CS & IT ENGINEERING

Algorithms

Analysis of Algorithms

Lecture No.- 08

By- Aditya sir

Recap of Previous Lecture

Topic Properties

Topic Practice Questions

Topic PYQs

Topics to be Covered

Covered

Topic

Imp Questions

Topic

framework for determining Non-Recursive Algo Time Complexity

Topic

$\cancel{n} < \cancel{n}\log n$.

$1 < \log n$

Asymptotic Comparison of 2 functions

Ans:- A, B

MSQ

A → 85%

B → 69%

(Q1) $f(n) = n$

$g(n) = n\log n$ $\Longrightarrow$ $\boxed{f < g}$

$f \le g$

✓ A) $f = O(g)$

$f < g$ ✓ B) $f = o(g)$

C) $f = \Omega(g) \longrightarrow f \geq g$

D) $f = \omega(g) \longrightarrow f > g$

Q2) $f(n) = n^2(\log n)^1$

$g(n) = n(\log n)^{10}$

<u>MSQ</u>

B→75%

C→72%

A) $f = O(g)$

✓ B) $f = \Omega(g)$

✓ C) $f = \omega(g)$

D) $f = o(g)$

Soln :-

$$f \qquad\qquad g$$

$$n^2 \log(n) \qquad\qquad n(\log n)^{10}$$

$$n * n\cancel{\log(n)} \qquad\qquad n\cancel{(\log n)} * (\log n)^9$$

$$\underline{\underline{f > g}}$$

Poly ✓ $n > (\log n)^9 \longrightarrow \log$

V. V Imp

n_0

Given: $f(n) = n^3, \quad 0 < n < 10{,}000$
$\quad\quad\quad = n, \quad n > 10{,}000$

$g(n) = n, \quad 0 < n \leq 100$
$\quad\quad = n^3, \quad n > 100$

$\boxed{\text{Ans} \rightarrow C, D}$

$=$
$<$

X A) $f(n) = \Omega(g(n))$ for $n > 100$ → $f \underset{A}{\geq} g$

X B) $f(n) = o(g(n))$, For $n > 100$ → $f \underset{A}{\leq} g$

☆

↳ sometimes $=$ $(100 \rightarrow 10{,}000)$
sometimes $<$ $(10{,}000 \rightarrow \infty)$

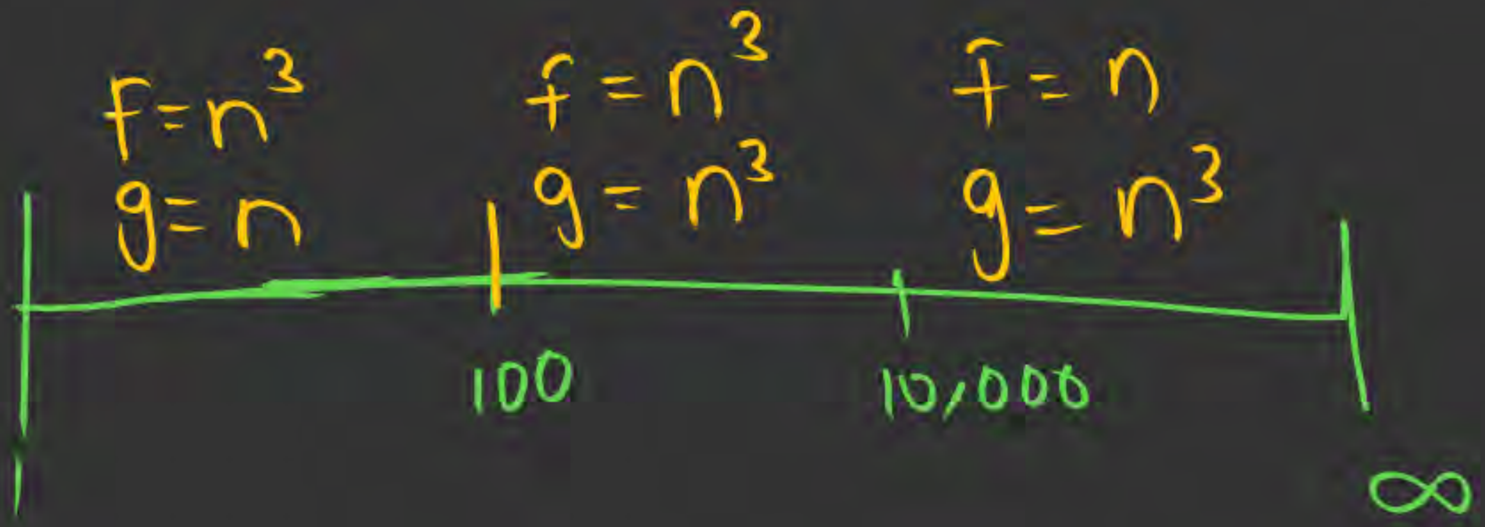
✓ C) $f(n) = O(g(n))$, For $n > 100$ → $f \underset{A}{\leq} g$

✓ D) $f(n) = o(g(n))$, for $n > 10{,}000$ → $f \underset{A}{<} g$

Soln:- Asymptotic Comparison

$$f(n) = \begin{cases} n^3, & 0 < n \le 10{,}000 \\ n, & n > 10{,}000 \end{cases}$$

$$g(n) = \begin{cases} n, & 0 < n \le 100 \\ n^3, & n > 100 \end{cases}$$



Best way

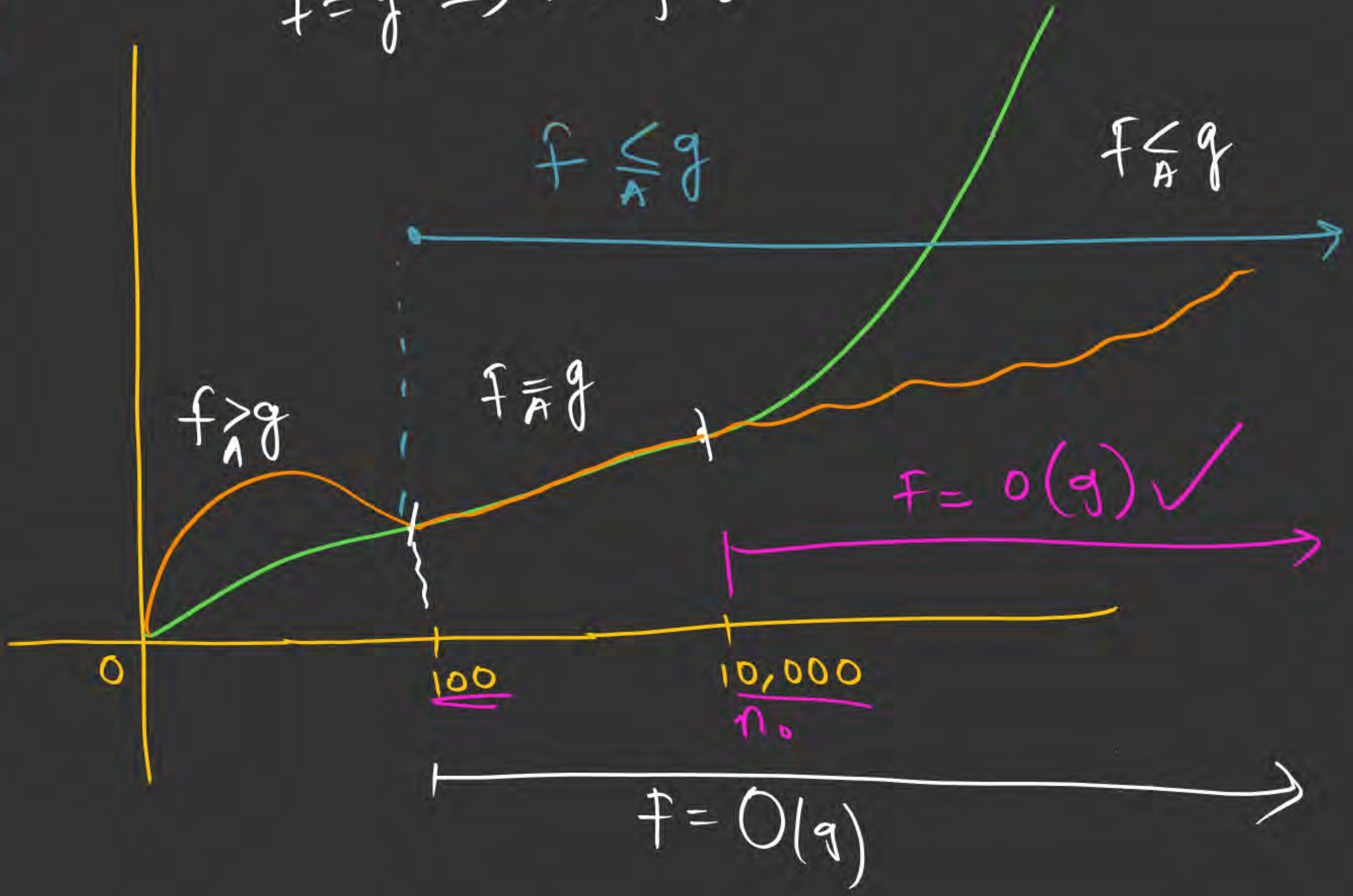
$$f(n) = \begin{cases} n^3, & 0 < n \le 100 \\ n^3, & 100 < n \le 10{,}000 \\ n, & n > 10{,}000 \end{cases}$$

$\boxed{n_0}$

$f = g \Rightarrow f \geq g$?

Orange → f

green → g



(Q) Arrange the given functions in increasing order of rate of growth:

$f_1 \rightarrow 2^n$, $f_2 \rightarrow n^{3/2}$, $f_3 \rightarrow n\log n$ $f_4 \rightarrow n^{(\log n)}$

A) f_2, f_3, f_1, f_4 ✗

B) f_3, f_2, f_1, f_4

[✓ C)] f_3, f_2, f_4, f_1

D) f_2, f_3, f_4, f_1 ✗

Soln

$f_1 \rightarrow 2^n \longrightarrow$ Expo

$f_2 \rightarrow n^{3/2} \longrightarrow$ Poly

$f_3 \rightarrow n\log n \longrightarrow$ Polylog

$f_4 \rightarrow n^{(\log n)} \longrightarrow$ Expo

$$f_2 \quad > \quad f_3$$

$$n^{3/2} \qquad n\log n$$

$$n^{(1+1/2)} \qquad n\log n$$

$$\cancel{n} * \sqrt{n} \qquad \cancel{n}\log n$$

$$\sqrt{n} > \log n$$

$$\left.\begin{array}{l} \boxed{f_2 > f_3} \\ \boxed{f_1 > f_4} \end{array}\right\} \longrightarrow \underline{f_3 < f_2 < f_4 < f_1}$$

$$2^n > n^{(\log n)}$$

Taking $\log_2()$ both sides.

$$\log_2(2^n) \qquad \log_2(n^{\log n})$$

$$n \times 1 \qquad \log n * \log n$$

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

* Imp Type:- NAT

(Q) You are given a database having 10^x records. There are 2 packages available for processing the data. Package A takes a time of $\boxed{10*n*\log n}$ while, Package B takes a time of $0.0001*n^2$ for processing `n' records. Determine the smallest integer x for which package A outperforms package 'B'.

outperforms

performs better.

↓

$(T_A < T_B)$

Ans - 6

Appr 1 :- value Substitution.

$n = 10^x$, $T_A = 10 \times n \log n$, $T_B = 0.0001 \times n^2$

Let $x = 1$, $n = 10^1$

$T_A = 10 \times 10 \times \log_{10} 10$

$= 10^2$

$T_B = 10^{-4} \times 10^2$

$= 10^{-2}$

Let $x = 2$, $n = 10^2$

$T_A = 10 \times 10^2 \times \log_{10}(10^2)$

$= 2 \times 10^3$

$= 2000$

$T_A > T_B$

$T_B = 10^{-4} * (100)^2$

$= 10^{-4} \times 10^4$

$= 1$

Let $x = 3$

$n = 10^3$

$T_A = 10 \times (10^3) * \log_{10}(10^3)$

$= 10^4 \times 3$

$T_B = 10^{-4} \times (10^3)^2$

$= 10^{-4} \times 10^6$

$= 10^2$

Appr 2 :- find x such that $T_A < T_B$

$n = 10^x$

$$10 * n * \log_{10}(n) < 10^{-4} * (n^2)$$

$$10 * 10^x * \log_{10}(10^x) < 10^{-4} * (10^x)^2$$

$$x \times 10 \times 10^x < 10^{-4} * 10^{2x}$$

$$x \times 10 \times \cancel{10^x} < 10^{-4} * \cancel{10^x} * 10^x$$

$$10x < 10^{(x-4)}$$

$$x < \frac{10^{(x-4)}}{10}$$

$$x < 10^{(x-5)}$$

$x - 4 - 1$

Shortcut

{ Solve eqn & then check x }

$x = 5$

$5 < 10^{5-5}$

$5 < 10^0$

$5 < 1$? ✗

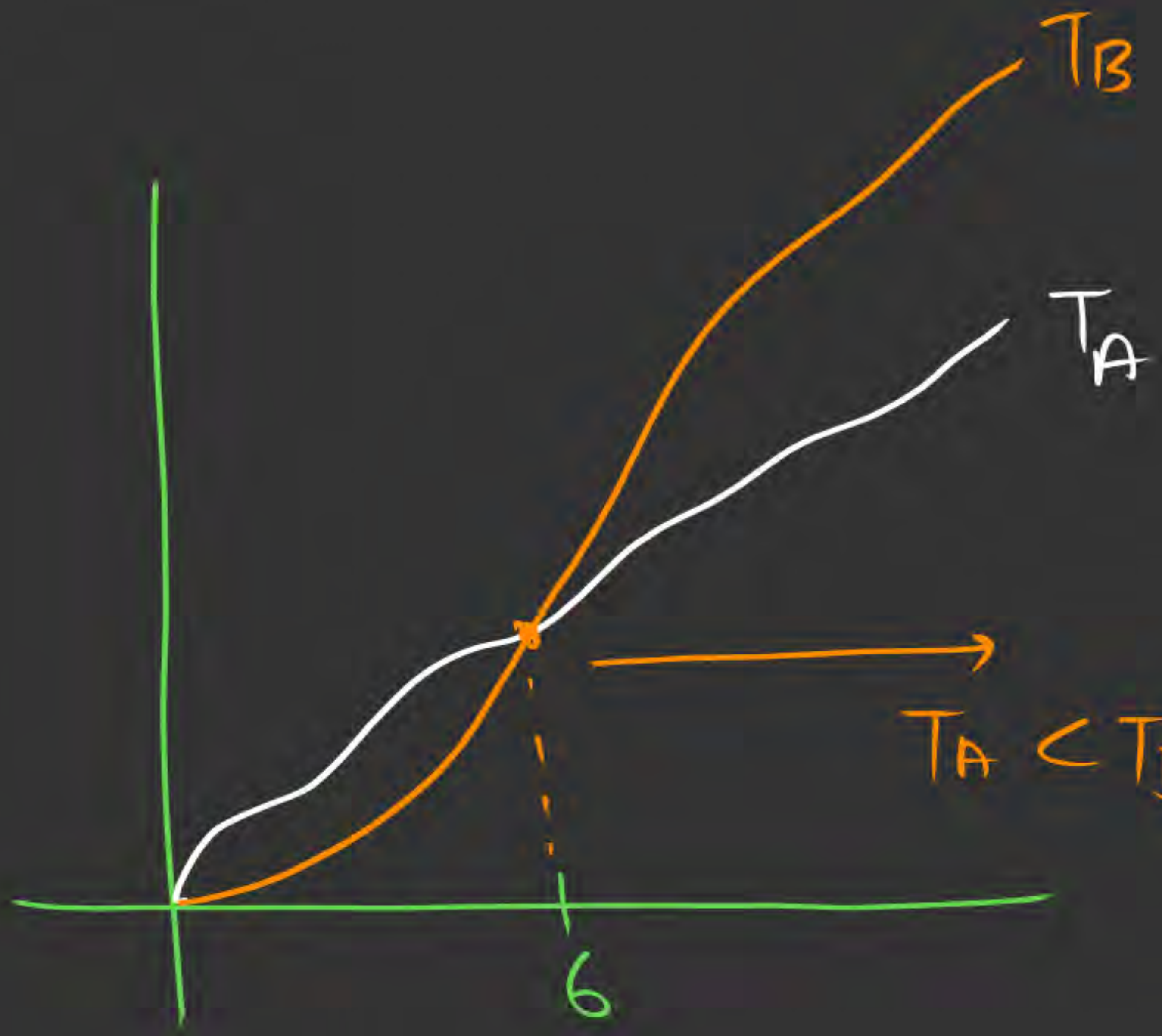
$x = \boxed{6}, 7, 8, 9 \ldots$

$(T_A < T_B)$

$x = 6$

$6 < 10^{6-5}$

$\boxed{6 < 10}$ ✓



$T_A < T_B$ (A is better than B)

[NAT]

→ V. Imp (GATE + Interviews)

#Q. An element in an Array is called Leader if it is greater than all elements to the right of it. The time complexity of the most efficient algorithm to print all Leaders of the given Array of size 'n' is _____.

Given: Last elem is not a leader

Time Complexity Analysis Of Non-Recursive Algos.

Example :-

✓	✗	✓	✗	✓	✓	✗	✓	✗
29	18	19	8	14	11	6	7	4

Leaders → 29, 19, 14, 11, 7 ✓

1 based index

Solution 1 : Brute Force

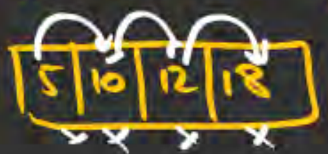
```
Algo AJleader(A[ ], n)
{ for(i=1 ; i<n ; i++) ✓
    { for(j=i+1; j≤n ;j++) ✓
        { if (A[i] < A[j]) ✓
            { break
            }
        }
        if( j==(n+1) )
            { print(A[i])
            }
    }
}
```

Should be True for Best Case

Time Complexity Analysis of Brute Force Approach.

1) **Best Case**: A is in increasing order.

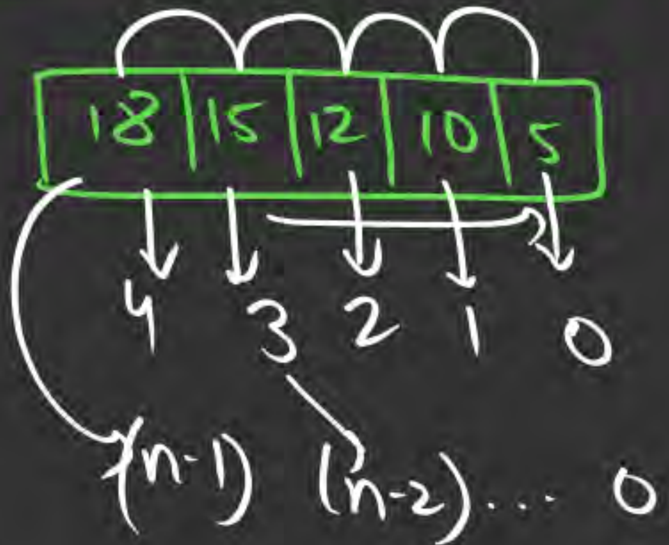
eg: A



Total Comparisons → $(n-1) * O(1) \rightarrow O(n-1)$
$\rightarrow \underline{\underline{O(n)}}$

2) **Worst Case**: A is in decreasing order.

eg: A:



Total comparisons = 4+3+2+1

In general ⇒ $(n-1)+(n-2)+(n-3)\dots 1$

$$= \left(\sum_{i=1}^{n} i\right) - n$$

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

$$= \frac{n^2-n}{2} \longrightarrow \underline{\underline{O(n^2)}}$$

Brute Force

BC → $O(n)$

WC → $O(n^2)$

$\boxed{\text{H.W}}$

2 mins Summary

Topic Imp PYQs + Practice

Topic TC analysis of Non-Recursive Algo.

THANK - YOU

Telegram Link: https://t.me/AdityaSir_PW