DESIGN AND ANALYSIS OF ALGORITHM

TUTORIAL-1

① Asymptotic notation is used to ~~design~~ describe the running time of an algorithm, how much time an algorithm take with a given input

Types

(i) Big-oh (O) :- $f(n) = O(g(n))$.

$g(n)$ is 'tight' upper bound of $f(n)$

(ii) Big omega (Ω)

$f(n) = \Omega g(n)$

$g(n)$ is 'tight' lower bound of $f(n)$

(iii) Theta (Θ)

Theta gives the tight upper and lower bound both.

(iv) Small-oh (-o)

o give us upper bound

(v) Small-omega (ω)

gives lower bound

(2) $i = 1, 2, 4, 8, 16 - - - n$

It forms G.P

$a = 1$

$r = \frac{2}{1} = 2$

$t_k = ar^{k-1}$

$n = 1 \cdot 2^{k-1}$

$n = \frac{2^k}{2}$

$2^k = 2n$

$k \log_2 2 = \log_2 2 + \log_2 n$

$k = \log_2 n + 1$ [As $\log_2 2 = 1$]

$TC \rightarrow O(\log_2 n + 1)$

$\rightarrow O(\log n)$

(3) $T(n) = \{3T(n-1)$ if $n > 0$ otherwise 1

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

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

$= 3^2 T(n-2)$

$= 3^3 T(n-3)$

$\Rightarrow 3^n T(n-n)$

$= 3^n T(0) = 3^n$

$TC \rightarrow O(3^n)$

(4) $T(n) = 2T(n-1) - 1$ if $n > 0$ otherwise 1

$T(0) = 1$

$T(1) = 2T(0) - 1 = 1$

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

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

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

So, Time complexity $= 2T(n) - 1 = 1$

$\Rightarrow O(1)$

(5)

i	s
1	1
2	3
3	6
4	10
⋮	⋮
k	n

Total $= \frac{k(k+1)}{2}$

while

$\frac{k(k+1)}{2} >= n$

$k(k+1) = 2n$

$k^2 + k = 2n$

$k^2 = n$

$TC \rightarrow O(\sqrt{n})$

(6)

Count	i
0	
1	1
2	2
3	3
⋮	⋮
k	$\sqrt{n}$

loop execute till value of count is k so

$k > = \sqrt{n}$

$\boxed{TC = O(\sqrt{n})}$

Recursive function

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

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

$T(n)^2 = T(n-6) + 2n^2 + 2$

⋮

$T(n) = T(n - k * 3) + kn^2 + k$

$T(n) = T(0) + n^3 + n$

$\boxed{TC \to O(n^3)}$

Let
$n - 3k = 1$
$n = 3k + 1$
$n = k$

(7)

i	j	k	times
$n/2$	$1 \to n$	$1 \to n$	$\log n \cdot \log n$
$n/2+1$	$1 \to n$	$1 \to n$	$\log^2 n$
⋮			
n			

Total $= \frac{n}{2} \log^2 n$

$\boxed{TC \to O(n \log^2 n)}$

(9)

i	j	times
1	$1 \to n$	n
2	$1 \to n$	$n/2$
⋮	⋮	
n	$1 \to n$	n

$\to n\left(n + \frac{n}{2} + \frac{n}{3} - - \frac{n}{n}\right)$

$\to n^2\left(1 + \frac{1}{2} + - - \frac{1}{n}\right)$

$\boxed{TC \to O(n^2)}$

(8)

i	j	time
1	$1 \to n$	n
2	$1 \to n$	n
3	$1 \to n$	n
⋮	⋮	⋮
k	$1 \to n$	n

total $= n^2$

(10) $f(n) = n^k$
$g(n) = c^n$
$f(n) = O(g(n))$
$n^k = O(c^n)$
$n^k \leq c^n$
Let $k = 1$ $\quad c = 2$
$n \leq 2^n$ with $n \in 2$
so relation is
$n^k = O(c^n)$