Date / /

Tutorial - 1

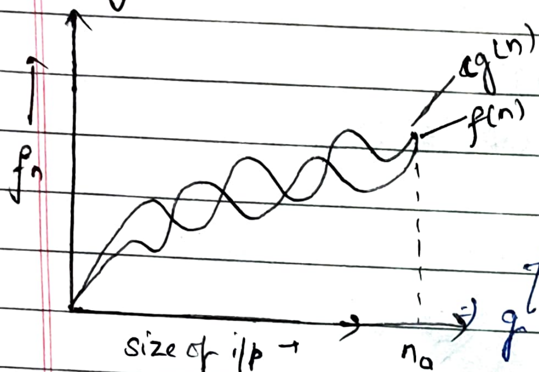
Name → Harshit Mehra
Section → CST
Semester → IV
C.Rno. → 31
Uni. Rno. → 2017497
Date → 10 March 2022

Date / /

Q1. Asymptotic Notations

They help you find the complexity (Tending to infinity) an algorith when i/p is very large.

(i) Big O (o)



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

if $f(n) \le C \cdot g(n)$ $\forall n \ge n_0$

for some constant $C > 0$

$\Rightarrow g(n)$ is tight upper bound of $f(n)$

(ii) Big Omega (Ω)

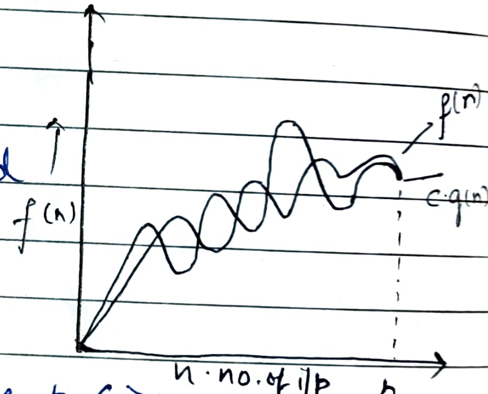
$f(n) = \Omega(g(n))$

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

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

iff $f(n) \ge c \cdot g(n)$

$\forall n \ge n_0$ for some constant $C > 0$



(iii) Theta (Θ)

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

g(n) is both tight upper & lower bound of f(n)

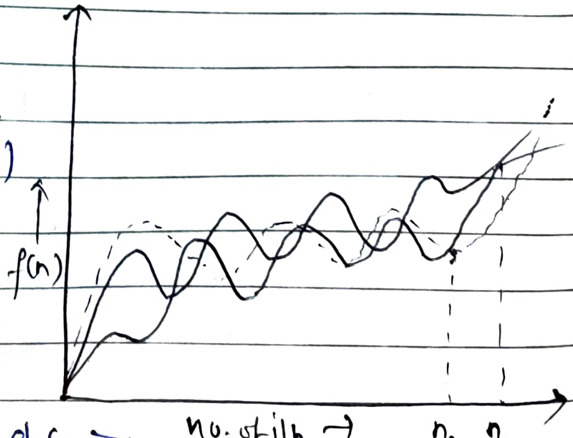
$f(n) = \Theta(g(n))$

iff

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

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

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



Date / /

(iv) Small o (O)

$f(n) = o(g(n))$

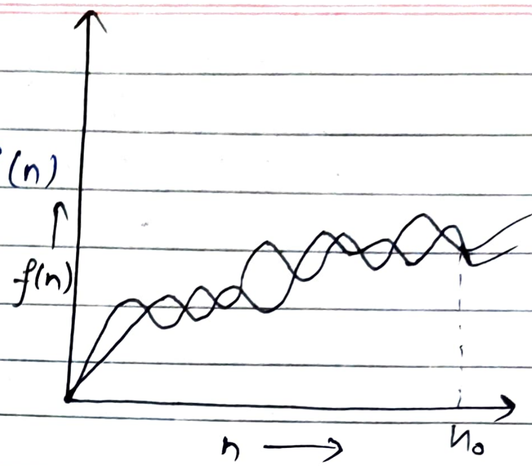
$g(n)$ is upper bound of $f(n)$

$f(n) = o(g(n))$

when $f(n) < c \cdot g(n)$

$\forall n > n_0$

& $\forall c > 0$



(v) Small omega (ω)

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

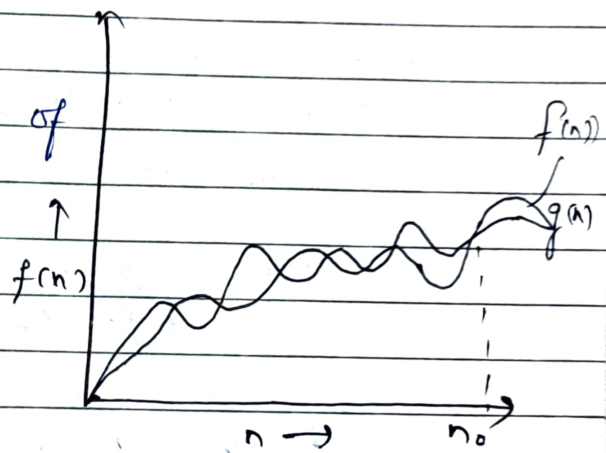
$g(n)$ is lower bound of $f(n)$

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

when $f(n) > c \cdot g(n)$

$\forall n > n_0$

& $\forall c > 0$



Q2. What should be the time complexity of

for ($i=1$ to n) $\{ i = i * 2 \}$

for ($i=1$ to n) // $i = 1, 2, 4, 8 \ldots\ldots n$

$\{ i = i * 2 \}$ // $O(1)$

$\Rightarrow \sum_{i=1}^{n} 1 + 2 + 4 + 8 + \ldots\ldots n$

GP k^{th} value $\Rightarrow T_k = ar^{k-1}$

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

$\Rightarrow n = 2^k$

$\Rightarrow 2n = 2^k$

$\Rightarrow \log 2n = k \log 2$

Date / /

$\Rightarrow \log 2 + \log n = k \log 2$

$\Rightarrow \log (n+1) = k$

$\Rightarrow O(k) = O(1 + \log n)$

$= O(\log n)$ Ans

Q3 $T(n) = \{3T.(n-1)$ if $n > 0$, otherwise $1\}$

$T(n) = 3T(n-1)$ —— ①

put $n = n-1$

$T(n-1) = 3T(n-2)$ —— ②

from ,d2

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

$= 9T(n-2)$ —— ③

putting $n = n-2$ in ①

$T(n) = 3(T(n-3))$ —— ④

$\Rightarrow T(n) = 27(T(n-3))$

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

putting $n-k=0$

$\Rightarrow n = k$

$\Rightarrow T(n) = 3^n [T(n-n)]$

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

$= 3^n \times 1$ $\{T(0) = 1\}$

$\Rightarrow T(n) = O(3^n)$ Ans

Date / /

Q4 $T(n) = \{2T(n-1) - 1$ if $n>0$, otherwise

$T(n) = 2T(n-1) - 1$ —— (1)

Let $n = n-1$

$\Rightarrow T(n-1) = 2T(n-2) - 1$ —— (2)

from (1) & (2)

$\Rightarrow T(n) = 2[2T(n-2) - 1] - 1$

$\Rightarrow T(n) = 4T(n-2) - 2 - 1$ —— (3)

let $n = n-2$

$\Rightarrow T(n-2) = 2T(n-3)$ —— (4)

from (3) and (4)

$\Rightarrow T(n) = 4[2T(n-3) - 1] - 2 - 1$

$\Rightarrow T(n) = 8T(n-3) - 4 - 2 - 1$

$\Rightarrow T(n) = 2^k T(n-k) - 2^{k-1} - 2^{k-2}$

$\Rightarrow GP = 2^{k-1} + 2^{k-2} + 2^{k-3} + \ldots$

$a = 2^{k-1}$

$r = 1/2$

$= \dfrac{a(1-r^n)}{1-r}$

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

$= 2^k(1-(1/2)^k)$

$= 2^k - 1$

Let $n-k = 0$

$\Rightarrow n = k$

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

$\Rightarrow T(n) = 2^n \cdot 1 - (2^n - 1)$

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

$\Rightarrow T(n) = O(1)$

Date / /

Q5 what should be time complexity of

```
int i=1, s=1;
while (i<=n)
{ i++; s=s+i;
  printf("#");
}
```

$i = 1, 2, 3, 4, 5, 6 \ldots\ldots$

$s = 1 + 3 + 5 + 10 + 15 \ldots\ldots n$

sum of $s = 1+3+6+10+\ldots\ldots n$ —— (1)

$s = 1+3+6+10+\ldots\ldots T_{n-1} + T_n$ —— (2)

from (1) and (2)

$0 = 1+2+3+4+\ldots\ldots n - T_n$

$\Rightarrow T_n = 1+2+3+4+\ldots\ldots k$

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

for k iterations

$1+2+3+\ldots\ldots+k <= n$

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

$\Rightarrow \frac{k^2+k}{2}$

$\Rightarrow O(k^2) <= n$

$\Rightarrow k = O(\sqrt{n})$

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

Rashe.

Date / /

Q6 Time complexity of

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

// O(1)

as $i^2 <= n$

$\Rightarrow i <= \sqrt{n}$

$i = 1, 2, 3, 4 \ldots\ldots \sqrt{n}$

$\sum_{i=1} 1+2+3+4+ \cdots \sqrt{n}$

$\Rightarrow T(n) = \frac{\sqrt{n} \times (\sqrt{n}+1)}{2}$

$\Rightarrow T(n) = \frac{n\sqrt{n}}{2}$

$\Rightarrow T(n) = O(n)$ Ans

Q7 Time complexity of

~~void fo fn (int n)~~

```
void fn(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 ++;
}
```

for $k = k^{*2}$

$k = 1, 2, 4, 8 \ldots\ldots n$

→ GP ⇒ $a = 1$ $r = 2$

$$R_n = \frac{a(r^n - 1)}{r - 1}$$

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

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

→ $\log n = k$

→

i	j	k
1	$\log n$	$\log n * \log n$
2	$\log n$	$\log n * \log n$
⋮	⋮	⋮
n	$\log n$	$\log n * \log n$

⇒ $O(n * \log n * \log n)$

→ $O(n \log^2 n)$ Ans

Q8. Time complexity of

```
function (int n)
{ int (n==1)
   return;                    // O(1)
   for (i=1 to n)             // i = 1,2,3,4.... n) → O(n)
   { for (j=1 to n)           // j = 1,2,3,4...... n ⇒ O(n^2)
      { print ('*');
      }
   }
   function (n-3);            // T(n/3)
}
```

Date / /

$\Rightarrow T(n) = T(n/3) + n^2$

$\Rightarrow a = 1, \quad b = 3, \quad f(n) = n^2$

$c = \log_3 1 = 0$

$\Rightarrow n^0 = 1 > (f(n) = n^2)$

$\Rightarrow T(n) = \theta(n^2)$ Ans

Q9. Time complexity of

void function (int n)

```
{
   for (i=1 to n)                     // O(n)
   {  for (j=1; j<=n; j=j+i)
        print ("*")                   // O(1)
   }
}
```

for $i=1$

$\rightarrow j = 1, 2, 3, 4 \ldots\ldots n = n$

for $i=2$

$\rightarrow j = 1, 3, 5 \ldots\ldots n = n/2$

for $i=3$

$\rightarrow j = 1, 4, 7 \ldots\ldots n = n/3$

⋮

for $i=n$

$\rightarrow j = 1 \ldots\ldots 1$

$\rightarrow \sum_{j=n}^{1} n + \frac{n}{2} + \frac{n}{3} + \frac{n}{4} + \ldots\ldots + 1$

$\rightarrow \sum_{j=n}^{1} n\left[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots\ldots \frac{1}{n}\right]$

$\sum_{j=n} n\,[\log n]$

Date / /

$\Rightarrow T(n) = [n \log n]$

$T(n) = O(n \log n)$ Ans

Q10 For functions, n^k & c^n, what is the asymptotic relation b/w these functions. Assume that $k >= 1$, & $c > 1$ are constant find out the value of c & n_0 for which relation holds.

→ as given n^k & c^n

relation b/w n^k & c^n is

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

as $n^k \leq a c^n$

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

for $n_0 = 1$

$c = 2$

$\Rightarrow 1^k \leq a$

$\rightarrow n_0 = 1$ & $c = 2$ Ans