DATE
PAGE

Tutorial-1

Vishal Chauhan
CST-21

1. What do you understand by Asymptotic notation? Define diff. asymptotic notation with example.

(i) Big O(n)

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

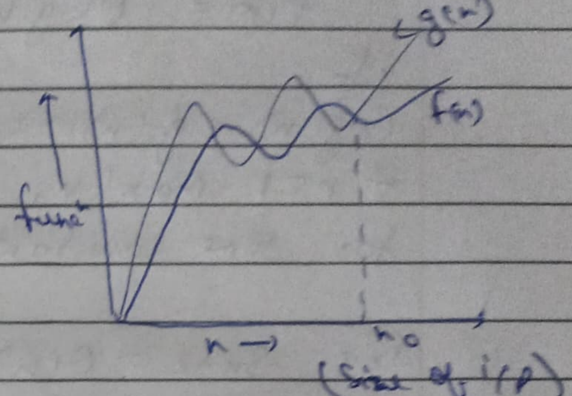
if

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

$\forall n \ge n_0$

for some constant, $c > 0$

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



Ex → $f(n) = n^2 + n$

$g(n) = n^3$

$n^2 + n \le c.n^3$

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

(ii) Big omega (Ω)

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

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

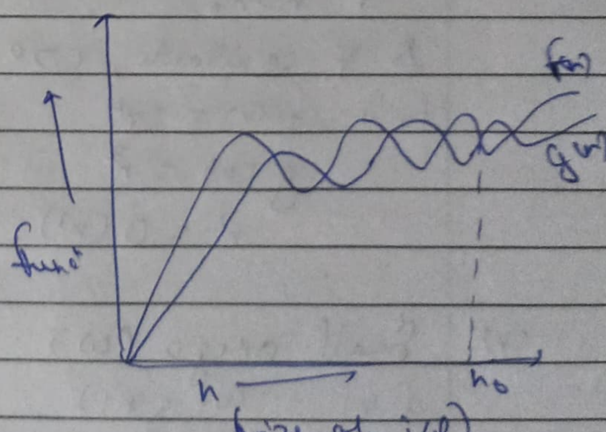
$f(n) = \Omega(g(n))$

if

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

$\forall n \ge n_0$

for some constant $c > 0$



Ex - $f(n) = n^3 + 4n^2$

$g(n) = n^2$

$n^3 + 4n^2 = \Omega(n^2)$

Vishal

DATE
PAGE

(iii) Big Theta (Θ)

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

g(n) is both "tight" upper bond & lower bond of function f(n)

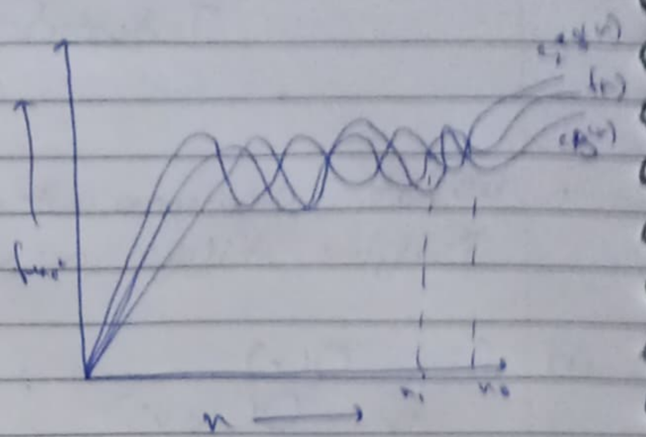
$f(n) = \Theta(g(n))$

if

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

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

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



Ex → $3n+2 = \Theta(n)$ as $3n+2 \ge 3n$ & $3n+2 \le 4n$ for n, $k_1 = 3$, $k_2 = 4$ & $n_0 = 2$

(iv) Small o (o)

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

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

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

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

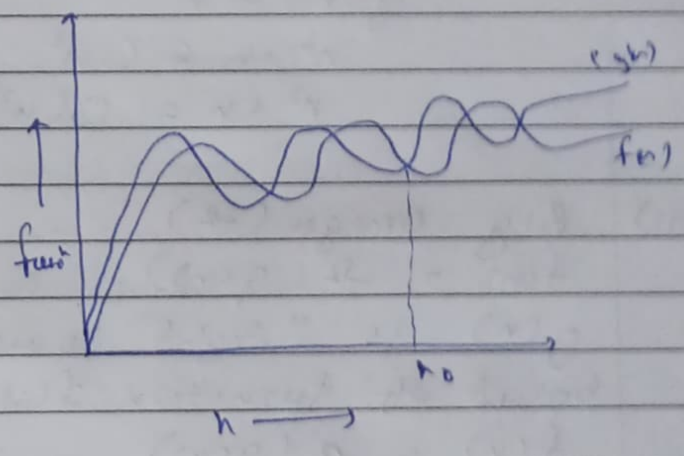
$\forall\ n > n_0$

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

Ex → $f(n) = n^2$

$g(n) = n^3$

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



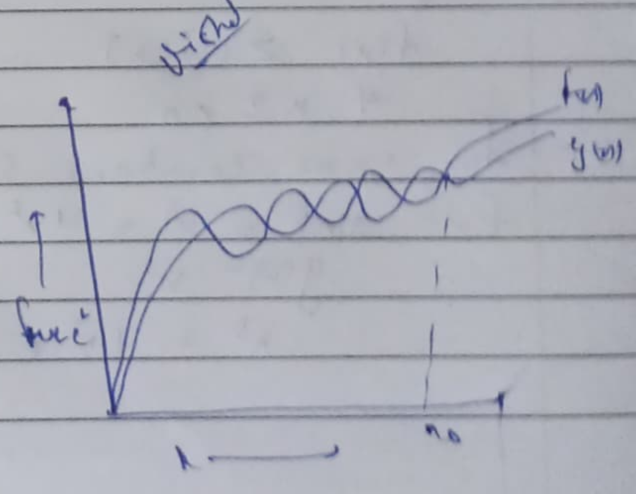
(v) Small omega (ω)

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

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

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

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



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

2 what should be time complexity of:
for(i=1 to n) { i = i * 2;}

Sol
for (i=1 to n)
{
$i = i * 2; \longrightarrow O(1)$
}

$i = \overbrace{1, 2, 4, ----, n}^{k}$

$a = 1$, $r = \frac{b_2}{b_1} = 2$

$t_k = a r^{k-1}$ (k^{th} value of GP)
$t_k = 2^{k-1}$
$t_k = \frac{2^k}{2}$ let $t_k = n$

$2n = 2^k$
$\log_2(2n) = k \log_2 2$
$\log_2 2 + \log_2 n = k$
$\log_2 n + 1 = k$
$O(\log_2 n + \cancel{1}) = O(\log_2 n)$

3. $T(n) = \{ 3T(n-1)$ if $n > 0$, otherwise 1}
$T(n) = 3T(n-1)$ —— ①
put $n = n-1$ in ①
$T(n-1) = 3T(n-2)$ —— ②
put ② in ①
$T(n) = 9T(n-2)$ — ③
put $n = n-2$ in ①
$T(n-2) = 3T(n-3)$ —— ④
put ④ in ③
$T(n) = 27T(n-3)$ —— ⑤

Sign

DATE
PAGE

generalising

$T(n) = 3^k T(n-k)$ — (5)

Let, $n-k=1$

$k = n-1$ — (6)

put (6) in (5)

$T(n) = 3^{n-1} T(1)$

$T(n) = 3^{n-1} \cdot 1$

$T(n) = \frac{3^n}{3}$

$= O(3^n)$

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

$T(n) = 2T(n-1) - 1$ — (1)

put $n = n-1$

$T(n-1) = 2T(n-2) - 1$ — (2)

put (2) in (1)

$T(n) = 4T(n-2) - 2 - 1$ — (3)

put $n = n-2$ in (1)

$T(n-2) = 2T(n-3) - 1$ — (4)

put (4) in (5)

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

generalising,

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

Let, $n-k=1$

$k = n-1$

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

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

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

Nirav

DATE

PAGE

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

$$= 2^{k-1}\left(1 - 1 + (1/2)^{k-1}\right)$$

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

(5) What should be time complexity for

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

$i = 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6$

$s = 1 + 3 + 6 + 10 + 15 + \ldots$

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

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

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

$T_k = 1 + 2 + 3 + 4 + \ldots + k$

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

for k iterations

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

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

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

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

$k = O(\sqrt{n}) \rightarrow \boxed{T(n) = O(\sqrt{n})}$

6. Time complexity of

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

as $i^2 <= n$

$i <= \sqrt{n}$

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

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

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

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

$$T(n) = O(n)$$

7. Time complexity of

```
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 $k = x^2$

$k = 1, 2, 4, 8 --- n$

G.P $\Rightarrow$ $a = 1$, $r = 2$

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

DATE
PAGE

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

$n = 2^{k} - 1$

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

1	1	k
1	$\log(n)$	$\log n * \log n$
2	$\log(n)$	$\log n * \log n$
⋮	⋮	⋮
n	$\log(n)$	$\log n * \log n$

$\Rightarrow O(n \times \log n \times \log n)$

$O(n \log^2 n)$

8. Time complexity of

```
function(int n)
{
    if (n==1) return;
    for (int 1 to n)
        for( j=1 to n)
            printf(" * ");
        }
    }
    function (n-3);
}
```

for:- for(i=1 to n) ~~fixed~~

we get j=n times every turn

∴ $i * j = n^2$

Now, $T(n) = n^2 + T(n-3);$

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

$T(n-6) = (n-6)^2 + T(n-5);$

$T(1) = 1;$

} k time

DATE
PAGE

Now subs. each value in $T(n)$

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

Let

$n - 3k = 1$

$k = (n-1)/3$

Total terms $= k+1$

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

$T(n) \simeq k n^2$

$\therefore \quad T(n) = O(n^3)$

(9) Time complexity of -

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

for : $i=1$ $\quad j = 1+2+ \cdots (n \geq j+i)$

$i=2$ $\quad j = 1+3+5+ \cdots$ "

$i=3$ $\quad j = 1+4+7+ \cdots$ "

n^{th} terms of AP is

$T(M) = a + d \times m$

$T(M) = 1 + d \times m$ $\quad$ ~~in~~ std

$(n-1)/d = m$

for $\quad i=1$ $\quad (n-1)/1$ times

$i=2$ $\quad (n-1)/2$ times

$\vdots$

$i = n-1$

DATE
PAGE

we get,

$T(n) = i_1 j_1 + i_2 j_2 + \cdots + i_{n-1} j_{n-1}$

$= \frac{(n-1)}{1} + \frac{(n-2)}{2} + \frac{(n-3)}{3} + \cdots 1$

$= n + \frac{n}{2} + \frac{n}{3} + \cdots \frac{n}{n-1} - n \times 1$

$= n\left[1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{n-1}\right] - n + 1$

$\approx n \times \log n - n + 1$

since $\int \frac{1}{n} = \log n$

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

(16) for the func^n n^k & c^n, what is the asymptotic relationship b/w these func^n? Assume that $k >= 1$ & $c > 1$ are constants find out the value of c & n_0 for which relationship holds.

As given n^k & c^n

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

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

$n^k \le a(c^n)$

$\forall\ n \ge n_0$ &

constant, $a > 0$

for $n_0 = 1$

$c = 2$

$\Rightarrow 1^k \le a 2^1$

$\Rightarrow n_0 = 1$ & $c = 2$

vishal