DATA STRUCTURE And Algorithms

BATCH - I

K. SARAVANA KARUPPU.

ASSIGNEMENT (ONE):

for (i=1; i<=n; i++).

for (j=1; j<=n; j++).

To find T(n)

as Consider (A) a Constant factor.

for (i=1; i≤n; i++) → (n) times

for (j=1; j<=n; j++) → (n) times:

if ("Hi") (to print). (n^2)

$\therefore$ T(n) to print ("Hi").

$O(n^2)$ //.

$O(1)$.

2). for (i = 1; i < n; i * 3). $\rightarrow$ n

for (j = 1; j < n; j++). $\rightarrow$ n.

If ("Hello").

As $i/1$

i.e $1 \times 3 = 3$

$3 \times 3 = 6 \ (3^2)$

$= 3^2 \times 3 = 3^3$.

:

3^k (till k)

Assume $i > n$

$\therefore i = 3^k$

$= 2^k > n$

$2^k = n$

$k = \log_3 n$

$\therefore O(\log_3 n)$

As $(j=1;\ j<=n;\ j++)$

let:

$1 + + + 1 + \cdots + n = n$

$\therefore O(n)$

Assuming $n = 12$

$i/_1$

$\underbrace{\cancel{3}\ \cancel{6}\ \cancel{4}}_{3} \Rightarrow n = 3$

$\log_2 12 = 3$

$= \log_3 12^3$

$= 3 \log_3 12$

$= 2.26$ // $\rightarrow$ The answer is said to be Sealed value $[\log n]$

A it is neave the numer of loop that is

$n = 3$

$\therefore T(n) = O(\log_3 n)$