20-01-2017 | 020-345 | 03

20

2017
January
Friday

12	December 2016						
wk	M	T	W	T	F	S	S
49				1	2	3	4
50	5	6	7	8	9	10	11
51	12	13	14	15	16	17	18
52	19	20	21	22	23	24	25
53	26	27	28	29	30	31	

01	January 2017						
wk	M	T	W	T	F	S	S
53	30	31					1
01	2	3	4	5	6	7	8
02	9	10	11	12	13	14	15
03	16	17	18	19	20	21	22
04	23	24	25	26	27	28	29

16/08/17

Asymtotic Notation

09.00

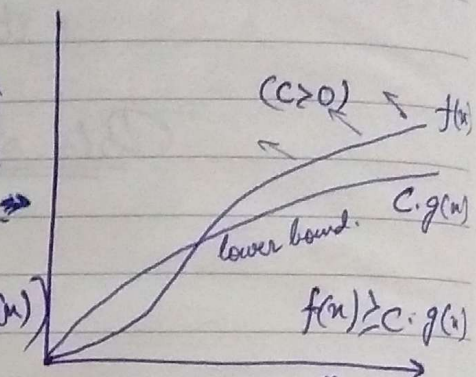
What is Big Oh ?

Thus: $f(n)$ is a function, $g(n)$ . . . . . .

. . . . . .

Big Oh → upper bound.

Big Ω → best case time complexity



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

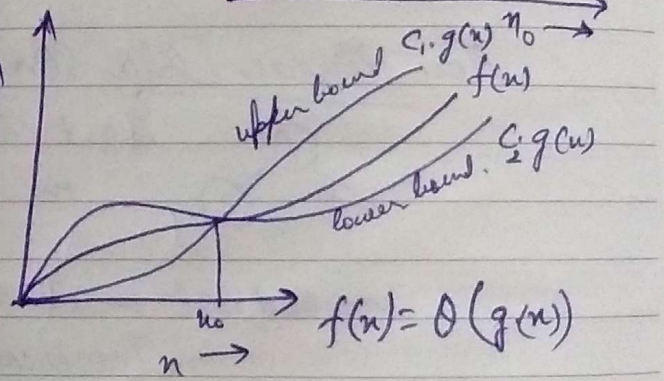
11.00

12.00

01.00

$C_1 \cdot g(n) \leq f(n) \leq C_2 \cdot g(n)$

$n_0 \geq 1$



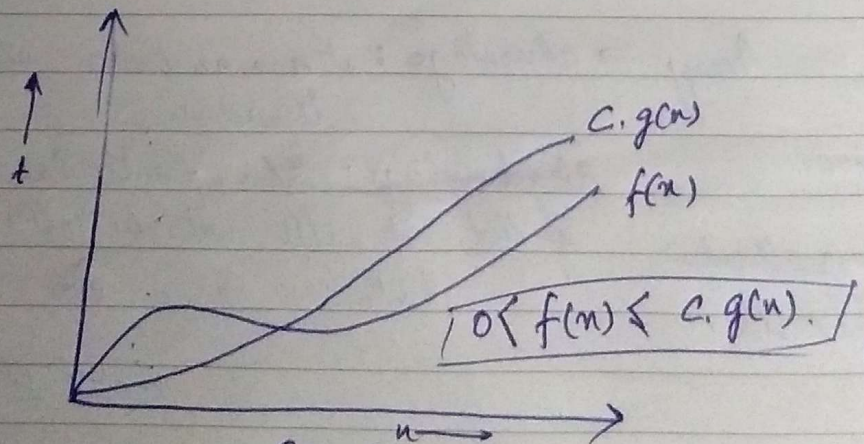
02.00

03.00

Graph showing Average Case.

04.00

05.00



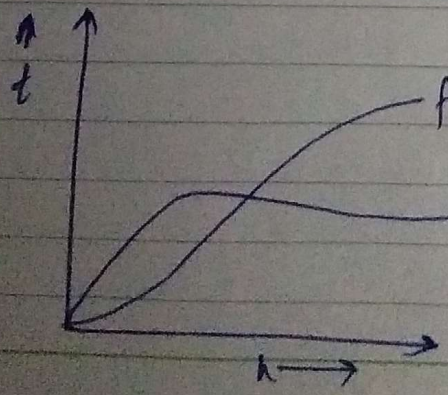
06.00

$0 \leq f(n) \leq C \cdot g(n)$.

Small oh.

07.00

Big Oh 'O' with restriction is small oh 'o'



$[\infty > f(n) \geq C \cdot g(n)]$.

Small omega (ω).

giving range for Big omega (Ω)

Time Complexity.

If it is not iterative or recursive then time complexity is $O(1)$.

```
A( )
{
  i=1;
  for(i=1 to n)
  print ("CSE");
}
```

→ $O(n)$.

```
for(i=1 to n)
  for(j=1 to n)
  print ("CSE");
```

→ $O(n^2)$

"Exceptions are possible."

```
A( )
{  i=1, s=1;
   while( s≤n)
   {  i++
      s=s+i;
      print("CSE");
   }
}
```

$\frac{n(n+1)}{2}$

s = 1 3 6 10 15 21 .. n

i = 1 2 3 4 5 6 .. k

No. of iterations

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

dominating

Hence time complexity.

$O(n^2)$.

Sunday 22

FEBRUARY MARCH APRIL MAY JUNE

23-01-2017 | 023-342 | 04

23 | 2017 January Monday

Find Time Complexity of the algo.

a) A( )

```
{
    i=1;
    for(i=1, i^2<=n; i++)
        print("CSE");
}
```

$s =$

i

$i^2 <= n$.

$i \sqrt{n}$

b) A( )

```
{
    for(i=1; i<=n; i++)
        for(j=1; j<=i; j++)
    for(k=1 k<=100 ; k++)
            print("CSE");
}
```