CS & IT ENGINEERING

Algorithms

Analysis of Algorithms

Lecture No.- 03

By- Dr. Khaleel Khan Sir

Recap of Previous Lecture

- **Topic** Types of Analysis
- **Topic** Asymptotic Notations: Big Oh, Omega and Theta
- **Topic**
- **Topic**
- **Topic**

Topics to be Covered

Topics

- Small Notations
- Properties of Asymptotic Notations

Topic : Analysis of Algorithms

Big O notation is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity.

Big O is a member of a family of notations invented by Paul Bachmann, Edmund Landau, and others, collectively called Bachmann-Landau notation or asymptotic notation. The letter O was chosen by Bachmann to stand for Ordnung, meaning the order of approximation.

In computer science, big O notation is used to classify algorithms according to how their run time or space requirements grow as the input size grows.

In analytic number theory, big O notation is used to express a bound on the difference between an arithmetical function and a better understood approximation; a famous example of such a difference is the remainder term in the prime number theorem.

Big O notation is also used in many other fields to provides similar estimates.

Big O notation characterizes functions according to their growth rates: different functions with the same asymptotic growth rate may be represented using the same O notation. The letter O is used because the growth rate of a function is also referred to as the order of the function. A description of a function in terms of big O notation usually only provides an upper bound on the growth rate of the function.

Associated with big O notation are several related notations, using the symbols O, Ω, ω and Θ, to describe other kinds of bounds on asymptotic growth rates.

Topic : Analysis of Algorithms

Definition: A theoretical measure of the execution of an algorithm, usually the time or memory needed, given the problem size n, which is usually the number of items. Informally, saying some equation f(n) = O(g(n)) means it is less than some constant multiple of g(n). The notation is read, “f of n is big oh of g of n”.

Formal Definition: f(n) = O(g(n)) means there are positive constants c and $\cancel{k}$ n_0, such that $0 < f(n) \leq cg(n)$ for all $n \geq \cancel{k}$ n_0. The values of c and $\cancel{k}$ n_0 must be fixed for the function f and must not depend on n.

Topic : Analysis of Algorithms

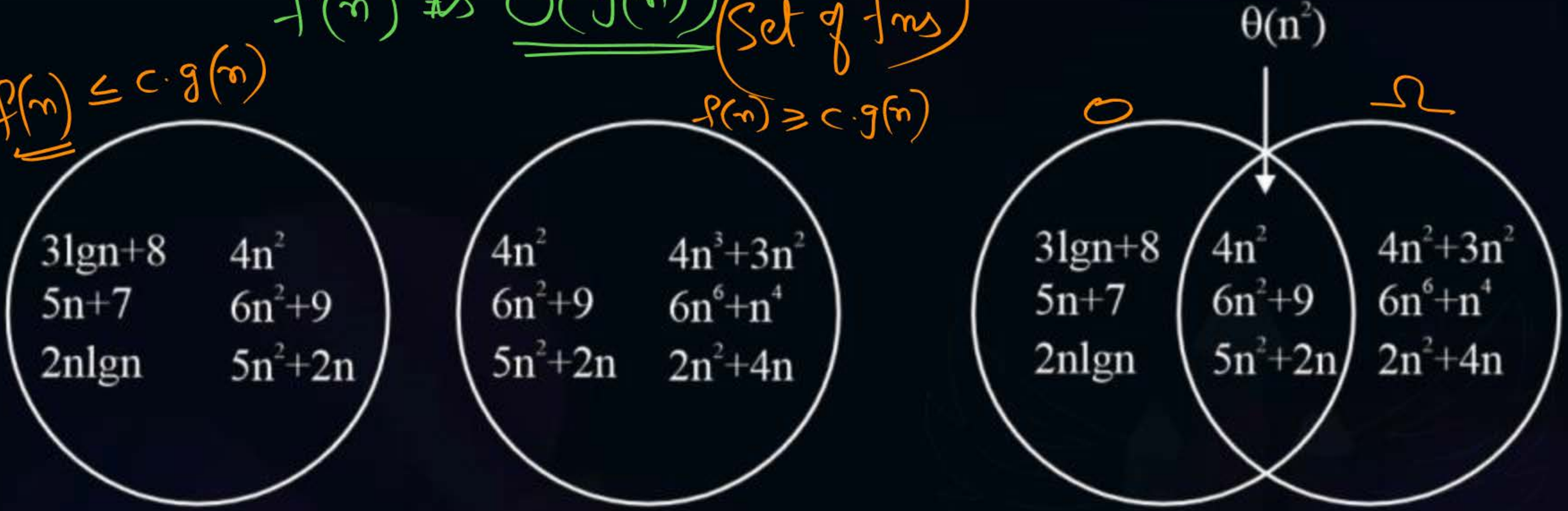
Big-Omega Notation (Ω):

Similar to big O notation, big Omega (Ω) function is used in computer science to describe the performance or complexity of an algorithm. If a running time is $\Omega(f(n))$, then for large enough n, the running time is at least k.f(n) for some constant k.

$f(n)$ is $O(g(n))$ (Set of fns)

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

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



(a) $O(n^2)$ (b) $\Omega(n^2)$ (b) $\theta(n^2) = O(n^2) \cap \Omega(n^2)$

Note:- The set $O(n^2)$, $\Omega(n^2)$, $\Theta(n^2)$. Some exemplary member are shown.

Small Notations :

The bounds Provided by Big-Notations (O, Ω) May or May Not be Tight ;

$$f(n) = n^2 \begin{cases} O(n^2) : \text{Tight} \\ O(n^3) : \dfrac{\text{Not tight}}{\text{Loose}} \end{cases}$$

However the bounds provided by Small Notations (o, ω) is always NOT tight (Loose) ⟨ Proper Bounds ⟩

4) Small-oh (o) : Proper U.B

$f(n)$ is $o(g(n))$ iff

$\boxed{f(n) < c \cdot g(n)}$, for all $c > 0$, whenever $n > n_0$, for $n_0 > 0$

5) Small omega (ω) : Proper Lower Bound

$f(n)$ is $\omega(g(n))$ iff

$\boxed{f(n) > c \cdot g(n)}$, for all $c > 0$,

whenever $n > n_0$, for $n_0 > 0$

Topic: Analysis of Algorithms

Ex:

$f(n) = n$

$O(n), O(n^2), O(n \cdot \log n)$ ✓

$o(n \cdot \log n), o(n^2), o(n^3)$ ✓

$o(n)$ ✗

$\Omega(n), \Omega(\log n), \Omega(1)$ ✓

$\omega(1), \omega(\log n)$ ✓

$\omega(n)$ ✗

$\Longrightarrow$ Smaller fn's are in order of Bigger fn's

$\Longrightarrow$ Larger fn's are in Omega of Smaller fn's

$f(n) = n \quad ; \quad g(n) = \log n$

(i) By taking Log's

(ii) By Substituting values

$f(n) = n^2 \; ; \; g(n) = n^3$

Topic: Analysis of Algorithms

$f(n) = n^2 - O(n^2)$

I. Any Big-oh satisfaction implies also small-oh satisfaction; (F)

II. Any small-oh satisfaction also implies Big-oh satisfaction (T)

$$f(n) = n^2 \begin{cases} o(n^3) \\ O(n^3) \end{cases}$$

Topic: Analysis of Algorithms

Analogy b/w Real No's & Asymptotic Notations

Let a, b : real No's ; f, g : fn's

I. $f(n) = O(g(n)) \Longleftrightarrow a \le b$

$\langle f(n) \le c \cdot g(n) \rangle$

II. $f(n) = \Omega(g(n)) \Longleftrightarrow a \ge b$

III. $f(n) = \Theta(g(n)) \Longleftrightarrow a = b$

IV. $f(n) = o(g(n)) \Longleftrightarrow a < b$

V. $f(n) = \omega(g(n)) \Longleftrightarrow a > b$

Topic : General Properties of Big Oh Notation

Let $d(n)$, $e(n)$, $f(n)$, and $g(n)$ be functions mapping nonnegative integers to nonnegative reals. Then

$a * d(n)$

$f(n) = n - O(n)$

$= 100 \cdot n - O(n)$

1. If $d(n)$ is $O(f(n))$, then $ad(n)$ is $O(f(n))$, for any constant $a > 0$.
2. If $d(n)$ is $O(f(n))$ and $e(n)$ is $O(g(n))$, then $d(n) + e(n)$ is $O(f(n) + g(n))$. $- O(\text{Max}(f(n), g(n)))$
3. If $d(n)$ is $O(f(n))$ and $e(n)$ is $O(g(n))$, then $d(n)e(n)$ is $O(f(n)g(n))$.
4. If $d(n)$ is $O(f(n))$ and $f(n)$ is $O(g(n))$, then $d(n)$ is $O(g(n))$. : Transitive
5. If $f(n)$ is a polynomial of degree d (that is, $f(n) = a_0 + a_1 n + \ldots + a_d n^d$) then $f(n)$ is $O(n^d)$.
6. n^x is $O(a^n)$ for any fixed $x > 0$ and $a > 1$. [Poly = O(Expo)] $(n+c)^m = O(n^m)$
7. $\log n^x$ is $O(\log n)$ for any fixed $x > 0$.
8. $\log^x n$ is $O(n^y)$ for any fixed constants $x > 0$ and $y > 0$. $n^2 = O(2^n)$

$(\log n)^x = O(n^y)$

$\log n^x = x \cdot \log n = O(\log n)$

$d(n) = n^2$; $e(n) = n^3$

$\downarrow$ $\quad\quad\quad\quad$ $\downarrow$

$O(\underset{f}{\underline{\underline{n^2}}})$ $\quad\quad\quad\quad$ $O(\underset{g}{\underline{\underline{n^3}}})$

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

$d(n) * e(n)$

$n^2 * n^3 = O(n^5)$

Topic: Analysis of Algorithms

$n^2 \underset{A}{\leq} 2^n$

$n=2$: $4 \quad 4$ $\quad$ $\boxed{n=3}$: $9 \quad 8$

Dominance Relation :

$$\boxed{\text{Constants} < \text{Logarithmic} < \text{Polynomial} < \text{Exponential}}$$

$$2^{100} < \log n < n < n\log n < n^2 < n^3 \ldots < 2^n < 4^n < n^n \ldots$$

$(\log \log n)$

$\sqrt{n}$

$$\boxed{n! \underset{A}{<} n^n}$$

$\underline{n > n_0}$

$$\left[\left(n(n-1)(n-2)\cdots 1\right) < \left(n \cdot n \cdot n \cdots n\right)\right]$$

$n! = o(n^n)$

$= O(n^n)$ ✓

Topic: Analysis of Algorithms

Proof for $n! < n^n$

Stirling's Approximation

$$n! \sim \sqrt{2\pi n} * \left(\frac{n}{e}\right)^n \quad < \quad \text{vs} \quad n^n$$

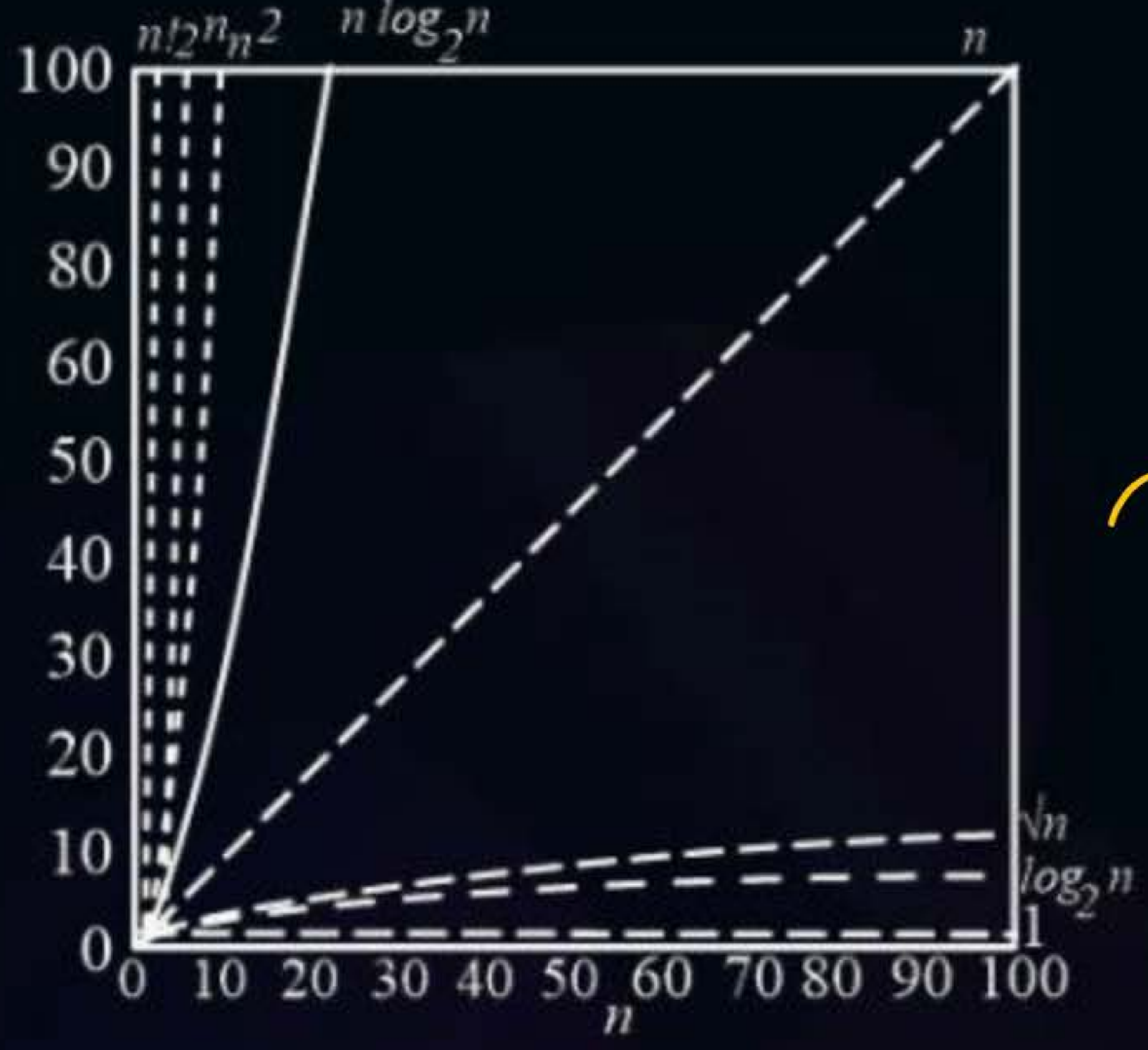
$$\sqrt{2\pi} \cdot \sqrt{n} \cdot \frac{\cancel{n^n}}{e^n} \quad \text{vs} \quad \cancel{n^n}$$

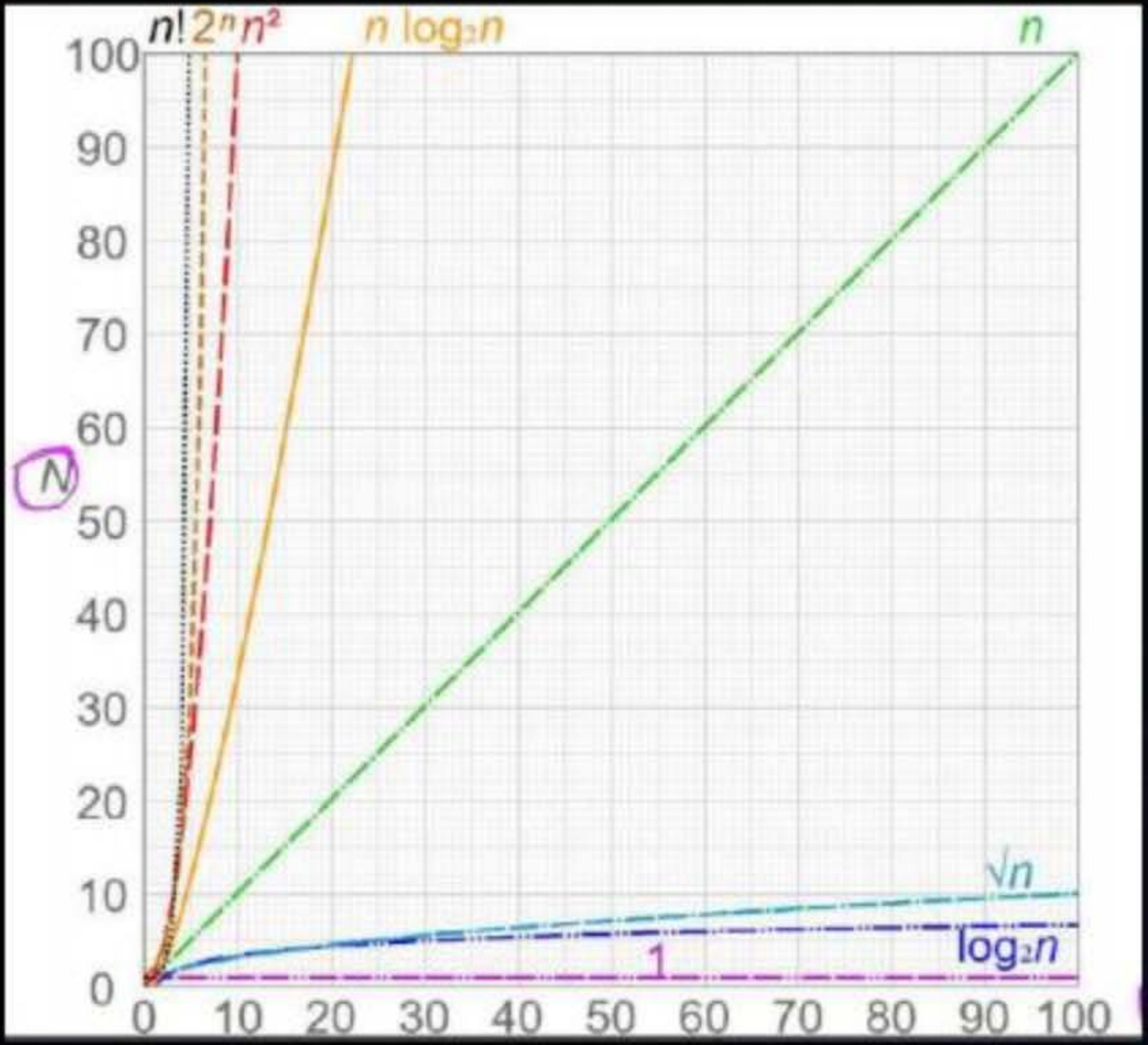
$$c \cdot \sqrt{n} \cdot \quad \text{vs} \quad e^n$$

$$\boxed{\sqrt{n} \quad \text{vs} \quad e^n} \qquad \sqrt{n} < e^n$$

Poly $\qquad$ Exp.

Topic: Analysis of Algorithms





Discrete Properties of ASN's

Property	O	Ω	Θ	o	ω
Reflexive	✓	✓	✓	✗	✗
Symmetric	✗	✗	✓	✗	✗
Transitive	✓	✓	✓	✓	✓
Transpose Symmetry	$f(n) = O(g(n)) \Rightarrow g(n) = \Omega(f(n))$			$f(n) = o(g(n)) \Rightarrow g(n) = \omega(f(n))$	

$$\left[\begin{array}{l} a \le b \\ \Rightarrow b \ge a \end{array}\right]$$

* **Tricotomy Property**

For any two real no's a, b

$a < b$

$a > b$

$a = b$

f, g : functions

$f < g : O, o$

$f > g : \Omega, \omega$??

$f = g : \Theta$

$\boxed{\text{Asn's does not Satisfy Tricotomy Property}}$

1) $f(n) = n$; $g(n) = \log n$

g is $O(f)$

2) $f(n) = 1/n$; $g(n) = n^2$

g is $\Omega(f)$

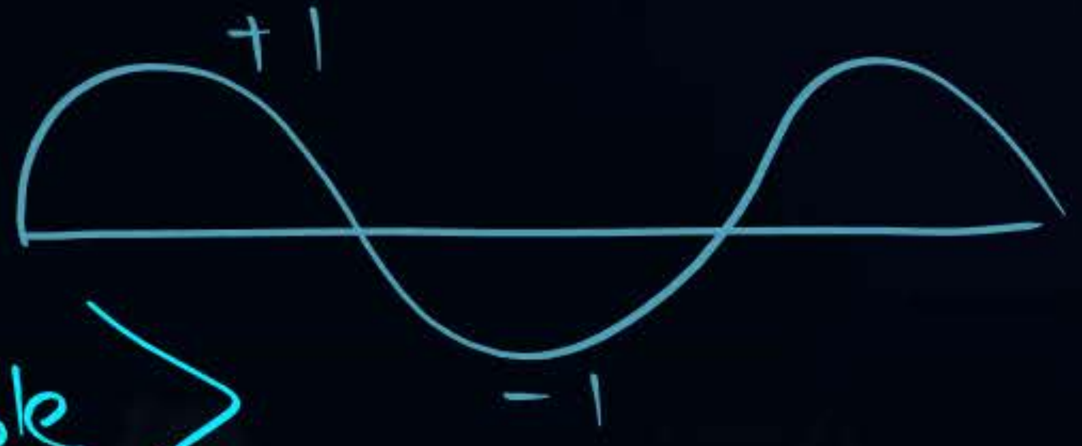
3) $f(n) = n^2$; $g(n) = 10n^2$

f is $\Theta(g)$

4) $f(n) = n$; $g(n) = n^{1+\sin n}$ *

Min : -1

Max : $+1$



$= n$ $\qquad$ $= n^{1-1} = n^{0} = 1$

$= n^{2}$

< f & g are InComparible >

Topic: Properties of Logarithms

$\log x^y = y \log x$

$\log xy = \log x + \log y$

$\underline{\log\ \log n} = \log(\log n)$

$a^{\log_b^x} = x^{\log_b^a}$

$n = 16$

$\log \log n$

$\log \log_2 16 =$

$\log_{10} n = \log_{10}^n$

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

$\log \frac{x}{y} = \log x - \log y$

$\log_b^x = \frac{\log_a^x}{\log_a^b}$

$\log \log_2 2^4 = \log_2 4 = 2$

$a = b^{\log_b a}$

$\log_c(ab) = \log_c a + \log_c b$

$\log_b a^n = n \cdot \log_b a$

$\log_b a = \frac{\log_c a}{\log_c b}$

$\log_b{}^{1/a} = -\log_b a$

$\log_b a = \frac{1}{\log_a b}$

$a^{\log_b c} = c^{\log_b a}$

Topic : Exponentials

For all real $a > 0$, m, n

$a^0 = 1$

$a^1 = a$

$a^{-1} = 1/a$

$(a^m)^n = a^{mn}$

$(a^m)^n = (a^n)^m$

$\boxed{a^m \cdot a^n = a^{m+n}}$

Topic: Summation Series

Airthmetic series

$$\sum_{k=1}^{n} k = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

Geometric series

$$\sum_{k=0}^{n} x^k = 1 + x + x^2 \dots + x^n = \frac{x^{n+1} - 1}{x - 1} (x \neq 1)$$

Harmonic series *

$$\sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \dots + \frac{1}{n} \approx \log n$$

$$\left(\sum_{x=1}^{n} 1/x \approx \int_{1}^{n} 1/x \cdot dx = [\log x]_{1}^{n} = \log n \right.$$

Topic : Geometric Sum Formula

1. The geometric sum formula for finite terms is given as:

if $r = 1$, $S_n = n*a$

if $|r| < 1$, $\boxed{S_n = \frac{a(1 - r^n)}{1 - r}}$ *

if $|r| > 1$, $\boxed{S_n = \frac{a(r^n - 1)}{r - 1}}$ *

Where

- a is the first term
- r is the common ratio
- n is the number of terms

Topic : Geometric Sum Formula

2. The geometric sum formula for infinite terms is given as:

   if $|r| < 1$, $\boxed{S_\infty = \dfrac{a}{1-r}}$

   If $|r| > 1$, the series does not converge and it has no sum.

$\log_2 n$

PYQ

$f(n) = \sum_{i=1}^{n} i^3 = x$, choices for x

I. $\theta(n^4)$ ✓ II. $\theta(n^5)$ ✗

III. $O(n^5)$ ✓ IV. $\Omega(n^3)$ ✓

TRUE

a) I, II, III ✗

b) I, II, III & IV ✗

c) I, III & IV ✓

d) I & III only

$$f(n) = \sum_{i=1}^{n} i^3 = x$$

$$x = \left[\frac{n(n+1)}{2}\right]^2 = n^4$$

$O(n^4)$ — $\Omega(n^4)$

$\Theta(n^4)$

Topic : Commonly occurring Asymptotic order

1. $f(n) = \sum_{i=1}^{n} i^d = \theta\left(n^{d+1}\right)$ (polynomial series)
2. $f(n) = \sum_{i=1}^{n} r^i = \theta(r^n) r \neq 1$ (Geometric series)
3. $f(n) = \sum_{i=1}^{n} \log i = \theta(n\log n)$ (sum of log)
4. $f(n) = \sum_{i=1}^{n} i^d \cdot \log i = \theta\left(n^{d+1} \cdot \log n\right)$ (poly-log series)
5. $\sum_{\mathrm{k}=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$ x $\neq 1$
6. $fn|x| < 1 \sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$

1) $f(n) = n^{1/\log n} = O(\quad)$

$$\frac{1}{\log_b a} = \log_a b$$

$$n = 2^{\log_2 n}$$

$$\left(2^{\cancel{\log_2} n}\right)^{1/\cancel{\log} n}$$

$$2 = O(1)$$

$$n^{1/\log n} = x$$

$$\frac{1}{\cancel{\log n}} * \cancel{\log n} = \log_2 x$$

$$\log_2 x = 1$$

$$\boxed{x = 2^1} = O(1)$$

$$n^{1/\log_2 n}$$

$$n^{\log_n 2}$$

$$n$$

$$2^{\log_n n} = 2$$

$$= O(1)$$

1) $f(n) = \sum_{i=1}^{n} 1/2^i$ = Value

$= O(\quad)$

2) $f(n) = \sum_{i=1}^{n} \log i = O(\quad)$

THANK - YOU