

# Introduction to Data Structures and Algorithms

---

# What are Data Structures?

- Way of organizing and storing data efficiently

- Helps in accessing, processing, and modifying data quickly

- Examples: Arrays, Linked Lists, Stacks, Queues, Trees, Graphs

# What are Algorithms?

- Step-by-step method to solve a problem
- Properties:
  - Input & Output
  - Definiteness (clear steps)
  - Finiteness (terminates)
  - Effectiveness (feasible)
- Examples: Searching, Sorting, Pathfinding

# Importance of DSA



- Efficient problem solving



- Improves performance & scalability



- Essential in competitive programming & interviews

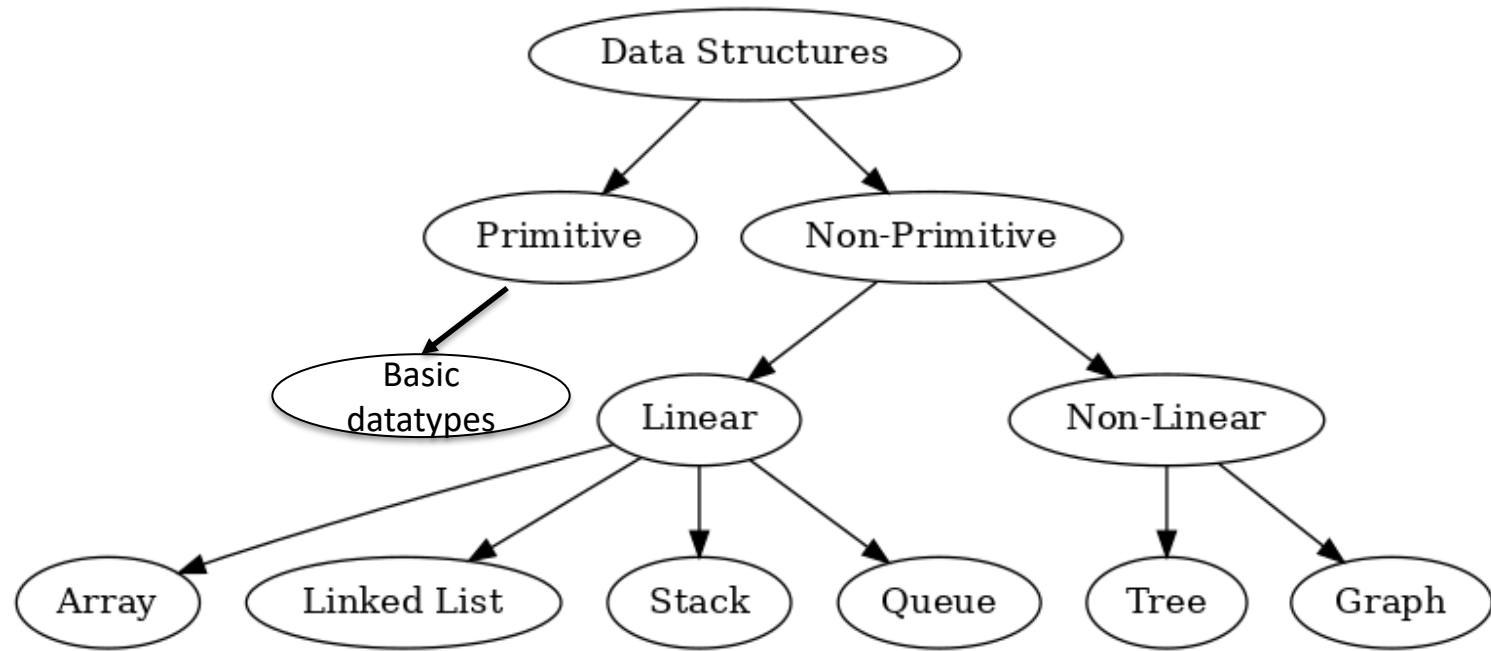


- Backbone of software systems (OS, DBMS, AI, Networks)



DSA = Heart of Computer Science

# Types of Data Structures



# Types of Algorithms

- Searching: Linear Search, Binary Search

- Sorting: Bubble Sort, Merge Sort, Quick Sort

- Divide & Conquer: Binary Search, Merge Sort

- Greedy Algorithms: Kruskal's, Prim's

- Dynamic Programming: Fibonacci, Knapsack

- Graph Algorithms: BFS, DFS, Dijkstra's

# Algorithm Analysis

- Performance Measurement:

- Time Complexity → How fast?

- Space Complexity → How much memory?

- Asymptotic Notations:

- Big-O (Worst case)

- $\Omega$  (Best case)

- $\Theta$  (Average case)

# Applications of DSA

- Databases (Indexing, Searching)

- Operating Systems (Scheduling, Memory Management)

- Networking (Routing, Shortest Path)

- Cybersecurity (Encryption algorithms)

- AI & ML (Decision Trees, Graphs, Optimization)

- Games (Pathfinding, Simulation)



# Summary

---

- Data Structures = Organization of Data

---

- Algorithms = Step-by-step solutions

---

- Both together → Solve problems efficiently

---

- Core skill for software engineers, researchers & students

# Thank You

