GALGOTIAS
UNIVERSITY

Asymptotic Notations

Session No.: 2
Course Name: Analysis and Design of Algorithms
Course Code: R1UC407B
Instructor Name: Dr. Mili Dhar

Review of the key concepts of session no

1. An algorithm is a sequence of computational steps that trans input into output.

2. Step by step instructions or flowchart or pseudocode are the represent an algorithm for a program.

3. pseudocode does not have a specific syntax like any of the programming languages and thus cannot be executed on a co

How to predict an algorithm's performanc for large inputs without actually running

At the end of this session students will be a to

1. Explain Asymptotic Notations (O, Ω, ʘ) based on Orders of Gr
2. Apply asymptotic notations to represent best case, worst average case complexity.

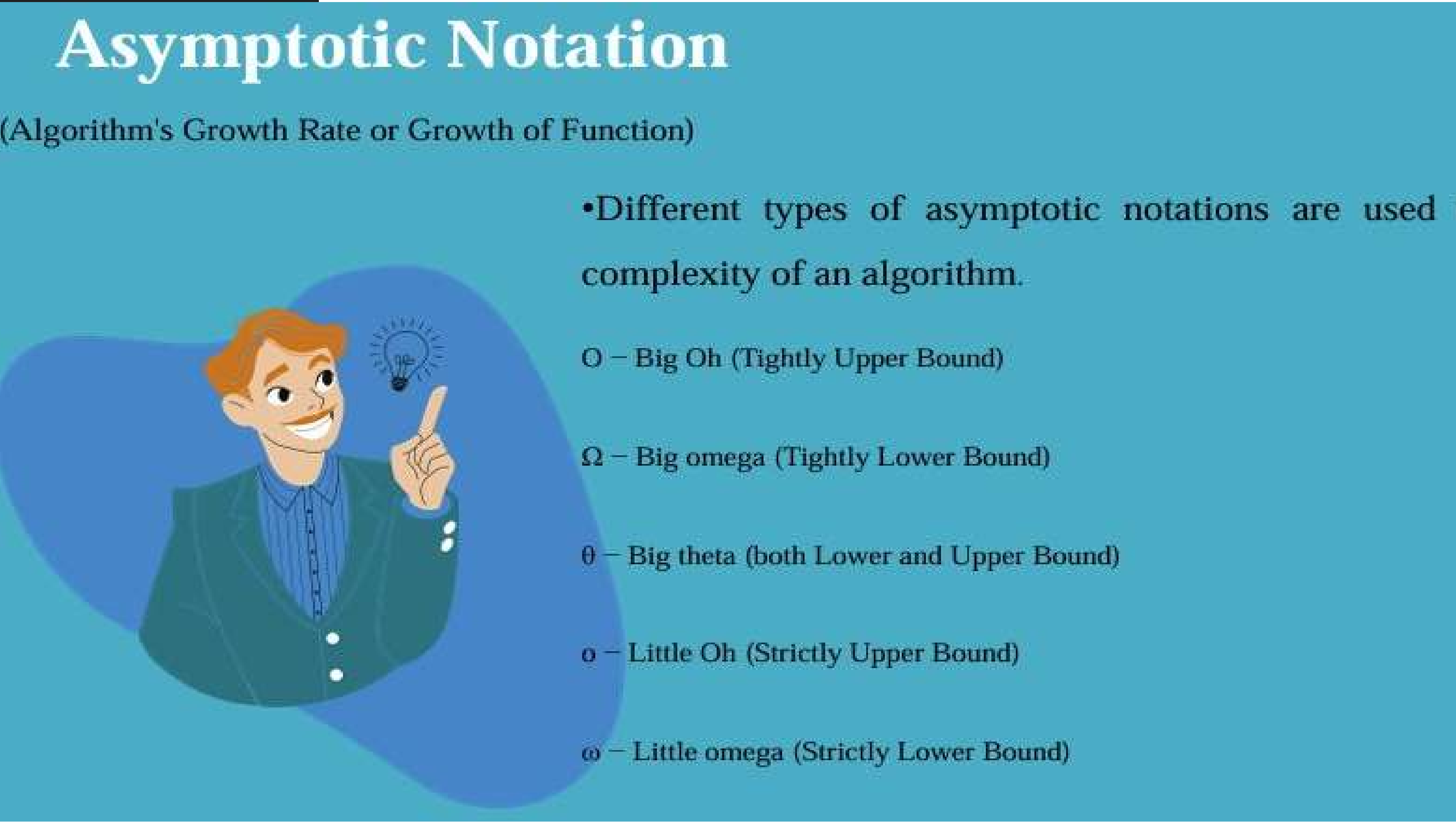
Session Outline

1. Introduction of Asymptotic Nota

2. Calculate time and space comp

3. A hands-on approach to apply Asymptotic Notations

Introduction of Asymptotic Notation



Asymptotic Notation

(Algorithm's Growth Rate or Growth of Function)

•Different types of asymptotic notations are used

complexity of an algorithm.

O – Big Oh (Tightly Upper Bound)

Ω – Big omega (Tightly Lower Bound)

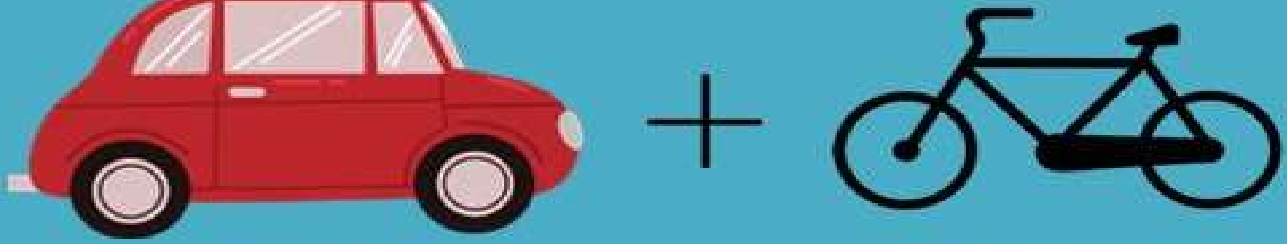
θ – Big theta (both Lower and Upper Bound)

o – Little Oh (Strictly Upper Bound)

ω – Little omega (Strictly Lower Bound)

Example of Asymptotic Analysis

(Best suitable for very large value of n or data sets)



Price: 35 Lakh

Price: 10,000/-

- The simplest example is a function f (n) = n^2+3n, the term 3n be compared to n^2 when n is very large.
- The function "f (n) is said to be asymptotically equivalent to n^2 is written symbolically as f (n) ~ n^2.

GALGOTIAS UNIVERSITY

Introduction of Asymptotic Notation

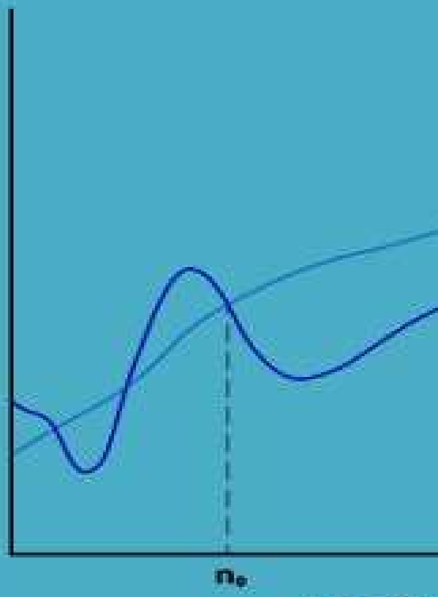
O – Big Oh Notation

- Tightly Upper Bound of an Algorithm
- Given a particular problem of size n. The time required by any algorithm for sc this problem is denoted by a function such as f(n).

Lets assume, f(n) and g(n) are two functions

f(n) = O(g(n)) if there exists a positive integer **n₀** and a positive constant **c**, such that

$$\mathbf{f(n) \leq c.g(n)\ \forall\ n \geq n_0}$$



f(n) = O(g

Introduction of Asymptotic Notation

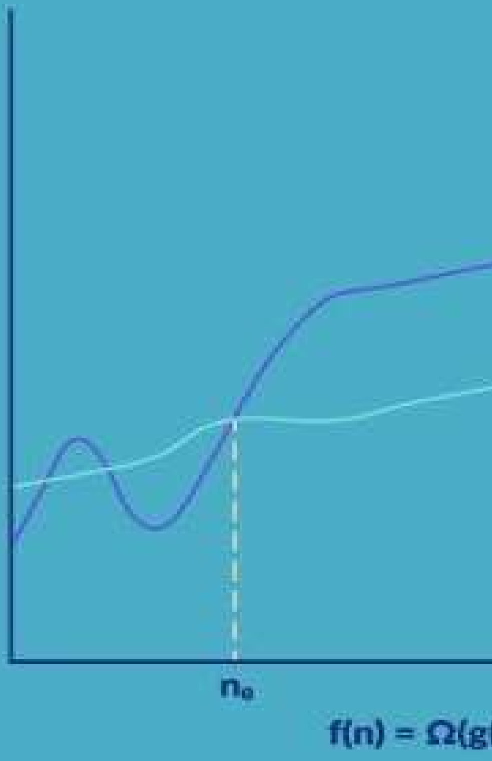
Ω – Big omega Notation

- Tightly Lower bound of the running time of an algorithm.

For a given function f(n) and g(n),

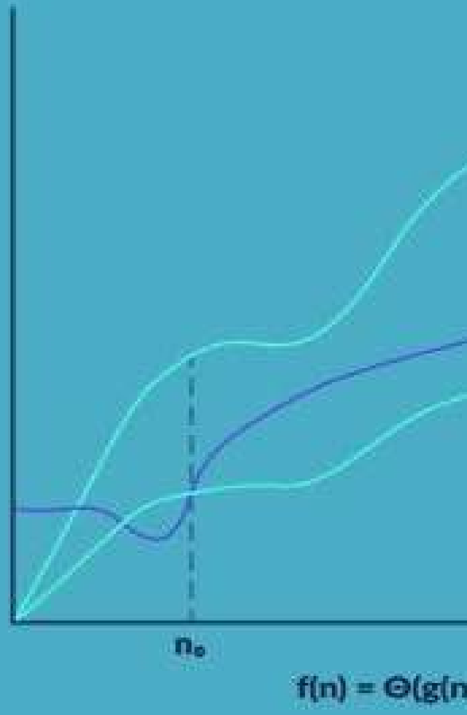
$f(n) \geq c.g(n)$, for $n \geq n_0$; $c > 0$; then

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



θ – Big theta Notation

- Represents the upper and the lower bound of the running time of an algorithm
- Used for analyzing the average-case complexity of an algorithm.



For a given function f(n) and g(n),

$$c_1 g(n) \leq f(n) \leq c_2 g(n), \text{ for } n \geq n_0 \text{ ; } c > 0 \text{ ; } n_0 \geq 1$$

The Theta Notation is more precise than both the big-oh and Omega notation. The function f (n) = θ (g (n)) if g(n) is both an upper and lower bound.

Introduction of Asymptotic Notation

o – Little Oh & ω – Little omega

Little-o Notation (Strictly Upper Bound)

f(n) = o g(n) if, f(n) < c.g(n)

for some constant n_0 and c.

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

f(n) < c.g(n) for c=2, n>=

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

Little omega Notation (Strictly Lower Bound)

f(n) = ω g(n)   if, f(n) > c.g(n)

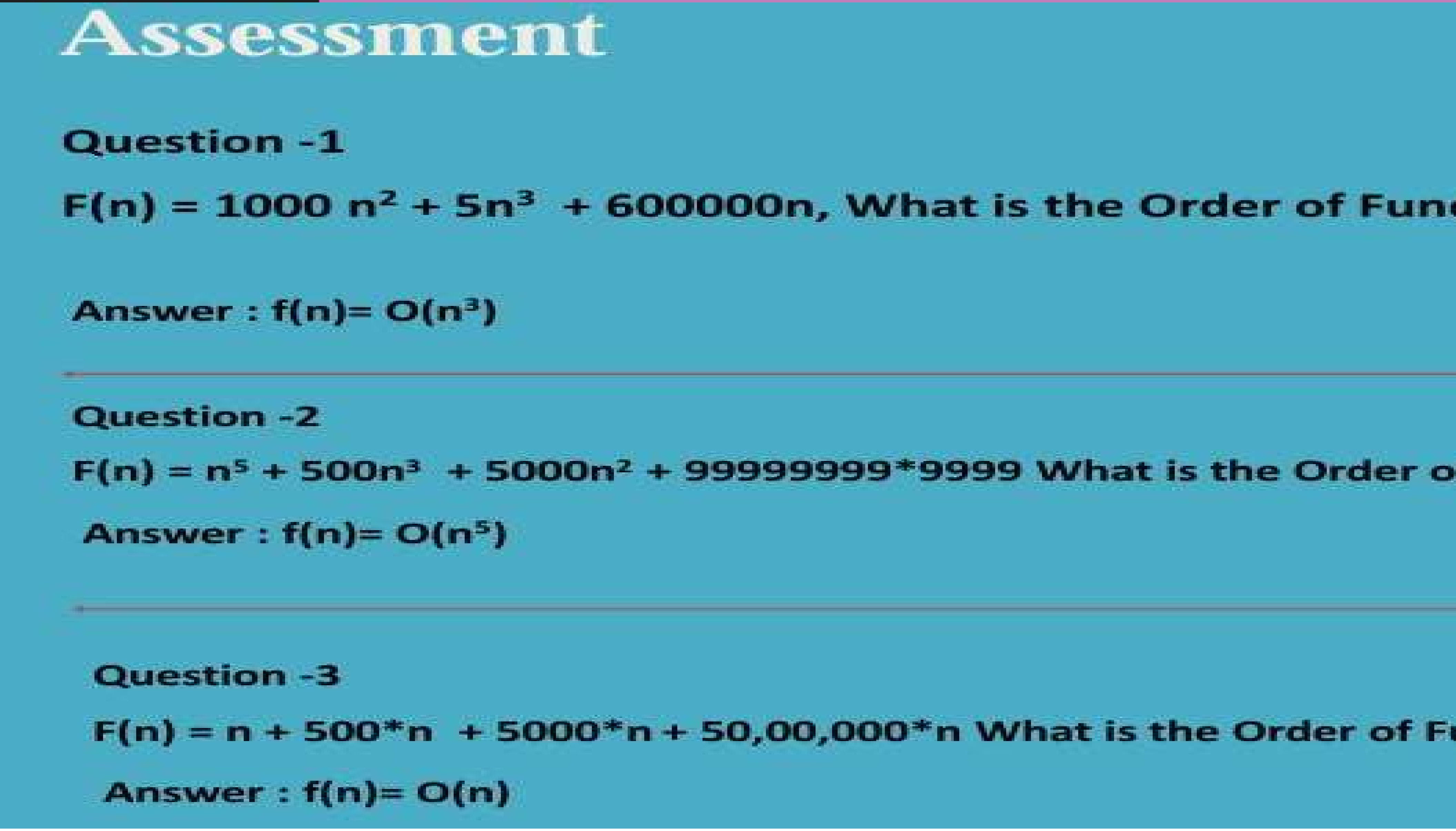
for some constant n_0 and c.

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

f(n) > c.g(n)

f(n)=ω (g(n))

Introduction of Asymptotic Notation



Assessment

Question -1

F(n) = 1000 n^2 + 5n^3 + 600000n, What is the Order of Fun

Answer : f(n)= O(n^3)

Question -2

F(n) = n^5 + 500n^3 + 5000n^2 + 99999999*9999 What is the Order o

Answer : f(n)= O(n^5)

Question -3

F(n) = n + 500*n + 5000*n + 50,00,000*n What is the Order of F

Answer : f(n)= O(n)

Learning Activity- 1

Write a pseudocode of swapping two numbers and find the tim complexity

Learning Activity-2

1. Write a pseudocode for the sum of n numbers in an array and find out time
2. Find the time complexity of linear search

Summary

Asymptotic notation describes the efficiency (time and space complexity)
as the input size grows large. It helps compare algorithms based on their gr

Common Notations:

- Big-O (O): Upper bound, worst-case complexity.
- Omega (Ω): Lower bound, best-case complexity.
- Theta (Θ): Tight bound, average-case complexity.

Next Session….

We will learn about the properties of asymptotic notations