ALGORITHMS

Date ____ Day ____

Diff b/w algo and Program

1) Algo is written in design phase whereas Program is written in implementation phase.
2) Needs domain knowledge, Needs programmer knowle
3) Can be written in any lang, Must be written in Program lang
4) Independend of hardware & software, depe on H/w, S/w.
5) use to anaylyes time & space, we test program for faults.

"Types Of Analysis Of Algo"

1) Experimental or Apostrium or Relative analysis:
Means analysis of algo after it is converted to code.
↳ Advantage: Exact values no rough
↳ Disadvantage: Final result not only dep on algo but also include factor → hardware, software, programming lang etc.

2) Apriori or Independent analysis or Absolute analysis:
We don analysis using asymptomatic notations and mathemetical tools of only algo.
Adv: Result depends only on Algo
Dis: Result is estimated or approx no precise value.

Day

Date

" ASYMPTOTIC NOTATIONS "

① BIG O NOTATION :

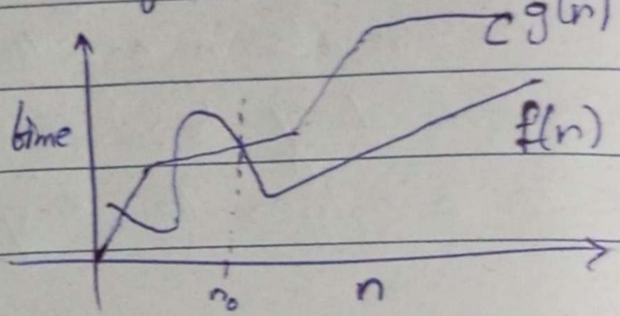
↳ Defines upper bond of an algo.

$f(n) \leq c \, g(n)$ → isse kuch farq nhi padta.

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

For $N >= N_0$

jab no. of inputs ki val N_0 se badi hogi to hi ye valid hoga.



g(n) will always be greater than f(n) after n_0.

↳ g(n) is the tight upper bound of f(n).
↳ least

Q: Assume that $f(n) = 5n+50$ and $g(n) = n$. is $f(n) = O(g(n))$

↳ first finding leading term in f(n) i.e 5n. [for large val of n 50 will not matter]

↳ Now we assume the value of c = 6 (jo dominant term ke saath val hogi usse ek zyada le lena)

n	5n +50 (f(n))	6n (c.g(n))
10	100	60
20	150	120
50	300	300
51	305	306

∴ we can say that g(n) is asymptotically greater than f(n) with c = 6 & $n_0 \geq 51$

| $5n+50 = O(n)$ for c = 6 & $n_0 \geq 51$

Page No.

Date ____________ Day ____________

$O(n)$ is the least upper bound (closest biggest val) of $5n+50$. $O(n^2)$, $O(n^3)$, $O(2n)$ are also upper bound but least upperbound is $O(n)$.

Q: Assume that $f(n) = 5n+50$ & $g(n) = \log_{10} n$. Is $g(n)$ the upper bound of $f(n)$?

$c = 1000$

n	$5n+50$	$1000 \log_{10} n$
10	100	1000
10^2	550	2000
10^3	8050	3000
10^4	50050	4000

$g(n)$ is not an upper bound of $f(n)$.

$3n+8n$

$12n$

Q: Find the upper bound for $f(n) = 3n+8$

STEP#01: Find the dominant term $\Rightarrow 3n$

STEP#01: Choose $g(n)$ acc to dominant term.

We can take n, $n\log n$, n^2, n^3 ... as $g(n)$ but we will take L.U.B (Least upper bound).

$g(n) = n$

STEP#03: Now Proof $c = 4$

n	$3n+8$	$4n$
7	29	28
8	32	32
10	38	40
100	308	400

Page No. ____

$n_0 = 8$

$f(n) = O(g(n))$ for $c = 4$ & $n \geq 8$

Date ______ Day ______

- $f(n) = n^2 + 10$

Dominant term = n^2

$g(n) = n^2$

Now Taking $c = 2$

n	n^2+10	$2n^2$
1	11	2
2	14	8
3	19	18
4	26	32

$\boxed{n^2 + 10 = O(n^2) \text{ for } c = 2 \ \& \ n \geq 4}$

Q: Find the upper bound $f(n) = 2n^3 - 2n^2$

① Dominant term: $2n^3$

② $g(n) = n^3$, $c = 2$ (f(n) mein $2n^3$ mein se $2n^2$ minus ho rha hai. To kisi na kisi point par f(n) $2n^3$ se chota hoga hi.

③

n	$2n^3 - 2n^2$	$2n^3$
1	0	2
2	8	16
3	36	54
4	96	128

$\boxed{n_0 = 1}$, $\boxed{c = 2}$

Q: $f(n) = n^4 + 100n^2 + 35$

① Dominant term = n^4

② Let $g(n) = n^4$, $c = 2$

n	n^4+100n^2+35	$2n^4$
1	136	2
2	451	32
10	20,035	20,000
11	26,776	29282

$\boxed{n_0 = 11, \ c = 2}$

Date

2^n ≥ 3n^3 k baad hum ne
Se saari values check krein
no can be either 13, 14, 15....

Day

Q: $2^n + 3n^3$

1) Dominant term: 2^n

2) Let $g(n) = 2^n$, $c = 2$

n	$2^n + 3n^3$	2^{n+1}
1	5	4
2	28	8
10	4024	2048
11	6041	4096
12	9280	8192
13	14783	16384

$\boxed{n_0 = 13, \; c = 2}$

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

Q: $f(n) = 300$

1) Dominant term: 300

2) Let $g(n) = 1$, $c = 300$

$300 \leq 300$

$\therefore 300 = O(1)$ for $c = 300$ & $n_0 = 1$

$\frac{1}{2}n^2 - 3n$

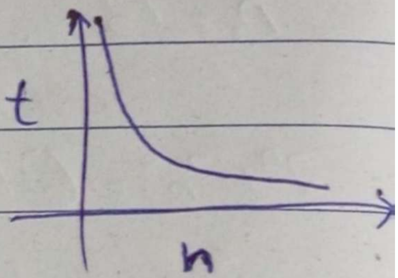
Date ____________ Day ____________

"FUNCTIONS IN ASYMPTOTIC NOTATIONS"

① Decrement Functions:

A function in which the denominator is bigger than the numerator.

Eg: $\frac{c}{n}, \frac{n}{n^2}, \frac{n}{2^n}, \frac{n^2}{3^n} \ldots$

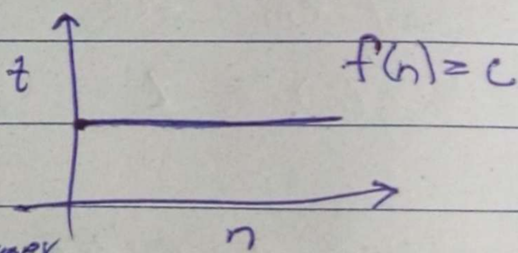


This func decr as the size of inp incr.

② Constant Functions:

A function parallel to x-axis

Eg: 100, 50029, 1 billion



→ Constant func is asymptotically bigger than the decrement func.

$f(n) = 100/n \quad g(n) = 10000$. Is $f(n) = O(g(n))$?

→ Humein g(n) ko ~~[illegible]~~ as small as possible krenge

$\therefore c = 1/10000 \quad g(n) = 1$

n	100/n	1
2	50	1
4	25	1
100	1	1
1000	0.1	1

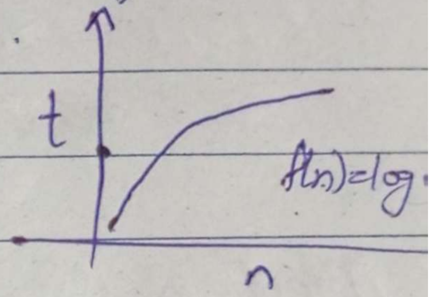
$\boxed{n_0 = 100}$

Date ______ Day ______

③ Logarithmic Functions:

A func that grows +vely, but with slower growth rate.

Ex: $\log n$, $(\log n)^{10}$, $\log(\log n)$ etc.



→ Logarithmic func is asymptotically bigger than the constant function

~~$f(n) = 100$; $g(n) = \log_{10} n$~~ , $c = 1$?

n	100	$\log_{10} n$
10	100	1
10^{100}	100	100
10^{1000}	100	1000.

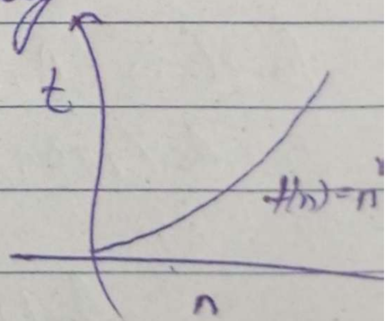
$\boxed{n_0 = 10^{100}}$

④ Polynomial Functions:

A func whose growth rate incr polynomially

Ex: $n^{0.01}$, $\sqrt{n}$, n^2, $n\log n$, n^k (k is +ve) ---

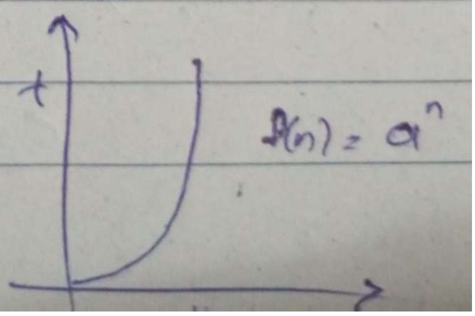
→ Poly time is asy bigger than log func.



③ Exponential Functions:

A func whose growth rate incr rapidly

Ex: 2^n, 3^n, $n!$, n^n, a^n etc



→ It is asym bigger than poly func.

Day

Date

Conclusion :

Decr Func < Const Func < Log Func < Poly Func < Exp Func

"Frequency Count Method"

Algorithm sum (A, n)
{

S = 0 ——— 1

for (i=0 ; i<n ; i++) ——— n+1 $\qquad$ 1+n+1+n+1

S = S + A[i] ——— n

return S; ——— 1

Time:

$f(n) = 2n + 3$

$O(n)$ dominant term

Space:

A — n

n — 1

S — 1

i — 1

$S(n) = n+3 \rightarrow O(n)$

- Sum of matrices

function (A, B, n){

for (i=0 ; i<n ; i++) → n+1

{

for(j=0; j<n; j++) → n × (n+1)

{

c[i,j] = A[i,j] + B[i,j]; → n×n

}

}

Date ______ Day ______

$f(n) = 2n^2 + 2n + 1 \quad O(n^2)$

$S(n) = 3n^2 + 3 \quad O(n^2)$

$A \to n^2 \quad C \to n^2$
$B \to n^2 \quad i \to 1$

- Matrix Multiplication:

```
for (i=0; i<n; i++) {                 —— n+1
    for (j=0; j<n; j++) {             —— n×(n+1)
        c[i,j] = 0                    —— n×n
    }   for (k=0; k<n; k++) {         —— n×n×n+1
            c[i,j] = c[i,j] + A[i,k] * B[k,j];
        }                             —— n×n×n
    }
}
```

$f(n) = 2n^3 + 3n^2 + 2n + 1 \quad O(n^3)$

$S(n) = 3n^2 + 4 \quad O(n^2)$

---

① for (i=1; i<n; i+=2) $O(n)$

Stmt $\longrightarrow n/2$

for (i=1; i<n; i+=20) $O(n)$

Stmt $\longrightarrow \frac{n}{20}$

Page No. ______

"Calculating Time Complexity"

① for (i=n ; i>0 ; i--)

stmt $\longrightarrow n$ $O(n)$

② for (i=1 ; i<n ; i=i+2)

stmt $\longrightarrow n/2$ $O(n)$

i=i+20

$\longrightarrow n/20$ $O(n)$

③ for (i=0 ; i<n ; i++) {

for (j=0 ; j<i ; j++)

{

stmt; }

}

i	j	no. of times
0	0	0
1	0,1	1
2	0,1,2	2
n	0--n	n

total time = 0+1+2+3 --- n

Sum of natural no. $= \frac{n(n+1)}{2}$

$f(n) = \frac{n^2+1}{2}$

$\rightarrow O(n^2)$

CamScanner

Date   Day

(4) $p=0;$
$for(i=1;\ p<=n;\ i++)$
$p = p+i$

i	p	n
1	$0+1=1$	
2	$1+2=3$	
3	$3+3=6$	
4	$6+4=10$	
⋮		
k	$1 \ldots +k =$	

$p = \frac{k(k+1)}{2}$

if the loop stops at 'k', then Assume $p>n$

$\because p = \frac{k(k+1)}{2}$

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

Now, $k^2 > n$

$k > \sqrt{n} \Rightarrow O(\sqrt{n})$

→ whenever the 'i' val is multiplying than it will take $\log_2(n)$ time.

(5) $for(i=1;\ i<n;\ i*=2)$
$stmt;$

i
1
$1\times 2 = 2^1$
$2^1\times 2 = 2^2$
$2^2\times 2 = 2^3$
$2^3\times 2 = 2^4$
⋮
2^k

Assume $2^k >= n$ (so loop stops)

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

$k \log_2 2 >= \log(n)$

$\boxed{k = \log_2(n)}$

we take ceil value of $\log_2 n$ i.e. $\lceil \log(n) \rceil$

Page No.

loop k andar wali ko n+1 likho ya n baat ekhi hai

Date ________ Day ________

(10) for(i=0 ; i<n ; i++) → ye 'n' times chalega → n
{
    for(j=1 ; j<n ; j=j*2){ → n × log n
        Stmt ; → n × log n
    }
}

$n + n\log n + n\log n$

$f(n) = 2n\log n + n$

$O(n) = n\log(n)$

- Summary:

for(i=0 ; i<n ; i++) → O(n)
for(i=0 ; i<n ; i=i+2) → n/2 O(n)
for(i=n ; i>1 ; i--) → O(n)

n/2 → O(n)
n/200 → O(n)
i = i+200

for(i=1 ; i<n ; i=i×2) → $O(\log_2 n)$
for(i=1 ; i<n ; i=i×3) → $O(\log_3 n)$
for(i=n ; i>1 ; i=i/2) → $O(\log_2 n)$

Date ______ Day ______

" ANALYSIS OF IF & WHILE "

① $a = 1;$
```
while (a < b) {
    stmt;
    a = a * 2;
}
```

a
1
2
2^2
2^3
2^k

Assum $a >= b$

$2^k >= b$

$k = \log_2 b$

$O(\log_2 b)$

② $i = n$
```
while (i > 1) {
    stmt
    i = i/2
}
```
$\rightarrow O(\log n)$

③ $i = 1,\ k = 1;$
```
while (k < n) {
    stmt;
    k = k + i;
    i++;
}
```

i	k
1	1
2	$1 + 1 = 2$
3	$2 + 2$
4	$2 + 2 + 3$
m	$2 + 2 + 3 + 4 \cdots + m = \frac{m(m+1)}{2}$

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

$\frac{m^2 + m}{2} = n \rightarrow m^2 \geq n \quad m = \sqrt{n}$

$O(n)$

Page No. ______

Date Day

(4) while (m != n)

if (m > n)
    m = m - n
else
    n = n - m

Loop to find GCD of two num.

m=16	n=2
14	2
12	2
10	2
8	2
6	2
4	2
2	2

7 times means half $n/2$

$O(n)$

(5) Algorithm test (n) {
    if (n > 5)
        printf(—);
    else
        for (i=1 ; i<n ; i++)
            printf(—);
}

Best Case → $O(1)$

Worst Case → $O(n)$

Date ____________ Day ____________

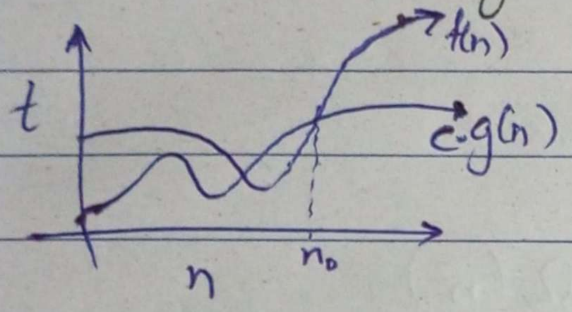
"BIG OMEGA"

→ Describes the $\boxed{\text{lower bound}}$ of an algo.

→ Used to express the best case running time of an algo

$$f(n) \geq c \cdot g(n) \longrightarrow f(n) = \Omega(g(n))$$

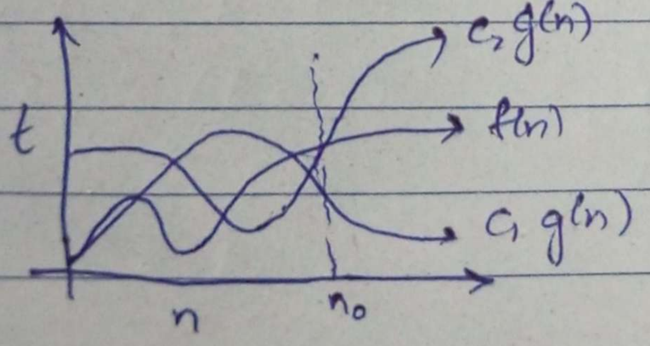
↳ g(n) is the asymptotic tight lower bound of f(n)



"BIG THEETA"

→ Describes Average bound.

$$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \Rightarrow f(n) = \Theta(g(n))$$



→ g(n) is the asymptotic tight bound of f(n)

$10n^2+5n$ $\quad$ $11n^2$ $\quad$ $10n^2+5n \le 11n^2$

15 $\quad$ 11 $\quad$ 105 $\quad$ 99 $\quad$ $5n \le n^2$

50 $\quad$ 44 $\quad$ $5 \le n$

Date ______ Day ______

Q: Find the lower bound: $n>$ $\quad$ ~~fnd~~ $10n^2+5n = O(g(n))$

$= c\, g(n)$

1. $f(n) = 10n^2 + 5n$ $\quad$ $\cancel{11}n^2$ $\quad$ n^3 $\quad$ $n!$ $\quad$ 2^n $\quad$ $10n^2+5n \le 11n^2$

Dominant term: $10n^2$ $\quad$ $5n \le n^2$

Let $g(n) = n^2$ (eliminate the constant) $\quad$ $5 \le n$

$c = 10$

n	$10n^2+5$	$10n^2$
1	15	10
2	45	40

$n_0 = 1$

$f(n) = \Omega(n^2)$

Show that $f(n) = n^3 + 3n^2 = \Theta(n^3)$

1. $f(n) \le c_2 \cdot g(n)$ $\quad$ $n^3+3n^2 \le 4n^3$

Dominant: n^3

let $g(n) = n^3$

Take, $c_2 = 2$

n	n^3+3n^2	$2n^3$
1	3	2
2	20	16
3	$127+3$	127×2

$n_0 = 3$

Date ____ Day ____

2. $f(n) \geq c_1 \cdot g(n)$

Do, n^3

let $g(n) = n^3$

$c_1 = 1$

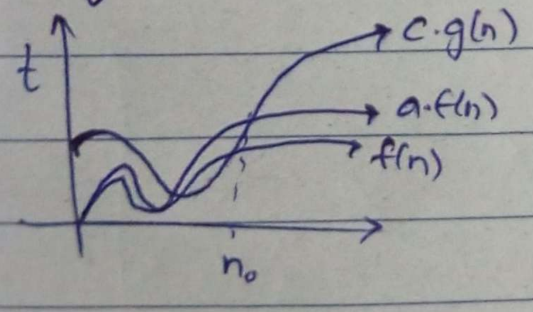
This is true for all $n > 1$

So. $f(n) = \Omega(n^3)$.

" PROPERTIES OF ASYMTOTIC NOTATIONS "

① General Property:

If $f(n) = O(g(n))$, then $a * f(n) = O(g(n))$;



Ex: $f(n) = 3n^2 + 9$ & $g(n) = n^2$. Is $f(n) = O(g(n))$?

let $c = 4$

$\therefore 3n^2 + 9 \leq 4n^2$

$9 \leq 4n^2 - 3n^2$

$9 \leq n^2$

$\boxed{3 \leq n}$

(we will not consider -ve value as 'n' cannot be -ve)

Date ______ Day

Now multiplying f(n) by 1000

$1000(3n^2 + 9) = O(g(n^2))$

$3000n^2 + 9000 = O(g(n^2))$

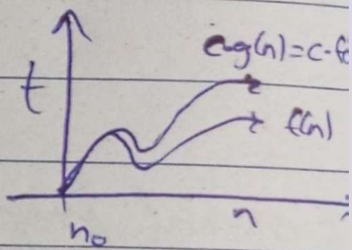
$3000n^2 + 9000 \leq 3001n^2$

$\therefore$ Constant multiplication doesnot affect.

→ This is also true for Ω, Θ.

(2) Reflexive Property:

If f(n) is given, then $f(n) = O(f(n))$



Let $f(n) = 3n$.

$g(n) = n$

$C = 6$

$1 \times 3n \leq 6n$

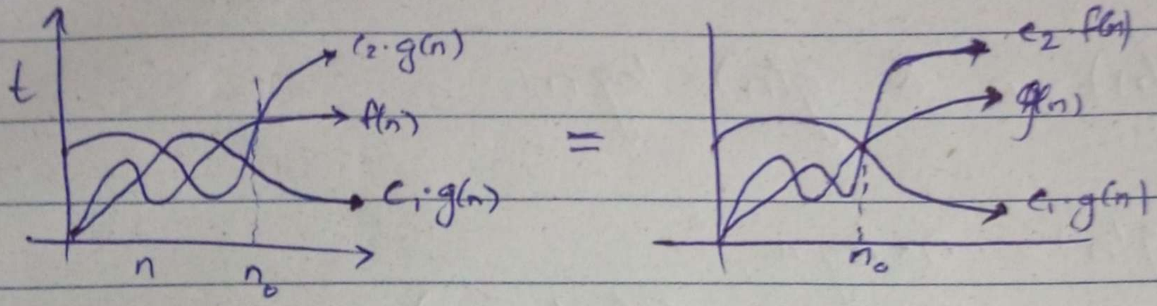
$0 \leq \cancel{6n}\ 3n$

$\boxed{0 \leq n}$

Date ______ Day ______

③ Symmetric Property:

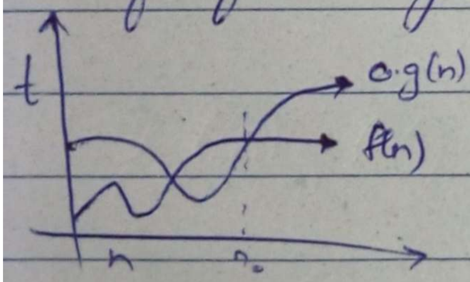
If $f(n) = \Theta(g(n))$, then $g(n) = \Theta(f(n))$



→ This property is not applicable on O and Ω

④ Transpose Symmetric Property:

If $f(n) = O(g(n))$, then $g(n) = \Omega(f(n))$;



$f(n) = n^2$, $g(n) = n^3$

$n^2 \leq n^3$ ✓ $\qquad n^3 \geq n^2$

⑤ Transitive Property:

If $f(n) = O(g(n))$, and $g(n) = O(h(n))$ then $f(n) = O(h(n))$

⑥ Addition Property:

$n \qquad n^2$ ✓

$f(n) + g(n) = O(\max(f(n), g(n)))$.

Let $f(n) = n$ and $g(n) = n^2$

$f(n) + g(n) = n + n^2 = O(n^2)$ also valid for Ω, Θ

Date

(7) Multiplication Property:

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

also valid for Ω & Θ.

Let $f(n) = n$ & $g(n) = \log n$

$f(n) * g(n) = n * \log n$

$= n\log n = O(n\log n)$

"Little O NOTATION"

$f(n) = o(g(n))$ iff $f(n) < c \cdot g(n)$

CamScanner