Time and Space Complexity ..

DSA Algorithms: Time and Space Complexity Overview

This document contains a comprehensive summary of commonly used Data Structures and Algorithms (DSA), categorized by type, along with their time and space complexities for best, average, and worst cases where applicable.

1. Sorting Algorithms

Algorithm	Best Time	Avg Time	Worst Time	Space	Stable
Bubble Sort	O(n)	O(n^2)	O(n^2)	O(1)	Yes
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)	Yes
Selection Sort	O(n^2)	O(n^2)	O(n^2)	O(1)	No
Merge Sort	O(n log n)	O(n log n)	O(n log n)	O(n)	Yes
Quick Sort	O(n log n)	O(n log n)	O(n^2)	O(log n)	No
Heap Sort	O(n log n)	O(n log n)	O(n log n)	O(1)	No
Counting Sort	O(n + k)	O(n + k)	O(n + k)	O(k)	Yes
Radix Sort	O(nk)	O(nk)	O(nk)	O(n + k)	Yes
Bucket Sort	O(n + k)	O(n + k)	O(n^2)	O(n + k)	Yes

2. Searching Algorithms

Algorithm	Best Time	Avg Time	Worst Time	Space
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)

Note: Binary Search works only on sorted arrays.

3. Recursion

- Time and space complexity depends on the recursive tree:
  - **Simple Recursion**: O(n) time and O(n) space
  - **Tail Recursion**: Can be optimized to O(1) space
  - **Fibonacci (naive)**: O(2^n) time, O(n) space
  - **Fibonacci (memoized)**: O(n) time, O(n) space

4. Hashing

Operation	Avg Time	Worst Time	Space
Insert	O(1)	O(n)	O(n)
Search	O(1)	O(n)	O(n)
Delete	O(1)	O(n)	O(n)

Note: Worst case occurs when many elements hash to same index.

5. Stack and Queue

Operation	Time	Space
Push/Pop (Stack)	O(1)	O(n)
Enqueue/Dequeue	O(1)	O(n)

6. Linked List

Operation	Time	Space
Insert/Delete at head	O(1)	O(n)
Insert/Delete at tail	O(n)	O(n)
Search	O(n)	O(n)

7. Trees

Binary Search Tree (BST)

Operation	Avg Time	Worst Time	Space
Insert	O(log n)	O(n)	O(n)
Search	O(log n)	O(n)	O(n)
Delete	O(log n)	O(n)	O(n)

Balanced BST (AVL, Red-Black Tree)

Operation	Time	Space
All Ops	O(log n)	O(n)

Binary Heap

Operation	Time	Space
Insert	O(log n)	O(n)
Delete Min/Max	O(log n)	O(n)
Peek	O(1)	O(n)

8. Graph Algorithms

Algorithm	Time Complexity	Space Complexity
BFS	O(V + E)	O(V)
DFS	O(V + E)	O(V)
Dijkstra	O((V + E) log V) (min-heap)	O(V + E)
Bellman-Ford	O(VE)	O(V)
Floyd-Warshall	O(V^3)	O(V^2)
Kruskal’s MST	O(E log E)	O(V)
Prim’s MST	O(E + log V)	O(V)
Topo Sort (DFS)	O(V + E)	O(V)

9. Dynamic Programming

- **Knapsack (0/1)**: O(nW) time, O(nW) space
- **Longest Common Subsequence (LCS)**: O(n*m) time and space
- **Longest Increasing Subsequence**: O(n log n) time (with binary search), O(n) space
- **Edit Distance**: O(n*m) time and space

10. Trie

Operation	Time Complexity	Space
Insert	O(L)	O(nL)
Search	O(L)	O(nL)
Delete	O(L)	O(nL)

n = number of words, L = average word length.

This covers the most important DSA algorithms and their time/space complexities. For advanced topics like segment trees, Fenwick trees, suffix arrays, etc., let me know if you'd like a continuation.