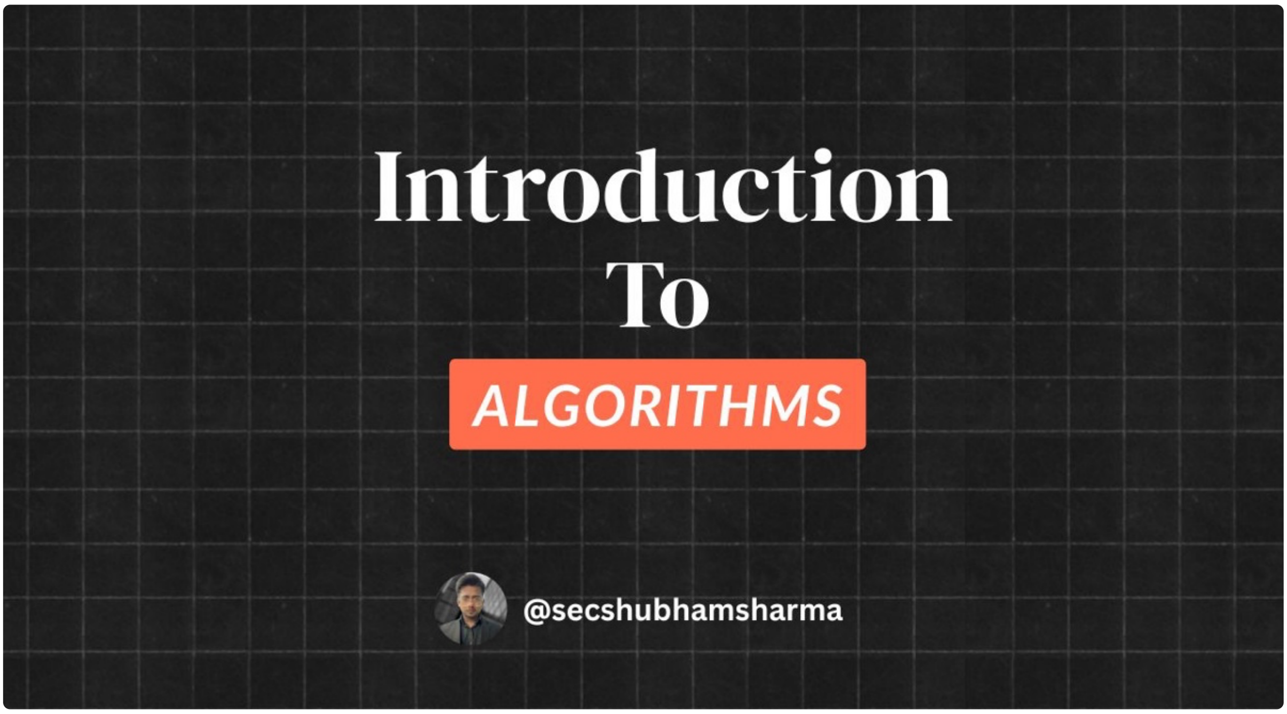
Introduction To Algorithms



An **algorithm** is a **step-by-step, well-defined procedure** to solve a specific problem in a finite number of steps.

Let's Get To Know Overview of Some Algorithms

Brute Force Algorithm

This is the most basic and straightforward way to solve a problem **try every possible option** until you find the answer.

> Example: Linear Search (checking each element one by one)

Greedy Algorithm

A **greedy algorithm** is a method that **makes the best choice at each step**, hoping that these local best choices will lead to the **overall best solution**.

> Example:
>
> - Fractional Knapsack Problem
> - Prim’s & Kruskal’s MST algorithms

Divide and Conquer

This is about **breaking the problem into smaller chunks**, solving them independently, and then combining the results.

> Famous examples:
>
> - Merge Sort
> - Quick Sort
> - Binary Search

Dynamic Programming (DP)

Dynamic Programming is a method of solving problems by breaking them into smaller subproblems, solving each subproblem once and storing their results to avoid repeating the same work.

> Examples:
>
> - Fibonacci using memoization
> - Longest Common Subsequence
> - 0/1 Knapsack

Between **Divide and Conquer** vs **Dynamic Programming**

Aspect	Divide and Conquer	Dynamic Programming
Problem Structure	Divides the problem into **independent subproblems**	Divides the problem into **overlapping subproblems**
Subproblem Reuse	**No reuse –** subproblems are solved **separately** each time	**Yes –** results of subproblems are **stored and reused**

- **Divide & Conquer**:
  ➤ Break into pieces, solve each from scratch
  ➤ No memory of past work
- **Dynamic Programming**:
  ➤ Break into pieces, **remember solutions**
  ➤ Reuse past answers to save time

Backtracking

Backtracking is a method of trying out all possible options and undoing choices that don't lead to a solution.

> Examples:
>
> - N-Queens Problem
> - Sudoku Solver

> - Permutations

Recursion

Recursion is a programming technique where a function solves a problem by calling itself on smaller inputs, continuing this process until it reaches a simple case it can solve directly (called the **base case**).

> Example:
>
> - Tower of Hanoi
> - Factorial
> - Tree traversals

Randomized Algorithms

A **randomized algorithm** is an algorithm that **uses** **random numbers** or random choices during its logic to solve a problem.

> Example:
>
> - Randomized QuickSort
> - Monte Carlo methods
> - Hashing algorithms

Graph Algorithms

Graph algorithms are a set of techniques used to **solve problems related to graphs**, such as finding paths, cycles, shortest distances, or connectivity between nodes.

Examples:

> - BFS (Breadth-First Search)
> - DFS (Depth-First Search)
> - Dijkstra’s algorithm
> - Bellman-Ford
> - Floyd-Warshall

Some More Specialized Alogrithm Types

1. **Sorting Algorithms**: Bubble, Insertion, Merge, Quick, Heap
2. **Searching Algorithms**: Binary Search, Interpolation Search
3. **String Algorithms**: KMP, Rabin-Karp, Z-algorithm
4. **Mathematical Algorithms**: Euclidean GCD, Sieve of Eratosthenes
5. **Machine Learning Algorithms**: Decision Trees, KNN, SVM, etc.

This was the overview of me giving you the algorithms, we will be diving deep in each and every topics in detaiil and we will be solving problems