Data Structure Operations Overview

Heap (Min-Heap / Max-Heap)

Operations:

- insert(): Add a new element while maintaining the heap property. O(log n)

- extractMin()/extractMax(): Remove and return the top element. O(log n)

- peek(): Return the top element without removing it. O(1)

- heapify(): Build a heap from an array. O(n)

Used in:

- Priority queues

- Sorting (Heapsort)

- Graph algorithms (Dijkstra)

Queue

Operations:

- enqueue(item): Add item to the end. O(1)

- dequeue(): Remove item from the front. O(1)

- peek(): View the front item without removing it. O(1)

- isEmpty(): Check if the queue is empty. O(1)

Used in:

- Breadth-first search (BFS)

- Scheduling tasks

- Buffers in IO operations

Deque (Double-Ended Queue)

Data Structure Operations Overview

Operations:

- push_front(item): Add to front. O(1)
- push_back(item): Add to back. O(1)
- pop_front(): Remove from front. O(1)
- pop_back(): Remove from back. O(1)
- peek_front()/peek_back(): View front or back item. O(1)

Used in:

- Sliding window problems
- Task schedulers

Nodes (Linked List Element)

Operations:

- createNode(item): Create a node with a value.
- addNext(node): Link to another node (singly or doubly).
- traverse(): Visit all nodes from head to end. O(n)
- delete(item): Remove a node with given value. O(n)

Used in:

- Linked lists
- Trees and graphs
- Stack and queue implementations