

Tutorial - 1.

Name : Abhiraj Brautam

Section : CST SPL-2

Semester: 4

Class R. No : 05

University R. No : 2018187

Date : 10th March 2022

Q1. Asymptotic Notation \rightarrow 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 ($O(n)$)

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

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

iff

$$f(n) \leq Cg(n)$$

$$\forall n \geq n_0$$

for some Constant,
 $C > 0$

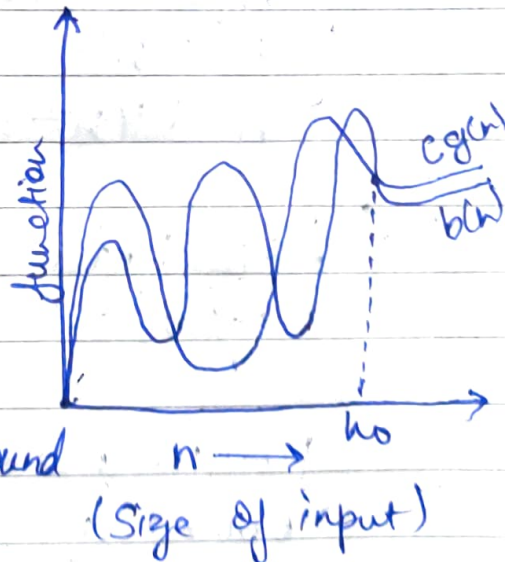
$g(n)$ is "tight" upper bound of $f(n)$.

$$\text{ex} \rightarrow f(n) = n^2 + n$$

$$g(n) = n^3$$

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

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



(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

$$f(n) \geq c \cdot g(n) \quad \forall n \geq n_0$$

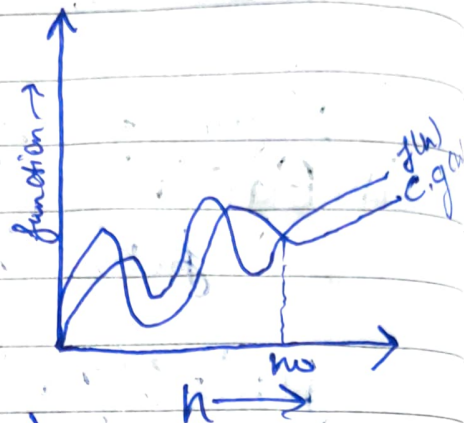
for some Constant;
 $c > 0$

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 $f(n)$.

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

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

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

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

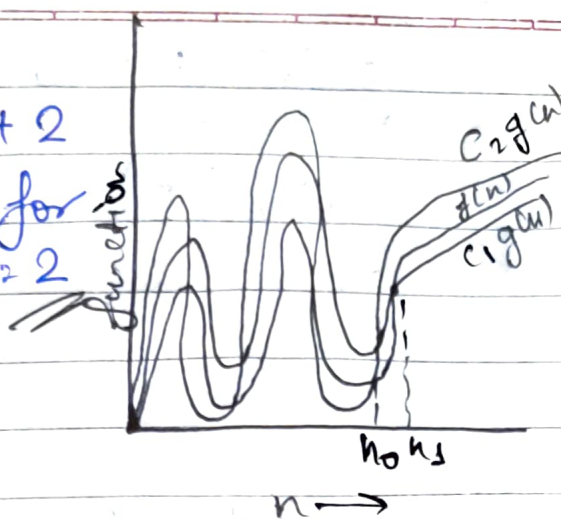
Abhishek

PAGE _____

DATE _____

Ex →

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



(iv) Small $o()$

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

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

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

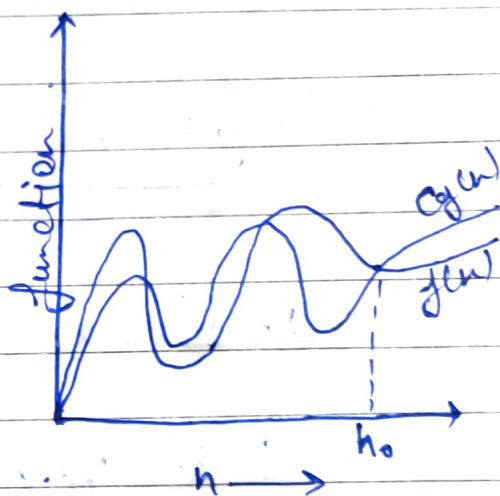
when $f(n) < Cg(n)$
 $\forall n > n_0$

& \forall Constant, $C > 0$

Ex → $f(n) = n^2$

$g(n) = n^3$

$n^2 = o(n^3)$



(v) Small $\Omega()$

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

$g(n)$ is lower bound of $f(n)$

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

when $f(n) > Cg(n)$

$\forall n > n_0$

& \forall Constant; $C > 0$

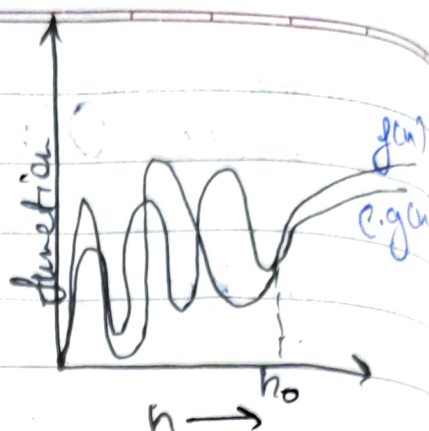
Abhishek

PAGE _____

DATE _____

$$f(n) = 4n + 6$$

$$g(n) = (1)$$



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

→ $i = 1, 2, 4, 8, 16, \dots, n$ (G.P.)
 $\underbrace{\hspace{10em}}_k$ ~~$O(n)$~~ $\rightarrow O(k)$

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

$$\therefore \text{GP } k^{\text{th}} \text{ value} = t_k = ar^{k-1}$$

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

$$n \geq 2^{k-1}$$

$$n \geq \frac{2^k}{2} \Rightarrow 2n \geq 2^k$$

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

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

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

Alho

PAGE _____

DATE _____

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

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

Put $n = n-1$

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

from 1 to 2

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

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

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

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

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

$$\text{Put, } n-k = 0 \\ n = k$$

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

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

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

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

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

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

Using Backward Substitution;

$$\left. \begin{array}{l} T(n) = 2T(n-1) \\ T(n-1) = 2T(n-2) \\ T(n-2) = 2T(n-3) \\ \vdots \\ T(1) = 2T(0) \\ T(0) = 1 \end{array} \right\} n \text{ levels}$$

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

$$\begin{aligned} T(n) &= 2^n \times T(0) \\ \therefore T(n) &= 2^n \times 1 \\ &= O(2^n) \end{aligned}$$

Q5. What should be time Complexity of

```
int i = 1, S = 1;
while (S ≤ n)
```

{

 i++; S = S + i;

 Printf("#");

}

→ $i = 1, 2, 3, 4, 5, 6, \dots$

$$S = 1 + 3 + 6 + 10 + 15 + 21 + \dots + n$$

$$O = 1 + 2 + 3 + 4 + \dots + n = T(n)$$

$$1 + 2 + 3 + 4 + \dots + k > n$$

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

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

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

$$k^2 > n$$

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

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

Q6. void function (int n)

```

{
    int i, Count = 0;
    for (i = 1; i * i ≤ n; i++)
        Count++;
}
    
```

→ $i = 1, 2, 3, \dots, n$
 $i^2 = 1, 4, 9, \dots, n$

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

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

$$a = 1, d = 1$$

$$t_k \leq \sqrt{n}$$

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

$$\sqrt{n} = k$$

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

Q7.

$$\begin{matrix} i \\ \frac{n}{2} \end{matrix}$$

$$\begin{matrix} j \\ \log(n) \end{matrix}$$

$$\begin{matrix} k \\ \log(n) \end{matrix}$$

$$\begin{matrix} \frac{n}{1} \\ \vdots \end{matrix}$$

$$n$$

$$\log(n)$$

$$\log n$$

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

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

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

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

$$Q8. T(n) = T(n-3) + n^2 \quad \text{--- (1)}$$

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

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

$$T(n-3) = T(n-6) + (n-3)^2 \quad \text{--- (3)}$$

$$\text{Put (3) in (1)}$$

$$T(n) = T(n-6) + (n-3)^2 + n^2 \quad \text{--- (4)}$$

$$\text{Put } n = n-6 \text{ in equation (1)}$$

$$T(n-6) = T(n-9) + (n-6)^2 \quad \text{--- (5)}$$

$$\text{Put (5) in (4).}$$

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

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

$$+ \dots + n^2$$

$$n - 3k = 1$$

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

Ashir

PAGE _____

DATE _____

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

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

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

$$= n^2 + \dots + 1$$

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

Q9.

i

1

2

j

n times

1+3+5+...+n times

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

$$a=1, d=2$$

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

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

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

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

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

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

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

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

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

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

Generalising;

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

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

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

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

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

$$\rightarrow n^2 + \frac{n}{2} + nk - n$$

k

$$\rightarrow n^2 + \frac{n}{2} + nk - n$$

k

after removing Constant term

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

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

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

$\forall n \geq n_0$ for some Constant a
for $n_0 = 1$

$c = 2$

$\rightarrow 1^k \leq O(2)$

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

— X —