Name:- Aman Sharma
SEC:- CS-D
Rollno.:- 10

Date
Page No.

Tutorial - 1

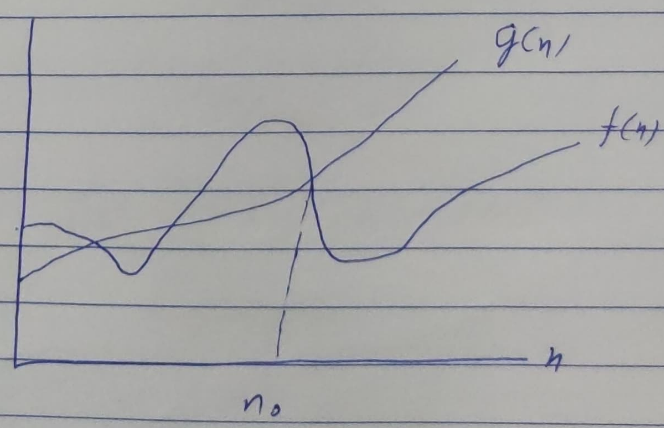
Q.1 What do you understand by Asymptotic notation. Define different Asymptotic notation with examples.

Sol:- They are ~~the~~ mathematical notation used to describe the running time of an algorithm when the input tends towards a particular value or a limiting value

There are mainly three asymptotic notations:

(i) Big-O-notation:

- provide worst complexity
- provide upper bound of an ~~running~~ running time algo



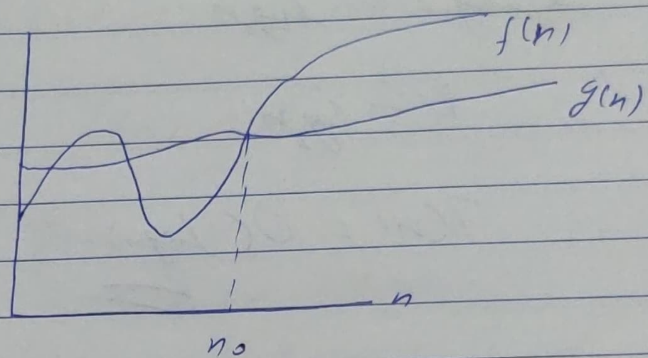
$f(n) = O(g(n))$

Date.
Page No.

$O(g(n)) = \{ f(n)$: there exist positive constant c & n_0 such that $0 \le f(n) \le g(n)$ for all $n \ge n_0$.

(ii) Omega Notation:

- provides best case time complexity
- ref. lower bound of running time algo

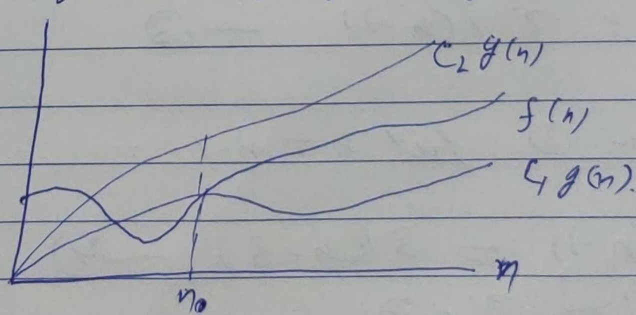


$$f(n) = \Omega(g(n))$$

$\Omega(g(n)) = \{ f(n)$: there exist positive constant c and n_0 such that $0 \le g(n) \le f(n)$ for all $n \ge n_0$.

(iii) Theta Notation (Θ-notation):

→ Used for analysing avg. time complexity



$\Theta(g(n)) = \{ f(n)$: there exist positive constant c_1, c_2, n_0 such that $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ for all $n \ge n_0$.

Date
Page No.

Q.2 What should be time complexity of :-

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

$2^k = n$

Taking log,

$k \log_2 2 = \log_2 n$

$k = \log_2 n$

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

Q.3 $T(n) = 3T(n-1)$ if $n>0$; otherwise 1. — ①

for $n = n-1$
$T(n-1) = 3T(n-2)$ — ②

from ① and ②,

$T(n) = 3^2 T(n-2)$ — ③

from ② put $n = n-1$

$T(n-2) = 3T(n-3)$ — ④
from ③ & ④
$T(n) = 3^3 T(n-3)$

Date.
Page No.

Generalising $T(n)$
$$T(n) = 3^k T(n-k) \quad \text{--- } ⑥$$

Put $n-k=1$

$k = n-1$

put $k = n-1$ in ⑥,

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

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

from ①,
$$T(n) = 3T(n-1)$$
put $n=1$
$$T(1) = 3T(0)$$
A/q
$$T(1) = 3 \cdot (1)$$
$$T(1) = 3$$

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

Time Complexity $\Rightarrow O(3^n)$

Date
Page No.

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

for $(n) = n-1$

$T(n-1) = 2T(n-2) - 1$ — ②

from ① & ②,

$T(n) = 2^2 T(n-2) - 1 - 2$ — ③

from ② put $n = n-1$

$T(n-2) = 2T(n-3) - 1$ — ④

from ③ & ④

$T(n) = 2^3 T(n-3) - 1 - 2 - 4$ — ⑤

Generating the term of eqn ⑤

$T(n) = 2^k T(n-k) - 1 - 2 - 4 = -2^k$ — ⑥

Put $n-k = 1$

$k = n-1$

put $k = n-1$ in eqn ⑥

$T(n) = 2^{n-1}\, T(1) - 1 \dfrac{(1-2^{n-1})}{(1-(1-1))}$

$= 2^{n-1} - 2^{n-1} + 1 = 1$

$\boxed{T(n) = O(1)}$

Q.5 Time Complexity of

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

i	s
1	1
2	3
3	6
4	10
5	15
:	:
k	$\frac{k(k+1)}{2}$

for k iteration

loop terminates when $\frac{k(k+1)}{2} > n$

$\therefore$ Time Complexity $k = O(\sqrt{n})$.

Q.6. Time Complexity of -

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

Date.
Page No.

i	count
1	0
2	1
3	2
:	:
k	k-1

Since $k * (k-1) < n$

$\therefore k \le \sqrt{n}$

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

Q.7 Time Complexity of :-

```
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	j	k
1	log n	log n * log n
2	log n	log n * log n
:	:	:
n	log n	log n * log n

Date.
Page No.

$\therefore \; O(n \times \log n + \log n)$

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

Q.8. Time Complexity of -

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

$\Rightarrow$ for ($i = 1$ to n)
we get $j = n$ times every then
$\therefore \; i \times j = n^2$

Now, $T(n) = n^2 + T(n-3);$
$T(n-3) = T(n-6) + (n-3)^2$] k times
$\vdots$
$T(1) = 1;$

Now substitute each value in $T(n)$

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

Date
Page No.

Let
$(n-3k) = 1$
$\therefore k = (n-1)/3$

$\therefore$ total time $= k+1$

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

$T(n) \simeq n^2 + n^2 + n^2 \ldots$ (k times $+1$)

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

$T(n) = \frac{(n-1)}{3} \times n^2$

$\therefore \boxed{T(n) = O(n^3)}$

Q.9. Time Complexity of -

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

for $i = 1$ $\quad j = 1 + 2 + \ldots (n \geq j+1)$
$i = 2$ $\quad j = 1 + 3 + 5 \ldots$ "
$i = 3$ $\quad j = 1 + 4 + 7 \ldots$ "

Date.
Page No.

m^{th} term of al is

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

$T(m) = 1 + d \times m$

$(n-1)/d = m$

		j	
for	$i=1$	$(n-1)/1$	times
	$i=2$	$(n-1)/2$	times
	$i=3$	$(n-1)/3$	times
	$i=n-1$	1	

We get,

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

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

$= n + n/2 + n/3 + \dots \quad \frac{n}{n-1} \quad - n \times 1$

$= n \times \log n \quad - n + 1$

Since $\int \frac{1}{n} = \log x$

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

Q.10 For the function, n^k & c^n, what is the asymptotic relationship between these functions.

Sol: We have given

n^k & c^n

by $k \geq 1$ & $c > 1$

for values $k \geq 1$, $c > 1$

We have $c^n \geq n^k$

$\therefore n^k = O(c^n)$
$\forall n \geq n_0$, & some constant $k_0 > 0$

$\Rightarrow k_0 c^n \geq n^k$
for $c > 1$ & $n = 1$

We get

$\Rightarrow k_0 c \geq 1$

$\therefore \boxed{c > 1 \ \& \ n_0 = 1}$

Ans