[AU APR/MAY 2010) → 8 marks
10: May-10, 11, 12, 13; Dec 10, 1

Asymptotic Notations and its properties

↳ Asymptotic

For choosing the best algorithm, we need to check the efficiency of each algorithm. The efficiency can be measured by computing the time complexity of each algorithms.

* Asymptotic Notation → time complexity represents

↓

Using this → we can give time complexity as,

1) fastest possible.
2) slowest possible.
3) Average time.

→ Various Notations:

Big oh Notation ⇒ Given no faster

- denoted by 'O'.
- It is the method of representing the upper bound of algorithm's running time.

A function f(n) is said to be in O(g(n)) if f(n) is bounded above by some constant multiple of g(n) for all large n.

Defn: Let f(n) and g(n) be 2 non negative fns. Let no and constant c → 2 integers.

no → denotes some value of i/p & $n > n_o$.

c → some constant, such that c>

ie)
1) f(n) & g(n) → Non-negative fn.
2) no & c → integers

where $n > n_o$, $c > 0$

⇓

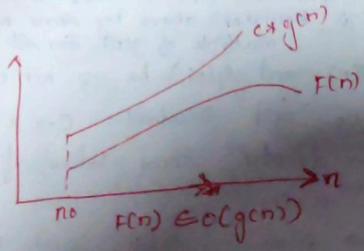
We can write,

- f(n) is big O of g(n)
- denoted as $f(n) \in O(g(n))$
- $f(n) < g(n)$ if g(n) is multiple of some const c

$$f(n) \leq c * g(n)$$

no → constant

↳ f(n) & g(n) are fns.



Eg:1 Consider function $F(n) = 2n+2$ and $g(n) = n^2$. Then we have to find some constant c, so that $F(n) \leq c \times g(n)$. As $F(n) = 2n+2$ and $g(n) = n^2$ then we find c for $n=1$.

function $f(n) = 2n+2$.

Let $n=1$, then

$F(1) = 2(1)+2$

$= 4$

& $g(n) = n^2$

$g(1) = 1$

ie) $F(n) > g(n)$

if $n=2$, then

$F(2) = 2(2)+2 \Rightarrow 6$

$g(2) = 4$

ie) $F(n) > g(n)$

if $n=3$, then

$f(3) = 2(3)+2 \Rightarrow 8$

$g(3) = 9$

ie) $F(n) < g(n)$ is true.

Hence we can conclude that for $n \geq 2$, we obtain,

$F(n) < g(n)$

Thus the upper bound of existing time is obtained by big oh notation

→ Omega Notation ⇒ at least as fast.

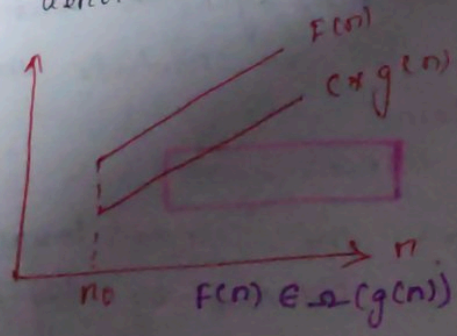
- denoted by Ω
- represented the lower bound of algorithm's running time.
- denotes shortest amt. of time taken by algthm.

Defn:

A function $F(n)$ is said to be $\Omega(g(n))$ if $F(n)$ is bounded below by some positive constant multiple of $g(n)$ such that,

$F(n) \geq c * g(n)$ for all $n \geq n_0$

It is denoted as $F(n) \in \Omega(g(n))$



Example

Consider $F(n) = 2n^2+5$ and

$g(n) = 7n$

if $n=0$

$F(0) = 2(0)+5 = 5$

$g(0) = 0$ i.e) $F(n) > g(n)$

if $n=1$

$F(1) = 2+5 \Rightarrow 7$

$g(1) = 7$ i.e) $F(n) = g(n)$

if $n=3$

$F(3) = 18+5 \Rightarrow 23$

$g(3) = 21$ i.e) $F(n) > g(n)$

Thus for $n \geq 3$, we get $F(n) > c * g(n)$

It can be represented as.

$2n^2+5 \in \Omega(n)$

(or) any $n^3 \in \Omega(n)^2$.

Θ Notation → at same rate

- denoted by 'Θ'. By this method the running time is b/w upper bound & lower bound.

Defn:

- Let $F(n)$ & $g(n)$ be 2 Non-negativ

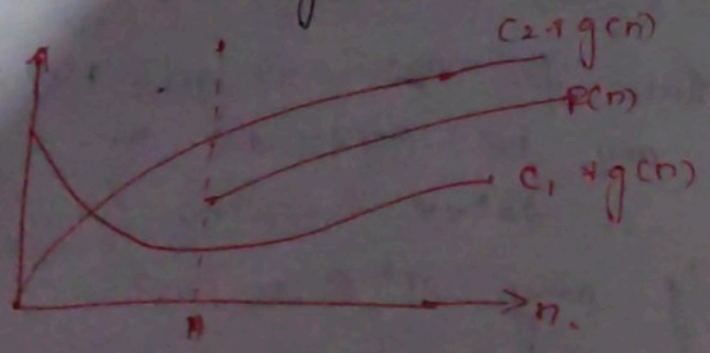
for

There are 2 positive constant. namely c_1 and c_2 such that.

$c_1 g(n) \leq f(n) \leq c_2 g(n)$ ✗.

Then we can say that, $c_1 * g(n) \leq F(n) \leq c_2 * g(n)$

$F(n) \in \Theta(g(n))$



$F(n) \in \Theta(g(n))$.

Eg:

if $f(n) = 2n+8$ & $g(n) = 7n$.

where $n \geq 2$

Ily $F(n) = 2n+8$

$g(n) = 7n$.

$f(n) = 10n^3 + 5n^2 + 17$

$c_1 = 10$ $c_2 = 32$

$c_1(n^3) \leq f_1(n) \leq$

ie) $5n < 2n + 8 < 7n$ for $n \geq 2$

Here, $c_1 = 5$ & $c_2 = 7$ with $n_0 = 2$

The theta notation is more precise with both big oh & omega Notation.

Some eg. of Asymptotic Order:

1) $\log_2 n$ is $F(n)$ then

$\log_2 n \in O(n)$ ∴ $\log_2 n \leq O(n)$. the order of growth of $\log_2 n$ is slower than n.

$\log_2 n \in O(n^2)$ ∴ $\log_2 n \leq O(n^2)$, the order of growth of $\log_2 n$ is slower than n^2 as well.

But.

$\log_2 n \in \Omega(n)$ ∴ $\log_2 n \leq \Omega(n)$

if a Condition for $F(n)$ is belonging to $\Omega(n)$ it should satisfy the co