**Important Algorithms**

**Binary Search** : Quickly find a value in a sorted array.

**Merge Sort :**  Efficient, stable sorting via divide-and-conquer.

**Quick Sort :** Fast sorting with average - case O(n log n) complexity.

**Bubble Sort:** Simple sorting by repeatedly swapping adjacent elements.

**Insertion Sort** : Builds the final sorted array one item at a time.

**Selection Sort :**  Selects the smallest element and swaps it to the front.

**Heap Sort :** Sorts using a binary heap data structure.

**DFS (Depth-First Search) :** Traverse deep in each branch before backtracking.

**BFS (Breadth-First Search)** : Traverse all neighbors at the current depth first.

**Dijkstra's Algorithm :** Finds shortest paths from a single source in a graph.

**Bellman-Ford Algorithm** : Computes shortest paths with negative edge weights allowed.

**Floyd-Warshall Algorithm** : Finds shortest paths between all pairs of vertices.

**Knapsack Problem :** Maximizes value within a weight limit (dynamic programming).

**Kadane's Algorithm :** Finds the maximum sum contiguous subarray.

**Prim's Algorithm :** Efficient minimum spanning tree for a connected graph.

**Kruskal's Algorithm** : Finds minimum spanning tree for a graph using disjoint sets.

**Hashing Algorithm :** Converts data of arbitrary size to a fixed size.

**RSA Algorithm :** A public-key cryptographic algorithm for secure data transmission.

**KMP(Knuth-Morris-Pratt) Algorithm :** Efficient substring search in a string.

**Boyer-Moore Majority Vote Algorithm :**  Finds the majority element in an array.

