

Data Structures and Algorithms - Study Notes

Chapter 1: Arrays and Strings

1.1 Array Fundamentals

• Arrays store elements in contiguous memory locations

• Time complexity: Access $O(1)$, Search $O(n)$, Insert/Delete $O(n)$

• Space complexity: $O(n)$

1.2 Common Array Operations

• Traversal: `for(int i = 0; i < n; i++)`

• Binary Search: $O(\log n)$ for sorted arrays

• Two Pointer Technique for pair problems

1.3 Dynamic Arrays vs Static Arrays

• Static: Fixed size, faster access

• Dynamic: Resizable, flexible but overhead

Chapter 2: Linked Lists

2.1 Singly Linked List

• Each node contains data and pointer to next node

• Insert at beginning: $O(1)$

• Insert at end: $O(n)$ without tail pointer

• Delete: $O(n)$ for search + $O(1)$ for removal

Chapter 3: Stacks and Queues

3.1 Stack (LIFO - Last In First Out)

â€ Operations: push(), pop(), peek(), isEmpty()

â€ Applications: Function calls, expression evaluation

â€ Balanced parentheses checking

3.2 Queue (FIFO - First In First Out)

â€ Operations: enqueue(), dequeue(), front(), isEmpty()

â€ Applications: BFS, task scheduling

Chapter 4: Trees

4.1 Binary Tree Properties

â€ Maximum nodes at level i : 2^i

â€ Maximum nodes in tree of height h : $2^{(h+1)} - 1$

â€ Minimum height: $\log_2(n+1) - 1$

4.2 Tree Traversals

â€ Inorder: Left -> Root -> Right

â€ Preorder: Root -> Left -> Right

â€ Postorder: Left -> Right -> Root

â€ Level order: BFS using queue