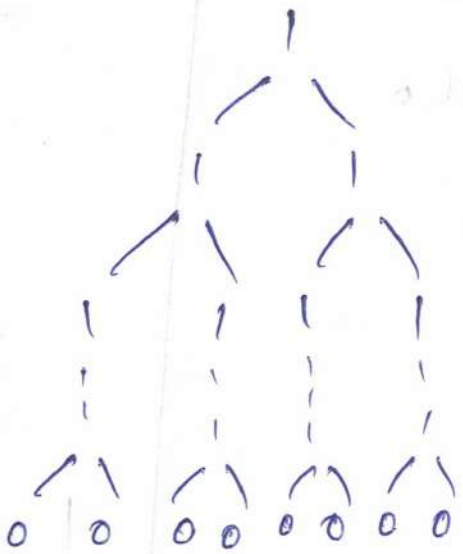


I/p Size Reduction Tree

$$\frac{n}{2^k} = 1 \Rightarrow k = \log_2 n$$

$$\#\text{Levels} = k+1 = \log_2 n + 1$$

Computation Tree



$\rightarrow 1 = 2^0 \times 1$

$\rightarrow 2 = 2^1 \times 1$

$\rightarrow 4 = 2^2 \times 1$

$= 2^{k-1} \times 1$

$= 2^k \times 0$

$$2^0 + 2^1 + 2^2 + \cdots + 2^{k-1}$$

$$= 2^{\log_2 n - 1 + 1} - 1 = 2^{\log_2 n} - 1$$

$$= n - 1$$