Sunil
PAGE
DATE

DAA ASSIGNMENT

Name :- Ritika Sharma
Class :- CST - SPL - 1
Class Roll no :- 13
Uni Roll no :- 2017534

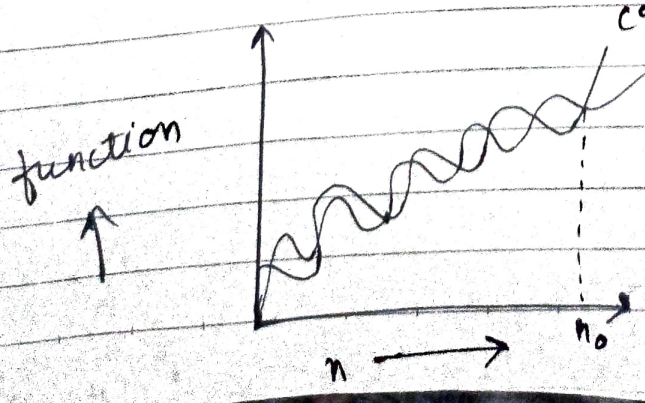
Ans 1. Asymptotic Notations

⇒ These notations are used to analyze and determine the running time of an algorithm
⇒ They tells the complexity of an algorithm when the input is very large
⇒ when input tends towards a particular value, we can use these notations to determine run-time of an algorithm.

TYPES OF ASYMPTOTIC NOTATIONS

(i) **Big - oh** :- used for upper bound values
:- gives the worst case complexity of an algorithm.

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



⇒ g(n) is the 'tight' upper bound

$f(n) = O(g(n))$
iff $f(n) \leq c \cdot g(n)$
$\forall\ n \geq n_0$ & some constant $c > 0$

Example

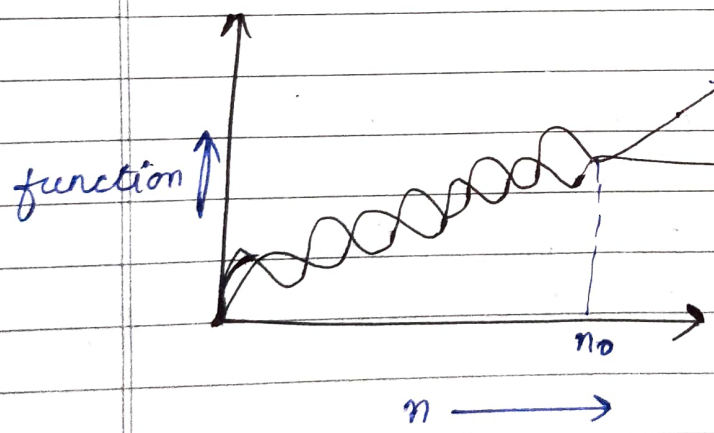
```
for(i=1; i<=n; i++)  ------> O(n)
{
   print (i);
}
```

$T(n) = O(n)$

(ii) **Big Omega (Ω) :-** represents the lower bound of an algorithm

:- provides best case complexity of an algorithm.

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



$g(n)$ is the 'tight' lower bound

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

iff

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

$\forall\ n \geq n_0$ and some constant $c > 0$.

Sunil

PAGE

DATE

Eg :- $f(n) = 2n^2 + 3n + 5$ $\quad g(n) = n^2$

$0 \leq c \cdot g(n) \leq f(n)$

$0 \leq c \cdot n^2 \leq 2n^2 + 3n + 5$

$c \leq 2 + \frac{3}{n} + \frac{5}{n^2}$

putting $n = \infty \Rightarrow \frac{3}{n} \to \infty$; $\frac{5}{n^2} \to \infty$

$\Rightarrow c = 2$

$2n^2 \leq 2n^2 + 3n + 5$

put $n = 1$

$2 \leq 2 + 3 + 5$

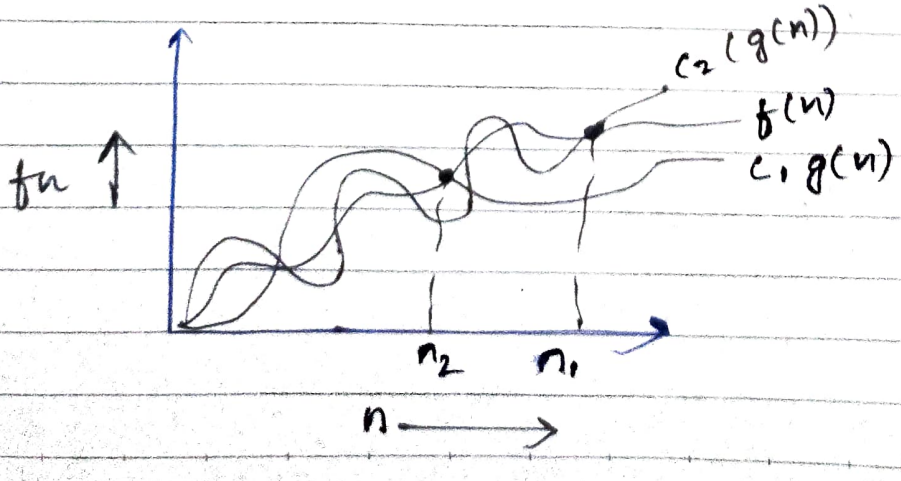
$2 \leq 10$ True.

$\boxed{c = 2}$, $\boxed{n = 1}$, $\boxed{n_0 = 1}$

$0 \leq 2n^2 \leq 2n^2 + 3n + 5$

iii Big Theta (Θ) :-

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



Sunil
PAGE
DATE

eg:- $f(n) = 10 \log_2 n + 4$ $\quad g(n) = \log_2 n$

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

$\Rightarrow 10 \log_2 n + 4 \leq 10 \log_2 n + \log_2 n$

$10 \log_2 n + 4 \leq 11 \log \log_2 n$

$c_2 = 11$

$4 \leq 11 \log_2 n - 10 \log_2 n$

$4 \leq \log_2 n$

$16 \leq n$

here,

$\forall \quad n \geq 16$

$n_2 = 16$

$c_2 = 11$

$f(n) \geq c_1 \cdot g(n)$

$10 \log_2 n + 4 \geq 2 \log_2 n$

$c_1 = 1 \quad , \quad n > 0$

$\Rightarrow n_1 = 1 \Rightarrow n_0 = \max(n_1, n_2)$

$n_0 = 16$

$\log_2 n \leq 10 \log_2 n + 4 \leq 11 \log_2 n$

$c_1 = 1$

$c_2 = 11$

$\Rightarrow \theta(\log_2 n)$

(iv) small oh (o) :-

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

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

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

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

$\forall \; n > n_0$

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

(v) small omega (ω) :

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

$\forall \quad n > n_0$

and $\forall \quad c > 0$

Sunil

PAGE

DATE

Ques 2.

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

i = $\underbrace{1, 2, 4, 8, 16 \ldots\ldots n}_{k\text{-terms}}$

$a = 1$, $r = 2$

$\Rightarrow$ k^{th} term $\Rightarrow$ $t_k = ar^{k-1}$

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

taking $\log_2$ both sides

$\log_2 n = \log_2 2^{k-1}$

$\log_2 n = k - 1$

$k = 1 + \log_2 n$

$T(n) = O(k)$

$= O(1 + \log_2 n)$

$= O(\log_2 n)$

Sunil

PAGE

DATE

Ques3. $T(n) = \{3T(n-1)$ if $n>0$, otherwise $1\}$

$$T(n) = 3T(n-1) \quad \text{---------} \quad (1)$$

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

$$T(n-1) = 3T(n-2) \quad \text{---------} \quad (2)$$

put (2) in (1)

$$T(n) = 3[3T(n-2)] \quad \text{---------} \quad (3)$$

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

$$T(n-2) = 3T(n-3) \quad \text{---------} \quad (4)$$

put (4) in (3)

$$T(n) = 3[3[3T(n-3)]] \quad \text{---------} \quad (5)$$

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

generalized form :- $\boxed{T(n) = 3^k T(n-k)}$

put $n-k=0$

$n=k$

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

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

$$\Rightarrow \boxed{O(3^n)}$$

Sunil

PAGE

DATE

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

$T(n) = 2T(n-1) - 1$ ——— (1)

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

$T(n-1) = 2T(n-2) - 1$ ——— (2)

put (2) in (1)

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

$T(n) = 4T(n-2) - 2 - 1$ ——— (3)

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

$T(n-2) = 2T(n-3) - 1$ ——— (4)

put (4) in (3)

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

$T(n) = 8T(n-3) - 4 - 2 - 1$ ——— (5)

generalized form

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

put $n - k = 0$

$n = k$

$T(0) = 1$ [given]

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

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

K terms

$a = 2^{n-1}$

$r = 1/2$

Sunil

PAGE

DATE

Sum of G.P. $= \frac{2^{n-1}\left(1-(1/2)^{n-1}\right)}{1-\frac{1}{2}}$

$= 2^n - 2$

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

$O(2) \Rightarrow O(1)$

Sunil

PAGE

DATE

Ques 5.

```
int  i = 1 , s = 1 ;
while (s ≤ n)
{
   i++;
   s = s + i ;
   printf ("#");          ----------> O(1)
}
```

S = 1, 3, 6, 10, 15 ---- n

(K terms)

K^{th} term

$t_K = t_{K-1} + K$

$K = t_K - t_{K-1}$ ——— (1)

$K = n - t_K - 1$

(loop runs K times)

Time complexity ⇒ $O(1+1+1+n \text{ ---- } t_{k-1})$

but $t_{k-1} = c$ (constant)

Time complexity ⇒ $O(3 + n - c)$

⇒ $O(n)$

PAGE
DATE

Ques 6. $i*i = \underbrace{1^2, 2^2, 3^2, 4^2, 5^2, \ldots n}_{k \text{ terms}}$

k^{th} term $\Rightarrow$ $t_k = k^2$

$k^2 = n$

$k = n^{1/2}$

time complexity $= O(1+1+1+n^{1/2}+1)$

$= O(n^{1/2}) = O(\sqrt{n})$

Ques 7. Time complexity of :-

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

$i \Rightarrow \frac{n}{2}, \frac{n+2}{2}, \frac{n+4}{2}, \frac{n+6}{2}$ ... upto n

$\Rightarrow \frac{n+0\times2}{2}, \frac{n+(1*2)}{2}, \frac{n+(2*2)}{2}, \ldots n.$

general form :-

$$\boxed{t_k = \frac{n+k*2}{2}}$$

DATE

total terms = $k+1$

$t_{k+1} = n$

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

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

$$n + 2k + 2 = 2n$$

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

i	j	k
$\frac{n}{2}$	$\log n$ times	$(\log_2 n)^2$
$\frac{n+2}{2}$	$\log_2 n$ times	$(\log_2 n)^2$
$\frac{n+4}{6}$	$\log_2 n$ times	$(\log_2 n)^2$

$\frac{(n-1)}{2}$ times $\Rightarrow$ $\left(\frac{n}{2} - 1\right)(\log_2 n)^2$

$$O\left(\frac{n}{2} \log^2 n - \log^2 n\right)$$

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

PAGE
DATE

Ques 8: Time complexity of :-

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

for function call :-

$\underbrace{n, n-3, n-6, n-9 \ldots\ldots 1}_{k \text{ terms}}$

AP $\Rightarrow d = -3$

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

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

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

$k = \frac{n+2}{3}$

$\Rightarrow$ function gives a recursive call $\frac{n+2}{3}$ times

$\Rightarrow$ Time complexity :- $\left(\frac{n+2}{3}\right)(n)(n)$

$= n^3$

$O(n^3)$

PAGE
DATE

$i = n/2$, $n+2/2$, $n+4/2$, $n+6/2$ - - - upto n

$\frac{n+0\times1}{2}$, $\frac{n+1\times2}{2}$, $\frac{n+2\times2}{2}$, $\frac{n+3\times2}{2}$ -- upto n

generalized form

$$t_k = \frac{n + k*2}{2}$$

total terms $= k+1$

$\Rightarrow t_{k+1} = n$

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

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

$\Rightarrow 2k = n-2$

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

$\Rightarrow$

i	j	k
$n/2$	$\log n$	$(\log n)^2$
$n+2/2$	$\log n$	$(\log n)^2$
$n+4/6$	$\log n$	$(\log n)^2$
⋮	⋮	⋮
$\frac{n}{(n/2-1)}$ term	$\log n$ term	$(\log n)^2$

$$\Rightarrow \left(\frac{n}{2} - 1\right)(\log n)^2$$

$$= O\left(n/2 \log^2 n - \log n\right)$$

$$= O\left(n \log^2 n\right)$$

Sunil

PAGE

DATE

Ques 9. TC of void function (int n)

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

for (i=1 ⟶ j = 1, 2, 3, 4, . . . n) = n

for i=2 ⟶ j = 1, 3, 5, - - - - n = n/2

for i=3 ⟶ j = 1, 4, 7, - - - - n = n/3

⋮

for i=n ⇒ j = 1, - - - - - n = 1

$$\sum_{j=n}^{1} n + n/2 + n/3 + n/4 + \cdots + 1$$

$$\Rightarrow \sum_{j=n}^{1} n\left[1 + 1/2 + 1/3 + 1/4 + \cdots - 1/n\right]$$

$$\Rightarrow \sum_{j=n}^{1} n \log n$$

$$\Rightarrow \boxed{T(n) = O(n \log n)}$$

Sunil
PAGE
DATE

Ques 10.

As given n^k & c^n
relation b/w n^k & c^n is $\boxed{n^k = O(c^n)}$

as $n^k \leq ac^n \; \forall \; n \geq n_0$ for (a) constant (no)
for $n_c = 1$
$c = 2$

$$1^K \leq a_2 1$$

$\boxed{n_0 = 1 \;\&\; c = 2}$