

Understanding Time Complexity

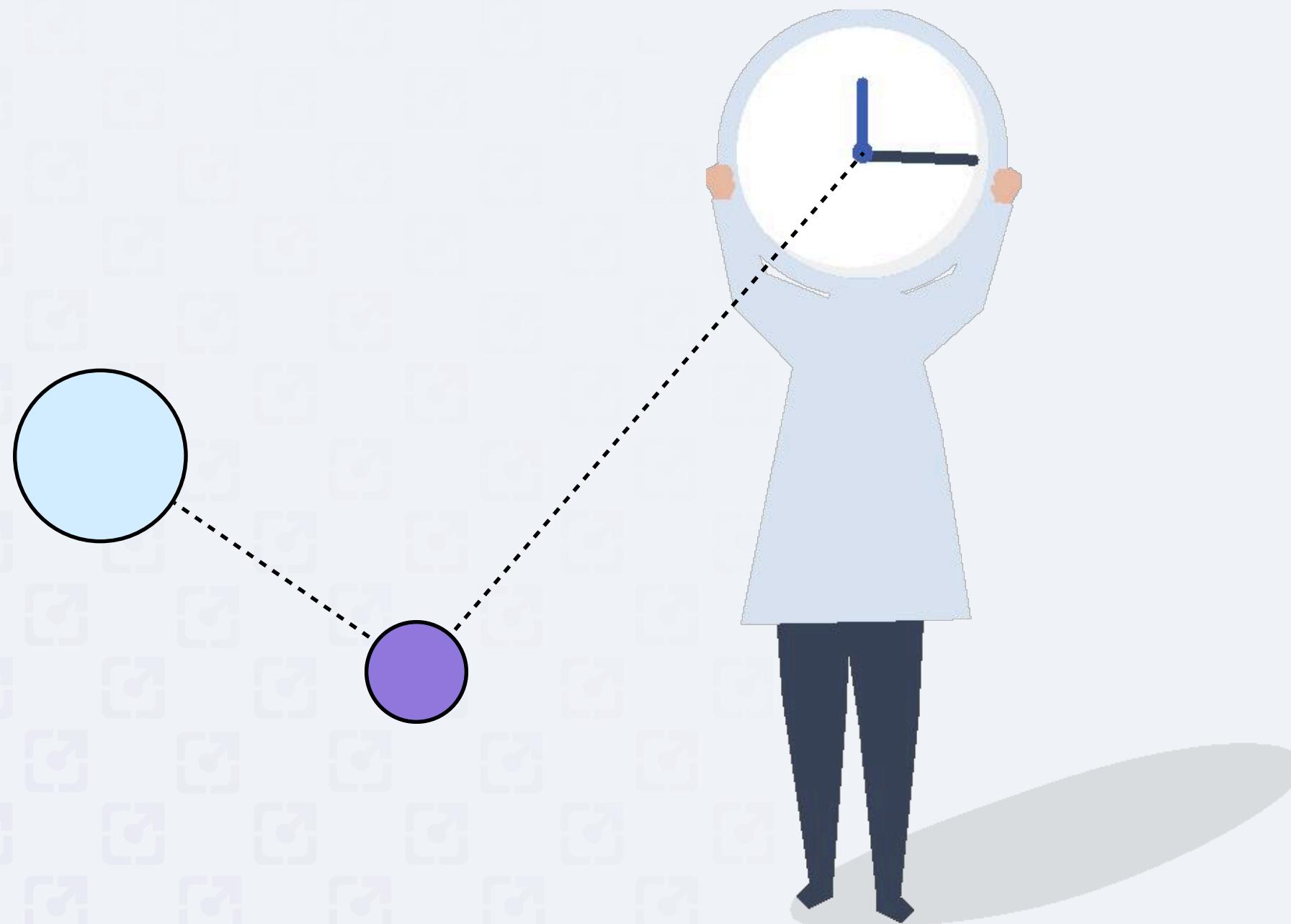
with **Real-Life**
Examples



Introduction

What is Time Complexity?

- Time complexity measures how the execution time of an algorithm grows with the input size (n).
- Why It Matters:
Understanding time complexity helps you write efficient code that scales well.



Next →

Types of Time Complexity

Common Time Complexities:

- **O(1)** - Constant Time: Execution time remains constant, regardless of input size.
- **O(log n)** - Logarithmic Time: Execution time grows slowly as input size increases.
- **O(n)** - Linear Time: Execution time grows linearly with input size.
- **O(n²)** - Quadratic Time: Execution time grows proportionally to the square of input size.
- **O(2ⁿ)** - Exponential Time: Execution time doubles with each increase in input size

Real-Life Analogy:

- **O(1): Finding the first page in a book.**
- **O(log n): Searching in a sorted phone book.**
- **O(n): Reading every page in a book**

O(1) Constant Time

Real-Life Example: Checking if the first element in a list is even.

Code Snippet

```
def is_even_first_element(arr):
    return arr[0] % 2 == 0

arr = [10, 15, 20, 25]
print(is_even_first_element(arr)) # Output: True
```

Time Complexity: O(1)

Constant time because it always checks the first element, regardless of array size

O(log n) Logarithmic Time

Real-Life Example: Binary search in a sorted list

Code Snippet

```
def binary_search(arr, target):
    left, right = 0, len(arr) - 1
    while left <= right:
        mid = (left + right) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return -1
```

Time Complexity: O(log n)

The search space is divided in half each iteration.

Example - $O(n)$ Linear Time

Real-Life Example: Finding the sum of all elements in a list.

Code Snippet

```
def sum_of_elements(arr):
    total = 0
    for num in arr:
        total += num
    return total

arr = [1, 2, 3, 4, 5]
print(sum_of_elements(arr)) # Output: 15
```

Time Complexity: $O(n)$

Each element is processed once.

Example - $O(n^2)$ Quadratic Time

Real-Life Example: Checking all pairs of a list.

Code Snippet

```
def all_pairs(arr):
    for i in range(len(arr)):
        for j in range(len(arr)):
            print(arr[i], arr[j])

arr = [1, 2, 3]
all_pairs(arr)
```

Time Complexity: $O(n^2)$

For every element, the entire list is traversed again.

Top 5 Sorting Algorithms

Algorithm	Best Time	Average Time	Worst Time	Space Complexity
Quick Sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$	$O(n \log n)$
Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$
Heap Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$	$O(1)$
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$

Pro Tip : Choose sorting algorithms based on the problem constraints and input size

Why Time Complexity Matters

Benefits of Knowing Time Complexity:

- Write efficient and optimized code.
- Solve real-world problems that require scalability.
- Perform better in technical interviews by choosing the right algorithm

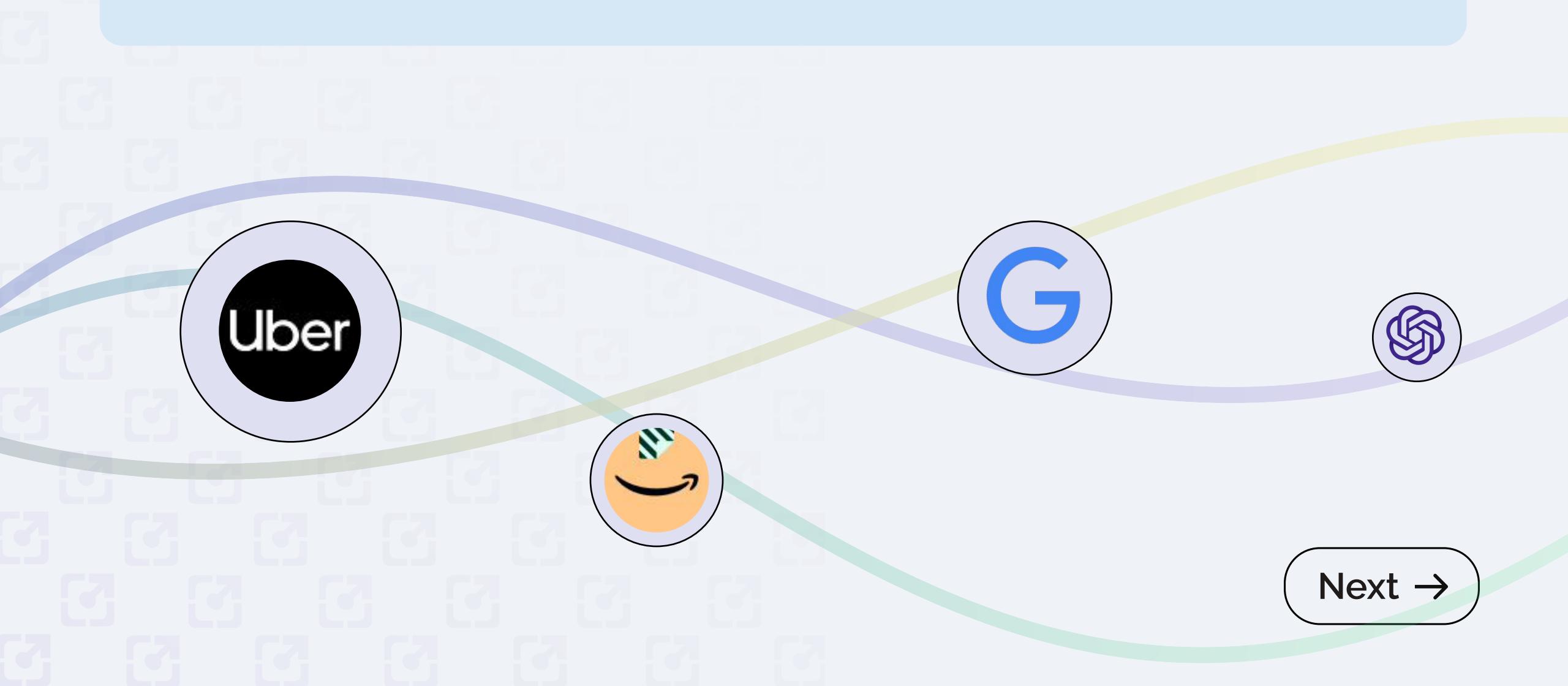


Next →

Real-Life Systems Using Efficient Algorithms

- Google Search: Implements logarithmic and linear search concepts for efficiency.
- E-commerce Platforms: Use sorting and searching to display products.
- Ride-Sharing Apps: Apply Dijkstra's algorithm for shortest route calculations.

Efficiency Drives Scalability: Understanding time complexity ensures systems remain fast and reliable.



Master Time Complexity

- Focus on solving problems with varying time complexities.
- Practice real-life examples on platforms like LeetCode and HackerRank.
- Learn how to optimize algorithms for better performance.

Start solving today and level up your coding skills for interviews and real-world challenges!

Upskill with



India's most trusted

Program For **Working Professional**

Data Structure Algorithms & System Design

With **Gen-AI** For Software Developers

Program electives:



GenAI



Product management



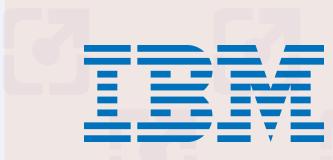
DevOps



Data engineering



Get Certification from :



IWOOLF UNIVERSITY/

