Tutorial-1.

Name : Abhinav Narain

Section : CST SPL-2

Semester: 4

Class R.NO: 41

University R.NO: 2017445

Date: 10th March 2022

Q1. Asymptotic Notation → They are the Mathematical Notations used to describe the running time of an algorithm when the input tends towards a particular value or a limiting time.

Different asymptotic notations :-

(i) Big O(n)

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

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

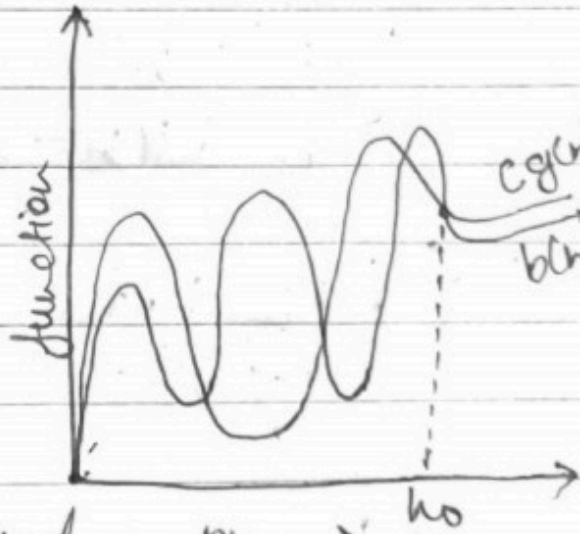
$f(n) \leq cg(n)$

$\forall\ n \geq n_0$

for some Constant,

$c > 0$

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



Ex → $f(n) = n^2 + n$

$g(n) = n^3$

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

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

(ii) Big Omega (Ω)

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

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

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

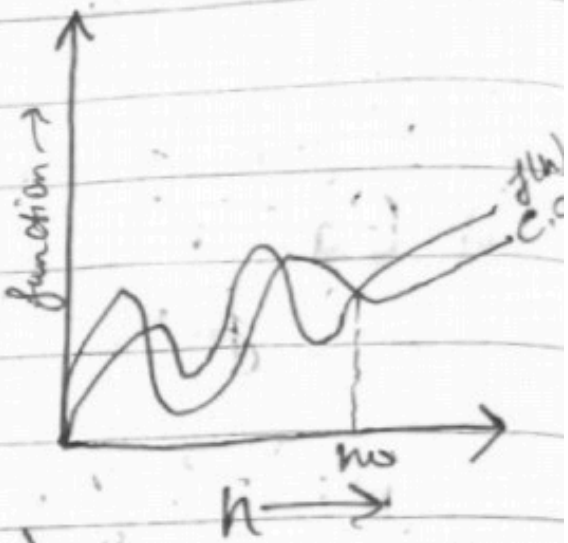
iff

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

$\forall\ n \geq n_0$

for some Constant:

$c > 0$



Ex:-

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

$g(n) = n^2$

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

(iii) Big Theta (Θ)

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

$g(n)$ is both "tight" upper & lower bound of function $f(n)$.

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

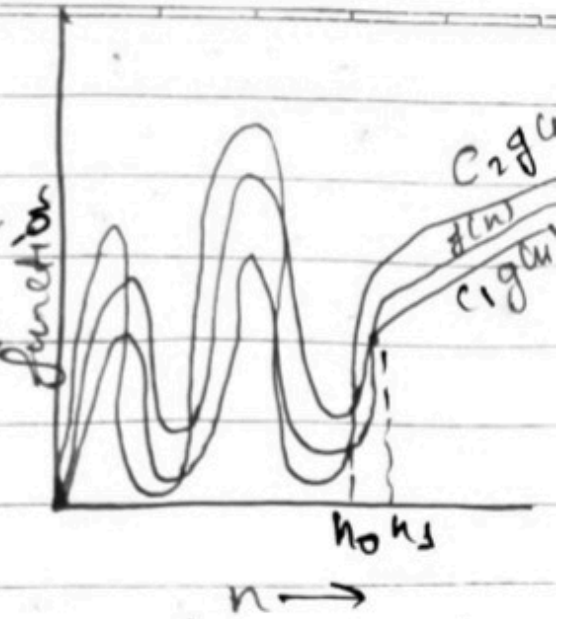
iff

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

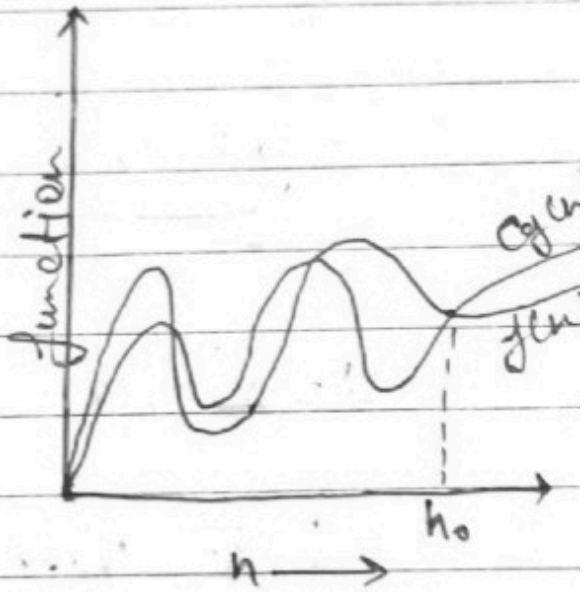
$\forall\ n \geq \max(n_1, n_2)$

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

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



(iv) Small o(o)
$f(n) = o(g(n))$
$g(n)$ is upper bound of function $f(n)$.
$f(n) = o(g(n))$
when $f(n) < cg(n)$
$\forall n > n_0$
& $\forall$ Constant, $c > 0$

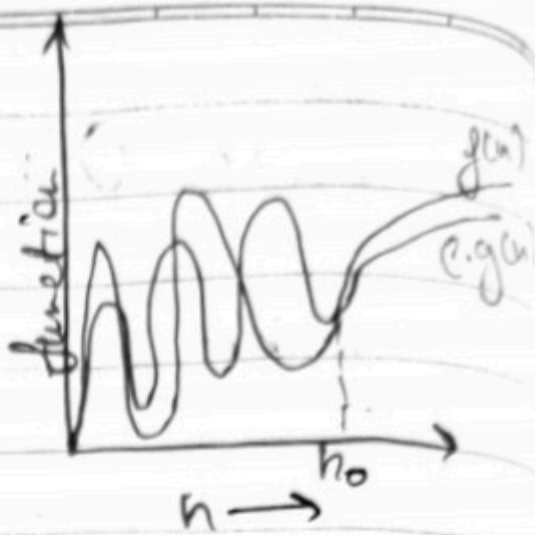


Ex → $f(n) = n^2$
$g(n) = n^3$
$n^2 = o(n^3)$

(v) Small Omega(ω)
$f(n) = \omega(g(n))$
$g(n)$ is lower bound of $f(n)$
$f(n) = \omega(g(n))$
when $f(n) > cg(n)$
$\forall n > n_0$
& $\forall$ Constant ; $c > 0$

$f(n) = 4n + 6$

$g(n) = (1)$



Q2. for(i = 1 to n)
{ i = i * 2; }

$\rightarrow$ i = 1, 2, 4, 8, 16, ..., n (GP)

$\leftarrow O(K)$

K

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

GP K^{th} value $= t_k = ar^{k-1}$

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

$n = 2^{k-1}$

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

$= \log(2n) = k \log 2$

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

$O(k) = O(1 + \log n)$

$= O(\log(n))$

Q3. $T(n) = \{3T(n-1)$ if $n>0$, otherwise $1\}$

→ $T(n) = 3T(n-1)$ — (1)

Put $n = n-1$

$T(n-1) = 3T(n-2)$ — (2)

from 1 to 2

$T(n) = 3(3T(n-2))$

$= 3T(n-2)$ — (3)

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

$T(n) = 3[T(n-3)]$ — (4)

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

$T(n) = 3(T(n-k))$ $\{k = 3\}$

Put, $n-k = 0$

$n = k$

$T(n) = 3^n[T(n-n)]$

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

$\{T(0) = 1\}$

$T(n) = 3^n \times 1$

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

Q4. $T(n) = \begin{cases} 2T(n-1) & , n > 0 \\ 1 & , n \leq 0 \end{cases}$

Using Backward Substitution;

$T(n) = 2T(n-1)$
$T(n-1) = 2T(n-2)$
$T(n-2) = 2T(n-3)$
.
.
$T(1) = 2T(0)$
$T(0) = 1$

} n level

Substituting value of $T(n-1)$, $T(n-2)$... till $T(1)$ in eqⁿ T we get;

$T(n) = 2^n \times T(0)$
$\therefore T(n) = 2^n \times 1$
$= O(2^n)$

Q5. What should be time Complexi

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

→ i = 1, 2, 3, 4, 5, 6, ... n

S = 1 + 3 + 6 + 10 + 15 + 21 + ... + n

0 = 1 + 2 + 3 + 4 + ... k - T(n)

1 + 2 + 3 + 4 + ... + k > n

$\frac{k(k-1)}{2} > n$

$\frac{k^2+k}{2} > n$

$\frac{k^2+k}{2} > n$

$k^2 > n$

$k = O(\sqrt{n})$

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

Q6. Void function (int n)

{

int i, Count = 0;

for(i = 1; i * i ≤ n; i++)

Count ++;

}

→ i = 1, 2, 3, ..., n

i^2 = 1, 4, 8, ... n

$i^2 \leq n$ & $i \leq \sqrt{n}$

k^{th} term, $t_k = a + (k-1)d$

$a = 1, d = 1$

$t_k \leq \sqrt{n}$

$\sqrt{n} = 1 + (k-1)1$

$\sqrt{n} = k$

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

Q7.

i	j	k
$\frac{n}{2}$	$\log(n)$	$\log(n)$
$\frac{n}{1}$	:	:
n	$\log(n)$	$\log n$

$\frac{n+1}{2}$ th

$O(i * j * k) = O(\frac{(n+1)}{2} * \log n * \log$

$= O(\frac{(n+1)}{2} * (\log_2 n)^2)$

$T(n) = O(n(\log_2 n)^2)$.

Q8. $T(n) = T(n-3) + n^2$ — (1)

$T(1) = 1$ — (2)

Put $n = n-3$ — (1)

$T(n-3) = T(n-6) + (n-3)^2$ — (3)

Put (3) in (1)

$T(n) = T(n-6) + (n-3)^2 + n^2$ — (4)

Put $n = n-6$ in equation (1)

$T(n-6) = T(n-9) + (n-6)^2$ — (5)

Put (5) in (4).

$T(n) = T(n-9) + (n-6)^2 + (n-3)^2 + n$

$T(n) = T(n-3k) + (n-3(k-1))^2 + (n-3(k-2))^2$

$+ \ldots n^2$

$n + 3k = 1$

$\frac{n-1}{3} = k.$

$T(n) = T(1) + \left(n - 3\left(\frac{n-1}{3} - 1\right)\right)^2 + \left(n - 3\left(\frac{n-1}{3} - 2\right)\right)^2$

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

$T(n) = 1 + 4^2 + 6^2 + \ldots n^2$

$= n^2 + \ldots + 1.$

$T(n) = O(n^2).$

Q9.

i	j
1	n times
2	$1+3+5+\ldots+n$ tim

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

$a = 1, \; d = 2$

$n = 1 + (k-1) \cdot 2$

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

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

$k = \frac{n+1}{2}$; no of terms.

for $i=2$ $j = \frac{n+1}{2}$ times

for $i=3$ $j = 1+4+7+\ldots+n$ times

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

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

$\frac{n-1}{3}+1 = k$

$k = \frac{n+2}{3}$; no. of terms.

Generalising;

for $i=n$ $j = \frac{n+k-1}{k}$ times.

$T(n) = \underbrace{n + \frac{n+1}{2} + \frac{n+2}{3} + \ldots \frac{n+k-1}{k}}_{n \text{ terms.}}$

General terms $= \frac{n+k-1}{k}$

$$\sum_{i=1}^{n} \frac{n+k-1}{k} \Rightarrow \frac{\sum_{i=1}^{n} n + \sum_{i=1}^{n} k - \sum 1}{k}$$

$$\Rightarrow \frac{n(n+1)}{2} + nk - n$$

→ $\frac{n^2 + \frac{n}{2} + nk - n}{k}$

→ $\frac{n^2 + \frac{n}{2} + nk - n}{k}$

after removing Constant te

$$T(n) = O(n^2)$$

Q10 $n^k = O(c^n)$

as $n^k \leq a \cdot c^n$

$\forall\ n \geq n_0$ for some Constant

for $n_0 = 1$

$c = 2$

$\Rightarrow n^k \leq O(2)$

$n_0 = 1$ & $c = 2$.

— X —