DESIGN AND ANALYSIS OF ALGORITHMS

TUTORIAL-1

Name:→ Srijan

Section:→ F

ROLL NO.:→ 30

Univ. ROLL NO.:→ 2017067.

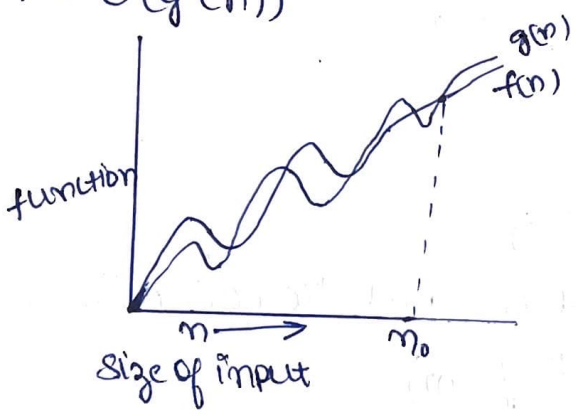
Ques:→

Solution:→ Asymptotic Notations:→ They are the mathematical notations used to describe the running time of an algorithm when the input tends towards a particular value or a limiting value.

Different asymptotic notation —

i) Big O(n)

$$f(n) = O(g(n))$$



$f(n) = O(g(n))$

iff $f(n) \leq c\,g(n)$

$\forall\ n \geq n_0$

for some constant, $c > 0$

g(n) is "tight" upper bound of f(n).

Eg. $f(n) = n^2 + n$

$g(n) = n^3$

$n^2 + n \leq cn^3$

$n^2 + n = O(n^3)$

ii) Big Omega (Ω)

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

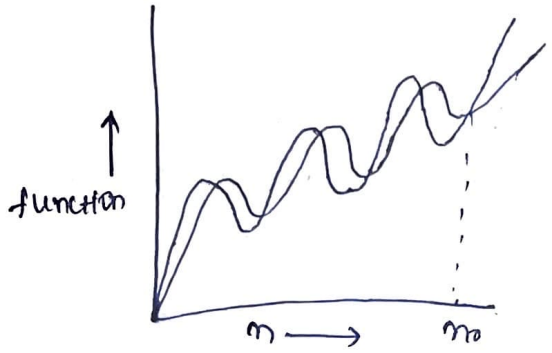
g(n) is "tight" lower bound of function f(n).

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

$$f(n) \geq c\, g(n)$$

$$\forall\ n \geq n_0$$

for some constant $c > 0$



Ex:- $f(n) = n^3 + n^2$

$g(n) = n^2$

$n^3 + n^2 = \Omega(n^2)$

iii) Big Theta (Θ)

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

g(n) is both "tight" upper and "lower" bound of function f(n)

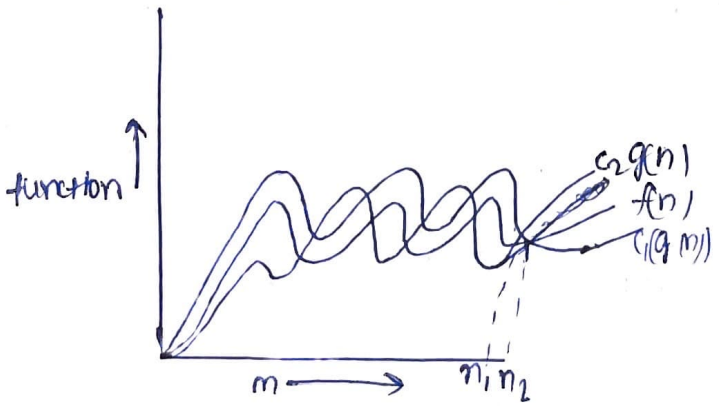
$$f(n) = \Theta(g(n))$$

iff

$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

$$\forall\ n \geq \max(n_1, n_2)$$

for some constant $c_1 > 0$ & $c_2 > 0$



Eg:- $3n+2 = O(n)$ as $3n+2 \geq 3n$ and

$3n+2 \leq 4(n)$ for n, $k_1 = 3$, $k_2 = 4$ & $n_0 = 2$

(iv) small $\theta(\theta)$:-

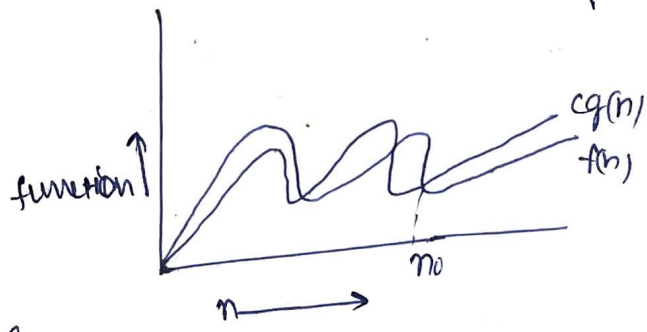
for $f(n) = \theta(g(n))$

$g(n)$ is upperbound of function $f(n)$

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

when $f(n) < cg(n)$

$\forall\ n > n_0$ & $\forall$ constants, $c > 0$



Eg:- $f(n) = n^2$

$g(n) = n^3$

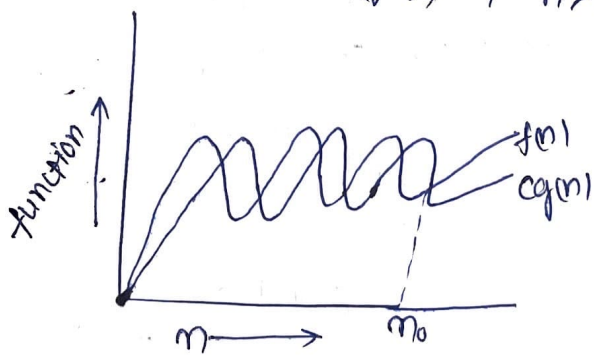
$n^2 = O(n^3)$

v) small omega $\omega(n)$

$f(n) \geq \omega(g(n))$

$g(n)$ is lower bound of $f(n)$.

$f(n) = \omega(g(n))$ $\forall\ n > n_0$ & $\forall$ constants, $c > 0$



$f(n) = 4n+6$; $g(n) = (1)$

<u>2011</u>

<u>Solution:</u> → for (i=1 to n)

{ i = i * 2 }

→ $i = \underbrace{1, 2, 4, 8, 16, \ldots n}_{k}$ (G.P)

$= O(k)$

$a = 1$, $r = \frac{2}{1} = 2$.

GP k^{th} value $= t_k = ar^{k-1}$

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

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

$2n = 2^k$

$\log(2n) = k \log 2$

$k = \log_2 2n$

$k = \log_2 2 + \log_2 n$

$k = 1 + \log_2 n$

Time complexity $= O(1 + \log_2 n)$

$= O(\log_2 n)$

3Qus:-

solution:→

$T(n) = 3T(n-1)$ —①

let $n = n-1$

$T(n-1) = 3T(n-2)$ —②

put ② in ①

$T(n) = 3 \times 3T(n-2)$ —③

put $n = n-2$

$T(n-2) = 3T(n-3)$ —④

put ④ in ③

$T(n) = 3 \times 3 \times 3T(n-3)$ —⑤

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

$= 3^n T(0)$

$= 3^n$

$= O(3^n)$

4Qus:-

solution:→

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

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

$= 2^2(T(n-2)) - 2 - 1 \cdots\cdots$

$\Rightarrow 2^n T(n-n) - 2^{n-1} - 2^{n-2} - 2^{n-3} \cdots\cdots 2^2 - 2^1 - 2^0$

$\Rightarrow 2^n - 2^{n-1} - 2^{n-2} - 2^{n-3} \cdots\cdots 2^2 - 2^1 - 2^0$

$\Rightarrow 2^n(2^n - 1)$

$T(n) = 1$

$= 2^n T(n-n) - 2^{n-1} - 2^{n-2} - 2^{n-3} - \cdots - 2^3 - 2^1 - 2^0$

$= 2^n - 2^{n-1} - 2^{n-2} - 2^{n-3} - \cdots - 2^2 - 2^1 - 2^0$

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

$T(n) = 1$

5 Qus

solution :→

```
Int i=1 , S=1;
while (S<=n) {
      i++;
      S= S+i;
      Printf ("#");
}
```

$S_i = S_{i-1} + i$

i is incrementing by one step

S " n " value of i

following will be value after four interations-

⇒ i=2 , S=3 1st iteration

⇒ i=3 , S=6 2nd "

⇒ i=4 , S=10 3rd "

let value of n be K ; value of S = 1, 3, 6, 10 - - - -

S represents a series of sum of first n natural numbers

for i=K , S= $\frac{K(K+1)}{2}$

for stopping loop

$\frac{K(K+1)}{2} > n \Rightarrow \frac{K^2+K}{2} > n$

$T(n) = O(\sqrt{n})$

6 Qus

solution :→

```
void function (int n) {
    int i, count=0;
    for (i=1; i*i<=n; i++)
        count++;
}
```

i = 1, 2, 3 - - - n

i^2 = 1, 4, 9 . - - - n^2

so $i^2 < n$ or $i <= \sqrt{n}$

$a_k = a + (k-1)d$

$a_1 = 1 \quad d = 1$

$a_k <= \sqrt{n}$

$\sqrt{n} = 1 + (k-1) \cdot 1$

$\sqrt{n} = k$

$T(n) = O(\sqrt{n})$

Ques :-

Solution :→

```
void function (int n) {
    int i, j, k, count = 0;
    for (i = n/2 ; i <= n ; i++)
        for (j = 1 ; j <= n ; j = j * 2)
            for (k = 1 ; k <= n ; k = k * 2)
                count ++;
}
```

$i = n/2$ $\quad j = \log_2 n \quad$ $k = \log_2 n$

$\vdots$

$(n/2 + 1)$ times $\quad \log_2 n \quad \log_2 n$

$O(i * j * k) = O((n/2 + 1) * \log_2 n + \log_2 n)$

$= O((n/2 + 1) * (\log n)^2)$

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

Ques :-

Solution :→

```
function (int n) {
    if (n == 1) return ;
    for (i = 1 to n) {
        for (j = 1 to n) {
            print ("*");
        }
    }
}
```

$T(n) = T(n-3) - n^2 \quad -①$

$T(1) = 1 \quad -②$

put $n = n-3$ in ①

$T(n-3) = T(n-6) + (n-3)^2 \quad -③$

put 3 in ①

$T(n) = T(n-6) + (n-3)^2 + n^2 \quad -④$

put $n = n-6$ in ①

$T(n-6) = T(n-9) + (n-6)^2 \quad -⑤$

put ⑤ in ④.

$T(n) = T(n-4) + (n-6)^2 + (n-3)^2 + n^2$

generalising

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

let $n-3k = 1$

$\frac{n-1}{3} = k$

$T(n) = T(1) + \left(n-3\left(\frac{n-1}{3}-1\right)\right)^2 + \left(n-3\left(n-\frac{1}{3}\right)\right)^2 + \dots + n^2$

$T(n) = 1 + (3+1)^2 + (6+1)^2 + \dots + n^2$

$T(n) = 1^2 + 4^2 + 7^2 \dots n^2$

$T(n) = n^2 + \dots + 1^2$

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

Ques:→

solution:→

```
void function (int n){
for (i=1 to n){
    for (j=1; j<=n; j=j+i){
        printf("* ");
    }
}
}
```

for $i=1$; $j \rightarrow n$ times

for $i=2$; $j = 1+3+5+\dots+n$

$a_n = a + (k-1)d$

$a=1 \quad d=2$

$n = 2 + (k-1)\times 2$

$\frac{n-1}{2} = k-1$

$k = \frac{n-1}{2} + 1$

$\boxed{k = \frac{n+1}{2}}$ No. of terms

for $i=2$, $j \rightarrow \frac{n+1}{2}$ times

for $i=3$, $j = 1+4+7+\dots+n$

$n = 1 + (k-1)\times 3$

$\frac{n-1}{3} + k$

for $i=3$, $j = \frac{n+1}{3}$ times

generalising

for $i=n$, $j = \frac{n+k-1}{k}$ times

Time complexity is

$$\underbrace{n + \frac{n+1}{2} + \frac{n+2}{3} + \cdots + \frac{n+k-1}{k}}_{n \text{ times}}$$

general term $= \frac{n+k-1}{k}$

$$\sum_{k=1}^{n} \frac{n+k-1}{k} = \frac{\sum_{1}^{n} n + \sum_{1}^{n} k - \sum 1}{k}$$

$$\Rightarrow \frac{\frac{n(n+1)}{2} + nk - n}{k}$$

$$T(n) \Rightarrow \frac{n^2 + \frac{n}{2} + nk - n}{k}$$

Neglecting constant terms.

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

100 Qus:-

solution :→ As given n^k & c^n

relation between n^k & c^n is

$$n^k = O(c^n)$$

as $n^k \leq cc^n$

$\forall\ n \geq n_0$ & some constant $a > 0$

for $n_0 = 1$

$c = 2$

$1^k \leq a_2$

$n_0 = 1$ & $c = 2$