Module - 1

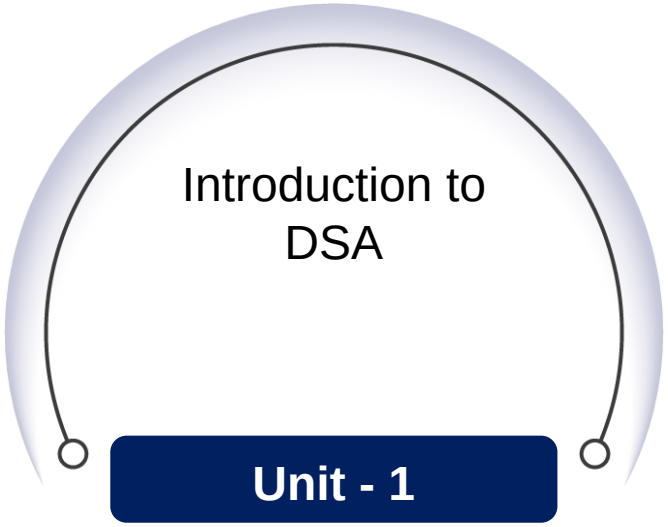
Introduction to Python with Data Structure

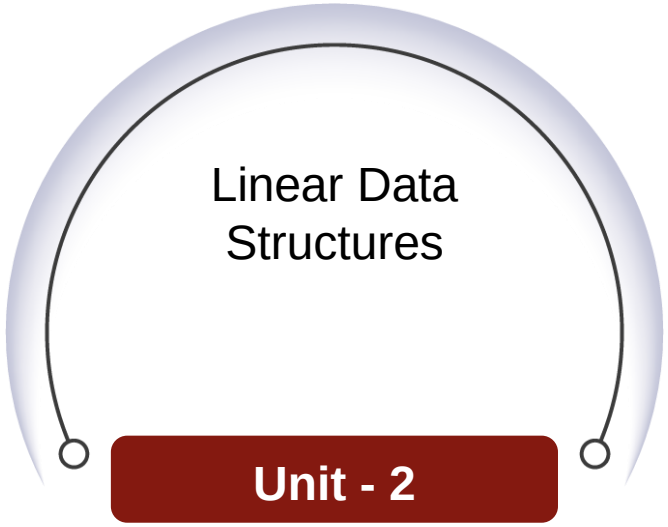
SAP

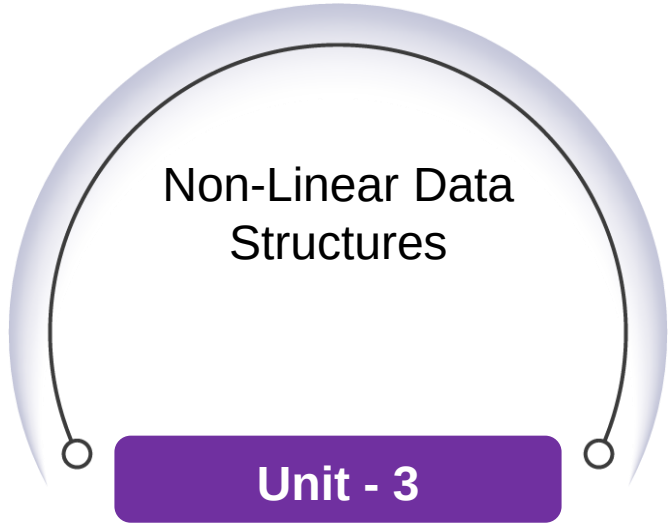
edunet foundation

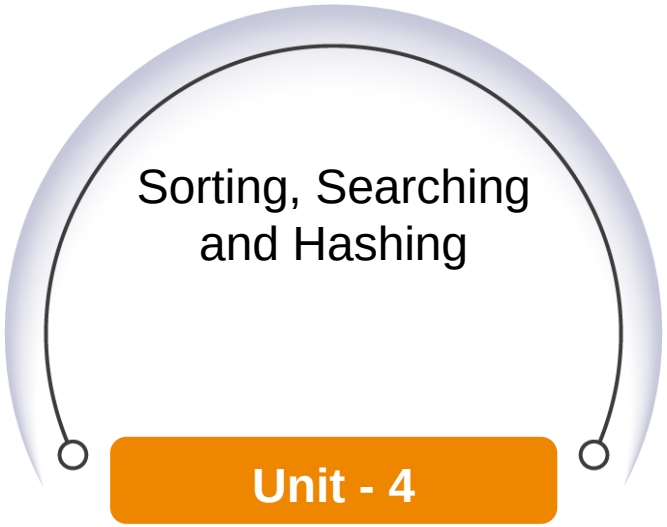
code unnati

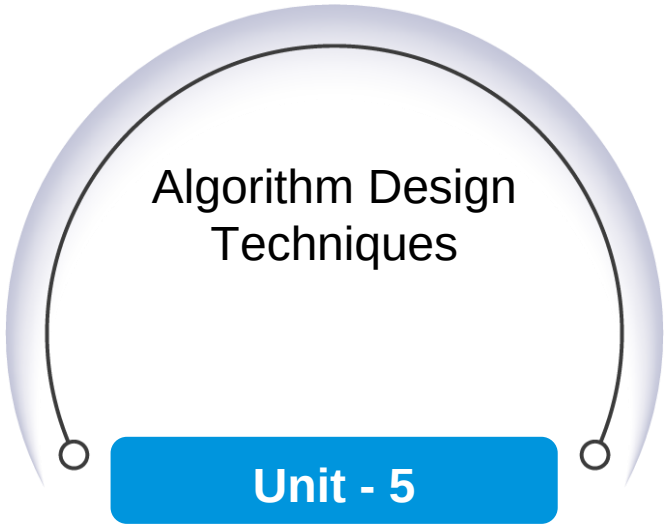
Units for Discussion











Unit - 1

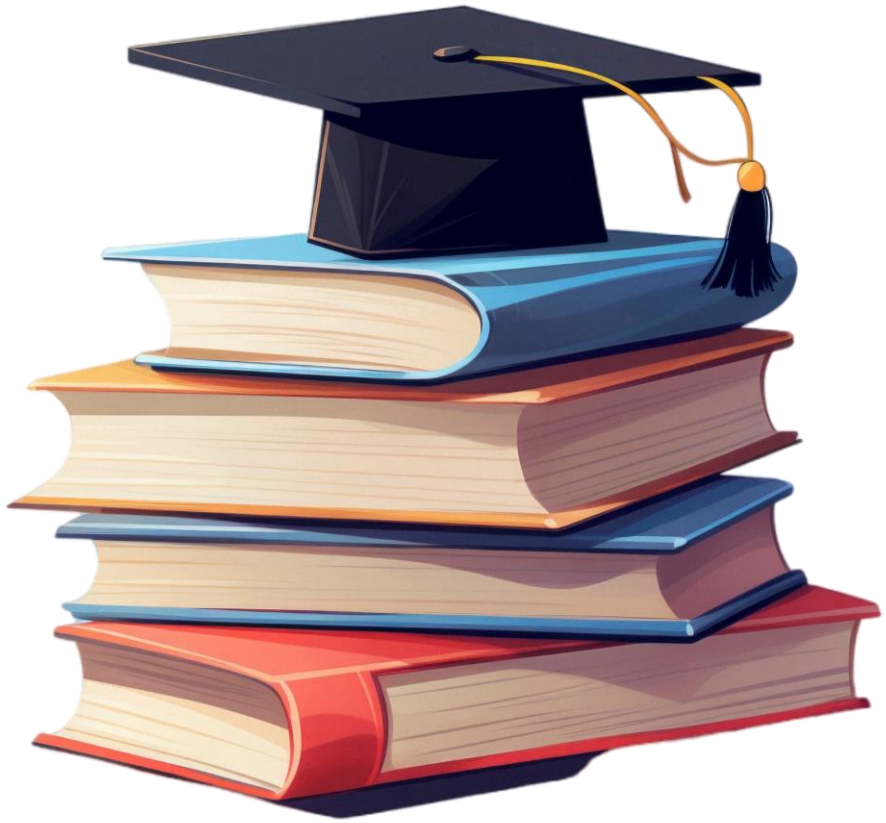
Introduction to DSA

DISCLAIMER

The content is curated from online/offline resources and used for educational purpose only.

Learning Objectives

- Introduction
- Overview of Data Structures and Algorithms
- Importance of Efficient Data Organization and Algorithm Designs
- Linear vs Non-linear data structure
- Introduction to Asymptotic analysis
- Basics of Algorithm Analysis and Complexity Theory
- Time and Space Complexity Analysis
- Introduction to Asymptotic Notation
- Summary



Source :

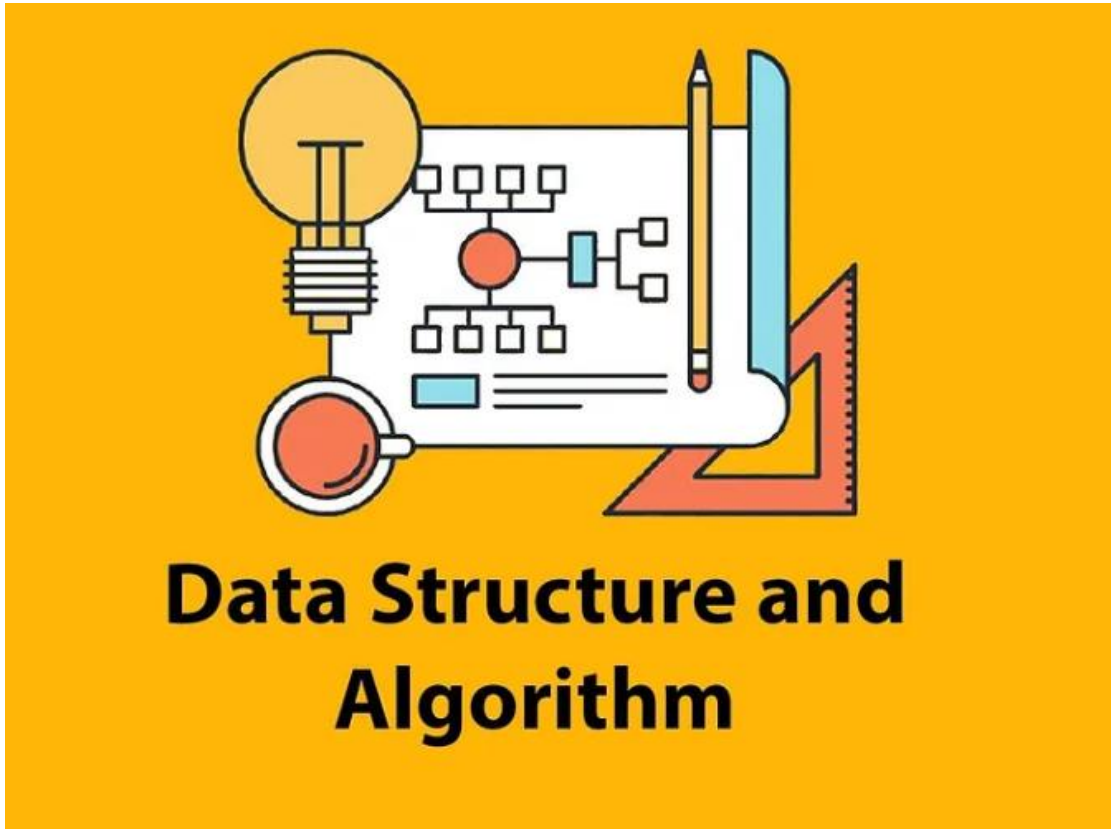
Reference

Overview of Data Structures and Algorithms

Definition of Data Structures:
- Ways of organizing and storing data to enable efficient access and modification.

Definition of Algorithms:
- Step-by-step procedures or formulas for solving problems.

Examples:
- Data Structures: Arrays, Linked Lists, Trees, Graphs
- Algorithms: Sorting & Searching



Source :

Reference

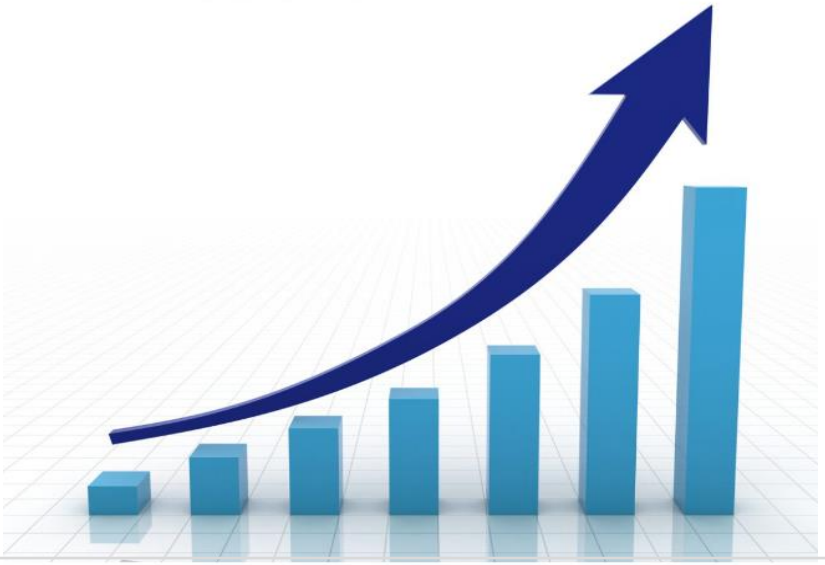
Importance of Efficient Data Organization and Algorithm Designs

Performance:
- Faster execution times and improved responsiveness.

Scalability:
- Ability to handle larger datasets and more complex operations.

Resource Management:
- Optimized use of memory and processing power.

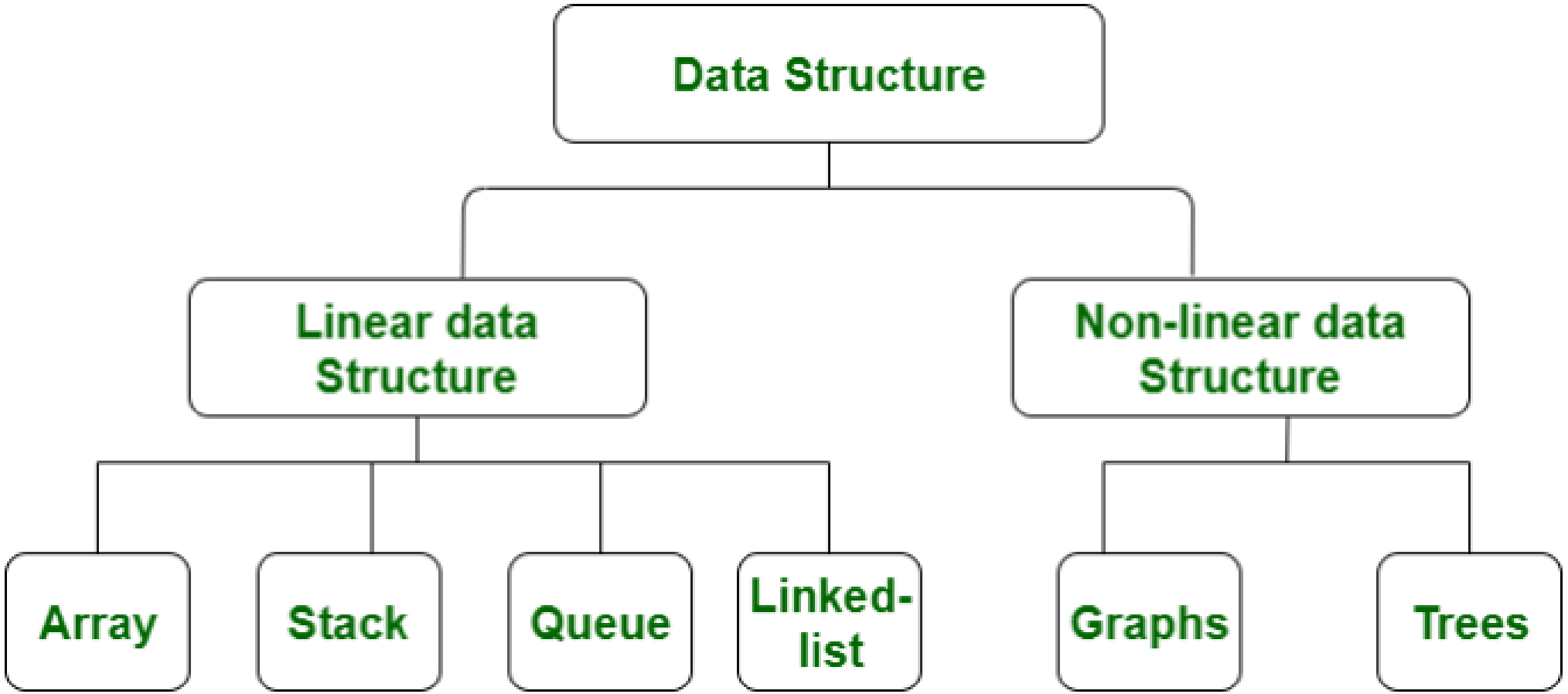
Examples:
- Search operations on an unsorted vs. sorted array.
- Time complexity differences between algorithms.



Source :

Reference
Reference

Linear vs Non-linear data structure



Source :

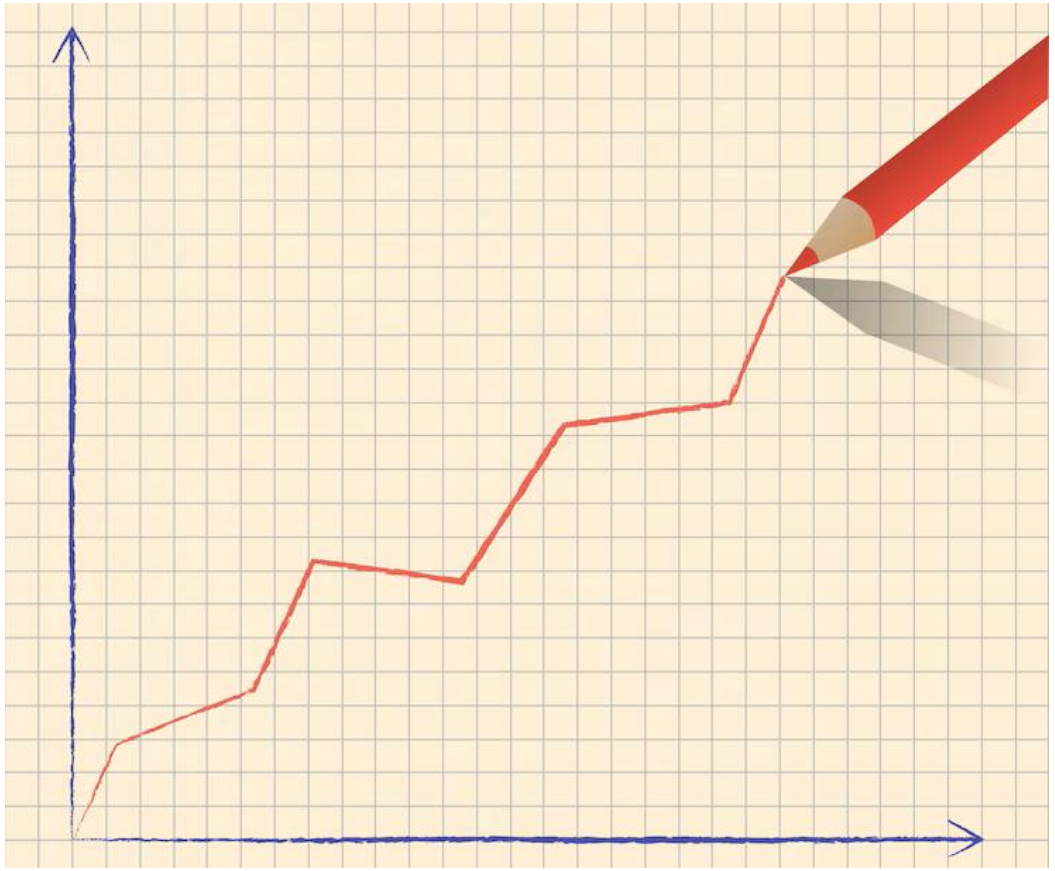
Reference

Linear Data Structure	Non-linear Data Structure
In linear data structure, single level is involved.	Whereas in non-linear data structure, multiple levels are involved.
In a linear data structure, data elements are arranged in a linear order where each and every element is attached to its previous and next adjacent.	In a non-linear data structure, data elements are attached in hierarchically manner.
In linear data structure, data elements can be traversed in a single run only.	While in non-linear data structure, data elements can't be traversed in a single run only.
In a linear data structure, memory is not utilized in an efficient way.	While in a non-linear data structure, memory is utilized in an efficient way.
Applications of linear data structures are mainly in application software development.	Applications of non-linear data structures are in Artificial Intelligence and image processing.
Linear data structures are useful for simple data storage and manipulation.	Non-linear data structures are useful for representing complex relationships and data hierarchies, such as in social networks, file systems, or computer networks.

Source :

Introduction to Asymptotic analysis

Asymptotic analysis evaluates algorithm efficiency by examining performance as input size approaches infinity, focusing on how time and space requirements scale without considering constant factors or lower-order terms.



What is the meaning of the term asymptotic in terms of Computer Science?

In computer science, asymptotic analysis describes algorithm efficiency as input size approaches infinity, focusing on resource usage (time and space) without being affected by specific details or constant factors.

Source :

Reference

Important to understand Asymptotic Analysis

- **Predicting Performance at Scale**: Forecasts algorithm performance with large inputs.
- **Comparing Algorithms**: Standardizes evaluation of different algorithms' efficiency.
- **Identifying Bottlenecks**: Pinpoints the most resource-intensive algorithm parts.
- **Designing Efficient Algorithms**: Assists in creating scalable and efficient algorithms.
- **Resource Management**: Optimizes computational resource use, especially in constrained environments.
- **Standardized Metric**: Uses Big O, Big Omega, and Big Theta for describing algorithm efficiency.

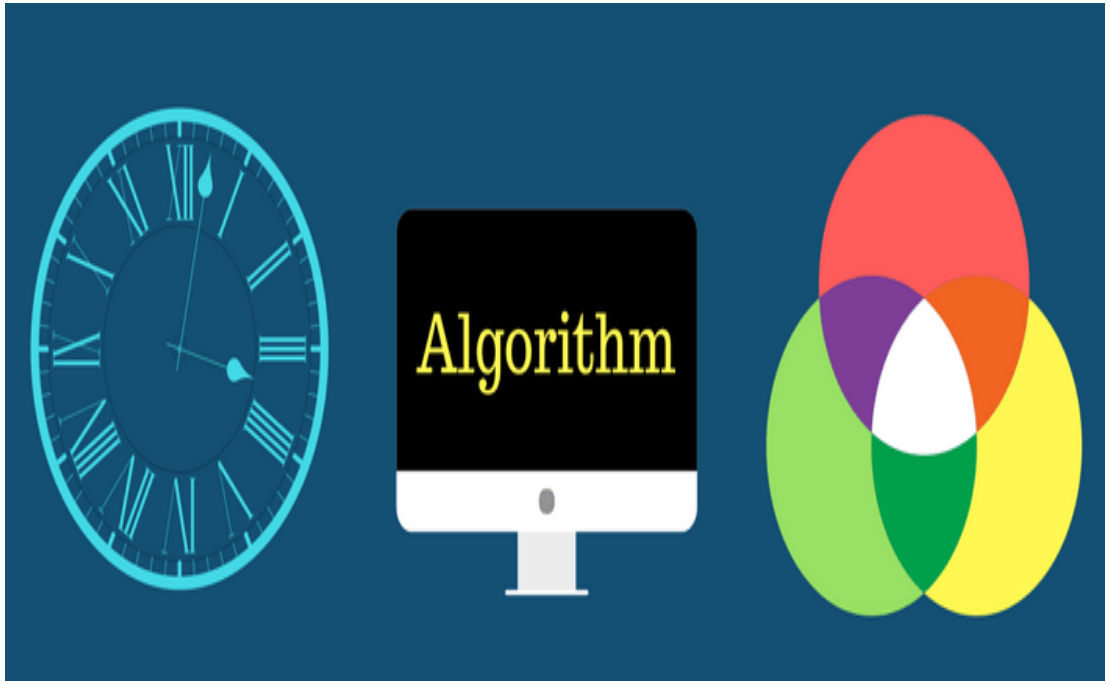
Source :

Basics of Algorithm Analysis and Complexity Theory

Algorithm Analysis:
- Evaluating the performance of algorithms in terms of time and space.

Complexity Theory:
- Classifying problems based on their inherent difficulty.

Key Concepts:
- Time Complexity
- Space Complexity



Source :

Reference

Time and Space Complexity Analysis

Time Complexity:

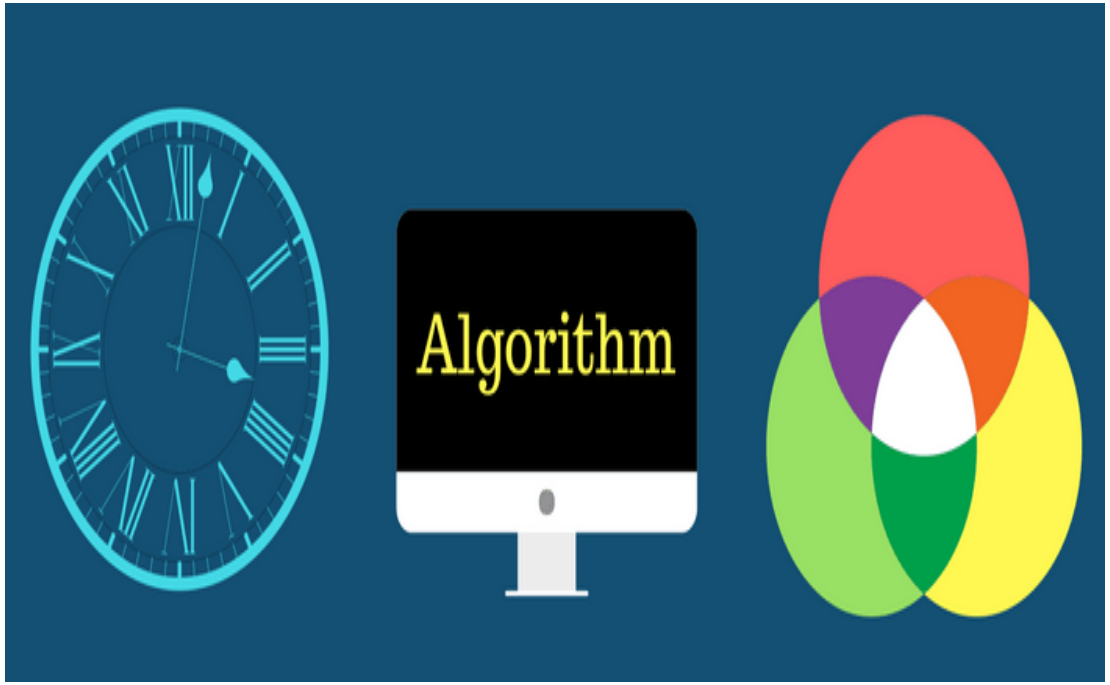
- Measures the amount of time an algorithm takes as a function of the input size.

Space Complexity:

- Measures the amount of memory an algorithm uses as a function of the input size.

Example Analysis:

- Analyzing a simple loop vs. a nested loop.



Source :

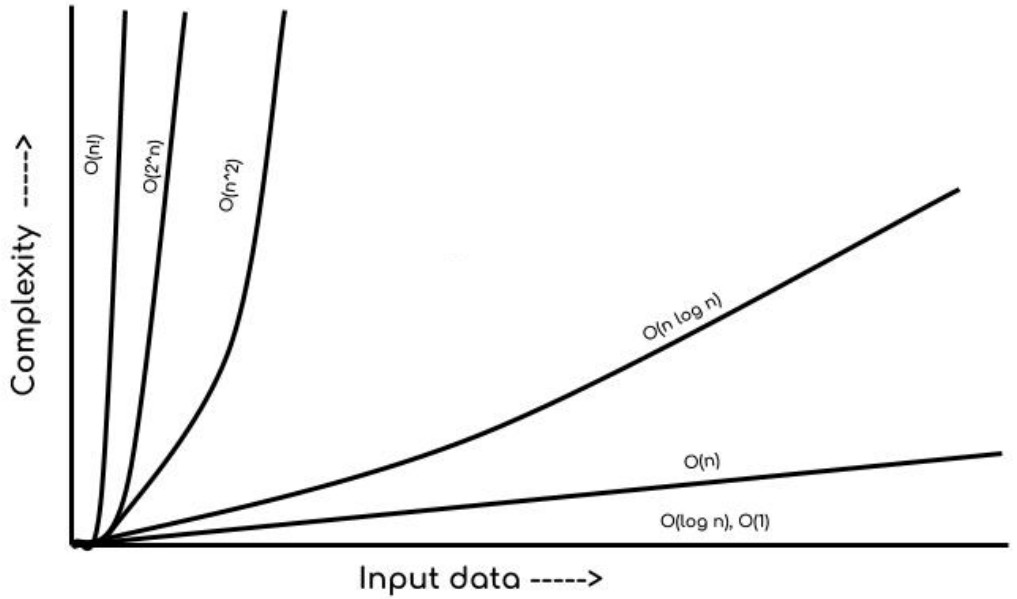
Reference

Asymptotic Notation

Big O (O):
- Upper bound on the time; worst-case scenario.

Big Omega (Ω):
- Lower bound on the time; best-case scenario.

Big Theta (Θ):
- Tight bound on the time; average-case scenario.



Reference

Practical Example:

- **Imagine we are tasked with sorting a list of one million elements**.
- If we use bubble sort, with a time complexity of $O(n^2)$, the number of operations needed scales quadratically, leading to approximately 10121012 operations.
- In contrast, using merge sort, with a time complexity of O(nlogn), requires roughly 107log2(107)≈2×108107log2 (107)≈2×108 operations. The difference is significant, demonstrating the practical impact of choosing an algorithm based on asymptotic analysis.

Conclusion

- Understanding asymptotic analysis is crucial because it enables the prediction, comparison, optimization, and efficient design of algorithms. It is a fundamental skill for computer scientists and software developers, ensuring that they can create and work with algorithms that are both effective and efficient, especially as they scale to handle large inputs.

Source :

Lab - 1

Time Complexity Analysis

Source :

Conclusion

We have completed this section and now we have understood about:

- Data structures and algorithms, importance of efficient data organization and algorithm designs
- Linear vs non-linear data structure, basics of algorithm analysis and complexity theory
- Time and space complexity analysis, & asymptotic notation

Source :

References

- Recommended Books:
  - "Data Structures and Algorithms in Python" by Michael T. Goodrich
  - "Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein
  - Cornell University lecture on Asymptotic Analysis

- Other resource reference link
  - Data Structure intro image
  - Intro to Asymptotic Analysis
  - Linear vs Non-linear data structure image

Source :


Q&A
Let's Start

Quiz

1. Which of the following is NOT a linear data structure?

a) Array
b) Linked List
c) Tree
d) Queue

Answer: C
Tree

Quiz

2. Why is it important to design efficient algorithms?

a) To minimize the use of CPU and memory resources
b) To ensure the program runs quickly
c) To handle larger datasets effectively
d) All of the above

Answer: D
All of the above

Quiz

3. Which scenario best describes a situation where efficient data organization is critical?

a) Sorting a small list of numbers
b) Searching a database with millions of records
c) Displaying a static webpage
d) Printing a document

Answer: B

Searching a database with millions of records

Quiz

4. Which data structure is used to represent hierarchical relationships?

a) Array
b) Tree
c) Stack
d) Queue

Answer: B
Tree

Thank You