Design and Analysis of Algorithms

DAA tests algorithm choice, complexity, and paradigms.

Big-O

Big-O describes upper-bound growth of time/space with input size.

Order from best to worst:

```
O(1) < O(log n) < O(n) < O(n log n) < O(n²) < O(2^n) < O(n!)
```

Searching

Linear search

- Works on sorted or unsorted arrays.
- Time: O(n)

Binary search

- Requires sorted array.
- Time: O(log n)
- Method: repeatedly divide search range in half.

Sorting

Algorithm	Best	Average	Worst	Stable?
Bubble sort	O(n) optimized	O(n²)	O(n²)	Yes
Selection sort	O(n²)	O(n²)	O(n²)	No
Insertion sort	O(n)	O(n²)	O(n²)	Yes
Merge sort	O(n log n)	O(n log n)	O(n log n)	Yes
Quick sort	O(n log n)	O(n log n)	O(n²)	No
Heap sort	O(n log n)	O(n log n)	O(n log n)	No

Divide and conquer

Steps:

1. Divide problem.

2. Solve subproblems.
3. Combine results.

Examples:

- binary search,
- merge sort,
- quick sort.

Greedy algorithms

Greedy chooses the best local option at each step.

Examples:

- activity selection,
- fractional knapsack,
- Prim's algorithm,
- Kruskal's algorithm,
- Dijkstra's algorithm.

Trap:

Greedy does not always give global optimum unless greedy-choice property holds.

Dynamic programming

Used when:

- overlapping subproblems,
- optimal substructure.

Examples:

- Fibonacci with memoization,
- 0/1 knapsack,
- longest common subsequence,
- matrix chain multiplication.

Memoization: top-down.
Tabulation: bottom-up.

Recursion

Recursion means function calls itself.

Must have:

- base case,

- recursive case.

Common recurrence:

```
T(n) = T(n/2) + O(1) -> O(log n)
T(n) = 2T(n/2) + O(n) -> O(n log n)
T(n) = T(n-1) + O(1) -> O(n)
```

Graph algorithms

BFS

- Uses queue.
- Finds shortest path in unweighted graph.
- Time: O(V+E).

DFS

- Uses stack/recursion.
- Useful for cycle detection and topological sort.
- Time: O(V+E).

Dijkstra

- Finds shortest path from source.
- Works with non-negative edge weights.

Kruskal

- Finds minimum spanning tree.
- Sorts edges by weight.
- Uses disjoint set/union-find.

Prim

- Finds minimum spanning tree.
- Grows tree from a starting vertex.

Common MCQ traps

- Binary search needs sorted input.
- Quick sort worst case is $O(n^2)$.
- Merge sort is stable and O(n log n), but needs extra space.
- BFS gives shortest path only in unweighted graph.
- Dijkstra does not work correctly with negative weights.
- DP is useful when subproblems repeat.