Data Structures

SY BTech(CSE)

Unit – 1

Introduction

PPT-7

Prof. Ms. Manisha A. Bhusa
Dept. Of CSE
COE Ambajogai

Objective:

Study of asymptotic notations.

Asymptotic Notations:

1. Big Oh (O)
2. Omega (Ω)
3. Theta (Θ)

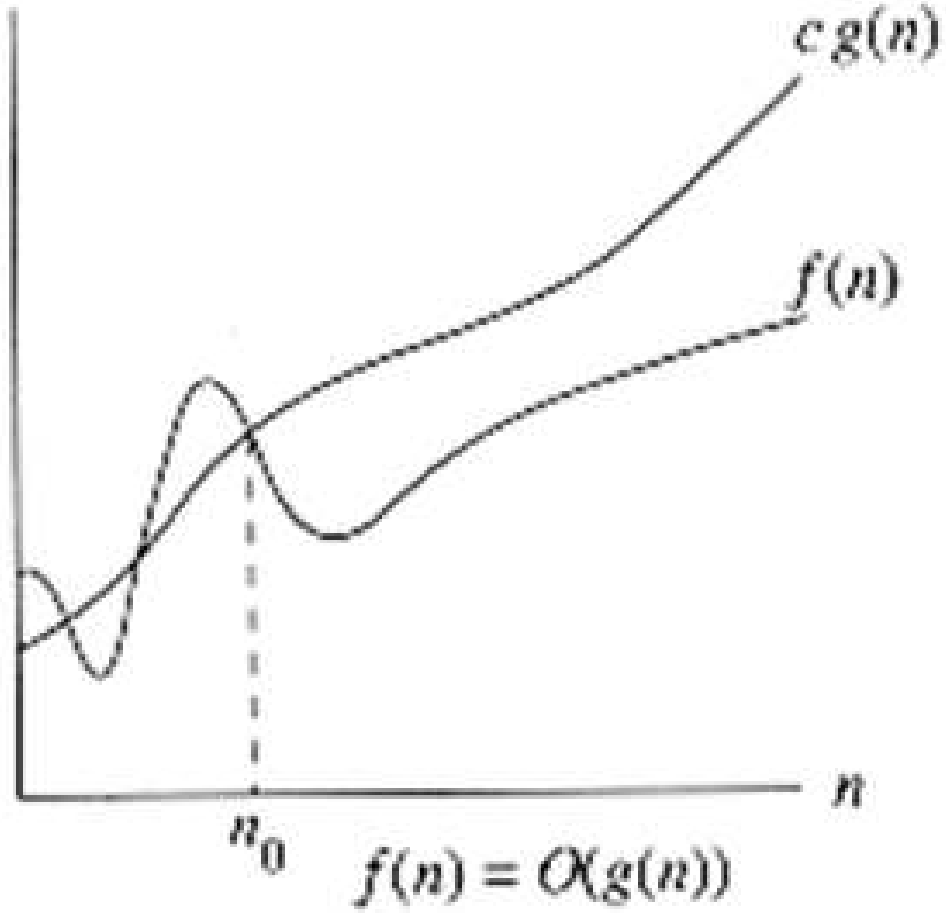
1. Big Oh (O): Gives Upper Bound

The function

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

iff there exists positive constants c and n0 such

that $f(n) <= c.\ g(n)$ for all n, $n>=n0$



Example of growth function

Ex: $f(n) = 3n + 2$

$c = 4$ $\quad\quad$ $g(n) = n$

$f(n) <= c.g(n)$

$3n+2 < 4n$

no.	f(n)		c.g(n)	Status
1	5	>	4	False
2	8	=	8	True
3	11	<	12	True
4	14	<	16	True

Complexity : O(n) for n>=2 & c = 4

Big Oh.xlsx

Ex: $f(n) = 10n^2+4n+2$

$c = 11$ $\qquad$ $f(n) <= c.g(n)$

$g(n) = n^2$ $\qquad$ $10n^2 + 4n + 2 < 11n^2$

no	f(n)		c. g(n)	Status
1	16	>	11	False
2	50	>	44	False
3	104	>	99	False
4	178	>	176	False
5	272	<	275	True
6	386	<	396	True

Complexity : $O(n^2)$ for n>=5 & c= 11

Big Oh.xlsx

Ex: $f(n) = 1000n^2 + 100n - 6$

Complexity: $O(n^2)$ for $n>=100$ & $c= 1001$

Ex: $f(n) = 6 \cdot 2^n + n^2$

no	2^n		n^2
1	2	>	1
2	4	=	4
3	8	<	9
4	16	=	16
5	32	>	25
6	64	>	36

Here, $2^n > n^2$ for $n>4$

Therefore, $c = 7$

$c \cdot g(n) = 7 \cdot 2^n$

Complexity: $O(2^n)$ for $c = 7$ and $n>=4$

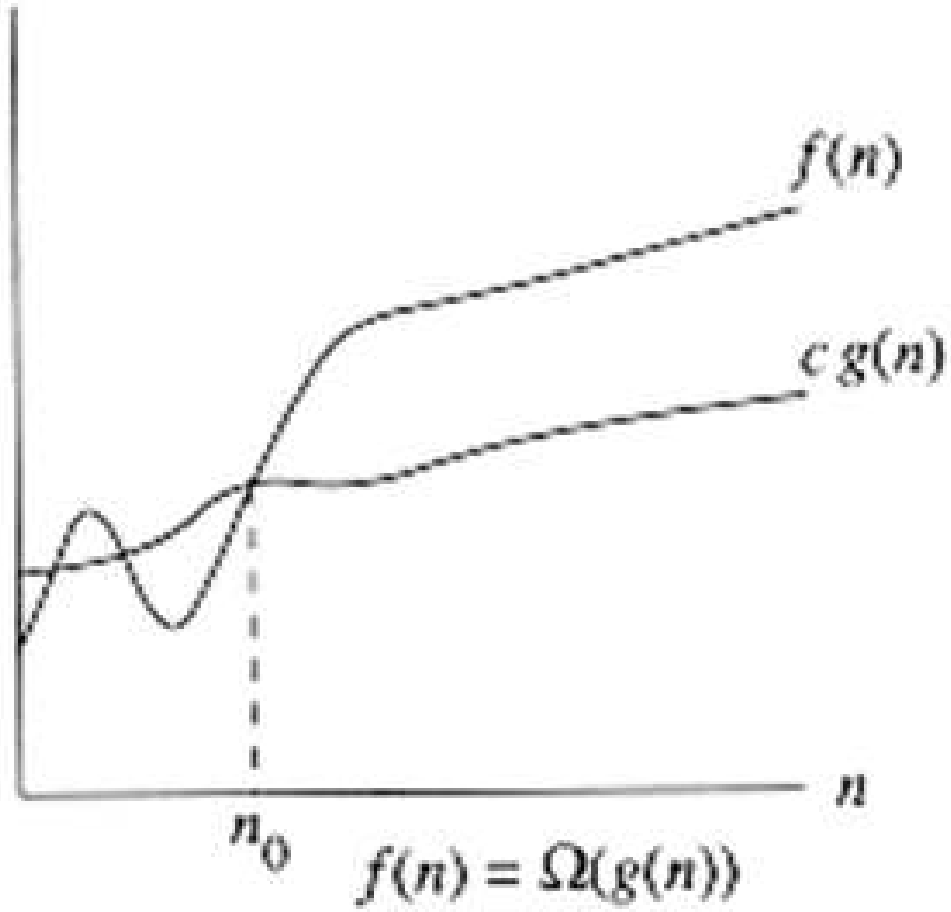
2. Omega (Ω)

The function

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

iff there exists positive constants c and n0 such that $f(n) >= c.\ g(n)$ for all n, n>=n0

Omega gives lower bound.



Example of growth function

Ex: $f(n) = 3n + 2$

$c = 3$

$g(n) = n$

no	f(n)		c.g(n)	Status
1	5	>	3	True
2	8	>	6	True
3	11	>	9	True

Complexity: $\Omega(n)$ for c=3 and n>=1

Omega.xlsx

Ex: $f(n) = 10n + 6$

$c = 11$ $\quad g(n) = n$

n0	f(n)		c.g(n)	Status
1	16	>	11	True
2	26	>	22	True
3	36	>	33	True
4	46	>	44	True
5	56	>	55	True
6	66	=	66	True
7	76	<	77	False
8	86	<	88	False

Omega.xlsx

Complexity: $\Omega(n)$ for c=11 and n<=6

$\Omega(n)$ for c=10 and n>=1

Ex: $f(n) = 10n^2 + 4n + 2$

$\Omega(n^2)$ for $c = 10$ and $n>=1$

Ex: $f(n) = 6.2^n + n^2$

$\Omega(2^n)$ for $c = 6$ and $n>=1$

3. Theta (Θ): Gives both upper and lower bound

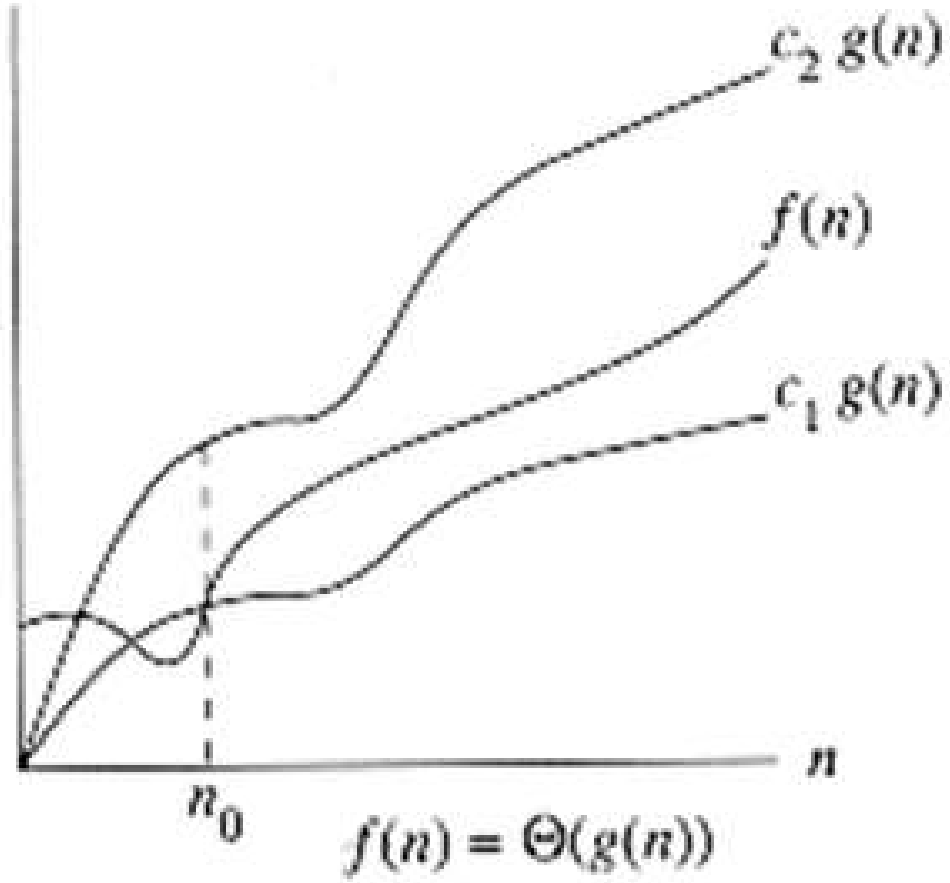
The function

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

iff there exists positive constants c1, c2 and n0

such that

$c1.g(n) <= f(n) <= c2.g(n)$ for all n, n>=n0



Example of growth function

Ex: $f(n) = 3n + 2$

$g(n) = n$

$c1.g(n) = 3n$ $c2.g(n) = 4n$

n0	c1.g(n)		f(n)		c2.g(n)	Status
1	3	<	5	>	4	False
2	6	<	8	=	8	True
3	9	<	11	<	12	True
4	12	<	14	<	16	True

Chart: Theta.xlsx

Complexity: $\Theta(n)$ for c1 = 3 , c2 = 4 and n>=2

Ex: $f(n) = 10n^2 + 4n + 2$

$c1.g(n) = 10n^2$

$c2.g(n) = 11n^2$

Complexity: $\Theta(n^2)$ **for**

$c1 = 10$ **and** $c2 = 11$ **and** $n>=4$

Ex: $f(n) = 2mn + 2m + 1$

$c1.g(mn) = 2mn$ $\quad\quad$ $c2.g(mn) = 3mn$

m	n	c1.g(mn)		f(n)		c2.g(mn)	Status
1	1	2	<	5	>	3	F
1	2	4	<	7	>	6	F
2	1	4	<	9	>	6	F
2	2	8	<	13	>	12	F
2	3	12	<	17	<	18	T
3	2	12	<	19	>	18	F
3	3	18	<	25	<	27	T

Complexity: $\Theta(mn)$ for $c1=2$, $c2=3$, $m>=1$ & $n>=3$