📘 What Is Time Complexity?

- 📦 Why Does It Matter?
- ⏰ Big-O Notation
- 🧪 Example: Linear vs. Quadratic
- 🧭 Key Takeaways

Time complexity is a way to describe **how the running time of an algorithm increases** as the size of the input grows.

Think of it as a **mathematical estimate** of how much time your code will take to run, especially when dealing with **large amounts of data**.

📦 Why Does It Matter?

Imagine sorting a list of 10 names vs. 10 million names. You want to know:

- Will your algorithm still be fast?
- Will it slow down a little or a lot?

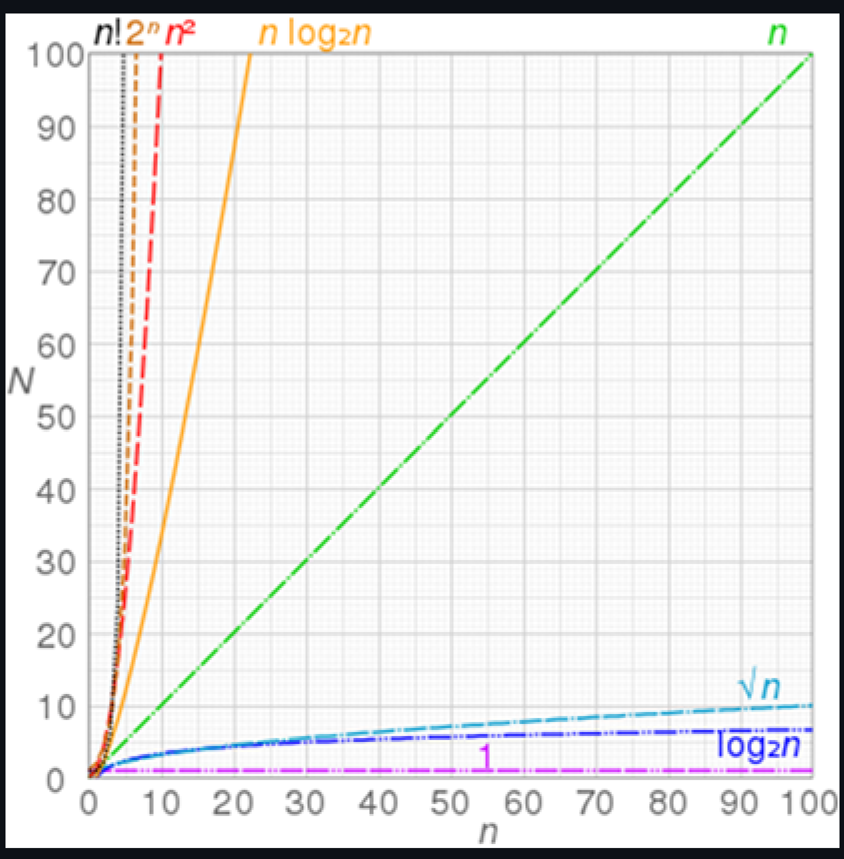
Time complexity helps you **predict performance** without actually running the program.

⏰ Big-O Notation

We use **Big-O notation** to express time complexity. It describes the **upper bound** of an algorithm's running time.

Here are some common time complexities:

Big-O Notation	Name	Example Scenario
O(1)	Constant time	Accessing an array element
O(log n)	Logarithmic time	Binary search
O(n)	Linear time	Looping through an array
O(n log n)	Log-linear time	Merge sort, Quick sort (average)
O(n^2)	Quadratic time	Nested loops (e.g., bubble sort)
O(2^n)	Exponential time	Recursive algorithms (e.g., brute force)



🧪 Example: Linear vs. Quadratic

Let's say you have a list of `n` numbers:

- A **linear algorithm** (like finding the max) checks each number once → (O(n))
- A **quadratic algorithm** (like comparing every pair) does (n \times n) comparisons → (O(n^2))

So if `n = 1,000`:

- Linear: ~1,000 steps
- Quadratic: ~1,000,000 steps 😳

🧭 Key Takeaways

- Time complexity helps you **compare algorithms**.
- It focuses on **growth rate**, not exact time.
- **Lower time complexity = better scalability**.

Footer Separator

🔭 Markdown Viewer

How to view the markdown files in a browser...

- Markdown Viewer

🧠 Lecture Practices

Here are the lecture Practices...

- Day 4
- Day 5
- Day 6

🔍 Lecture Quizzes

Here are the lecture quizzes...

- Day 4
- Day 5
- Day 6

Weekly Topics

Here are the topics for the week...

- Recursion
- Pseudocode
- Sorting
- Searching
- Maps
- Time Complexity