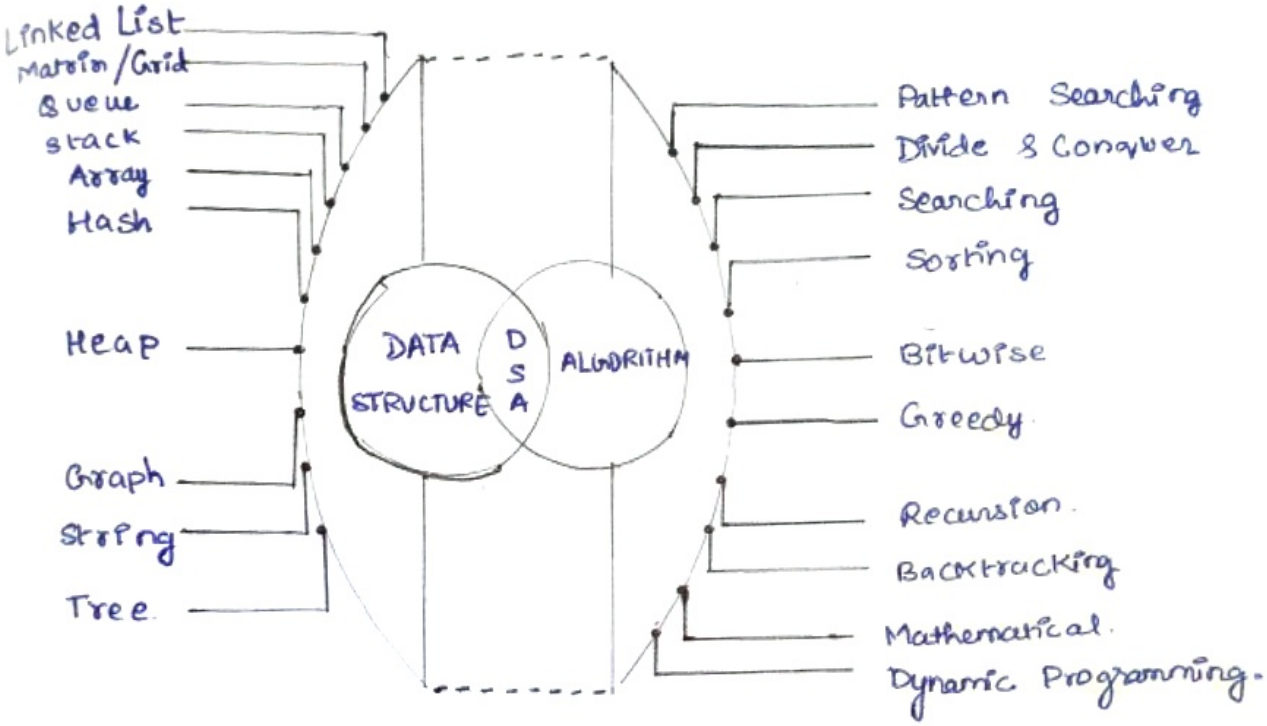
DSA

definition → Dsa (Data Structure and Algorithms) refers to the study of methods for organizing and storing data and the design of procedures for solving problems.

★ DSA stands for Data Structures and Algorithms.



★ 5 steps that we will follow to learn DSA ?

1. Learn at least one Programming Language (Mine is Java)
2. Learn about Time and Space Complexity (prefer Youtube Code with Harry)
3. Learn Data structure and Algorithms.
4. Practice, Practise and Consistency follow up.—.
5. Compete start testing and become a Pro.

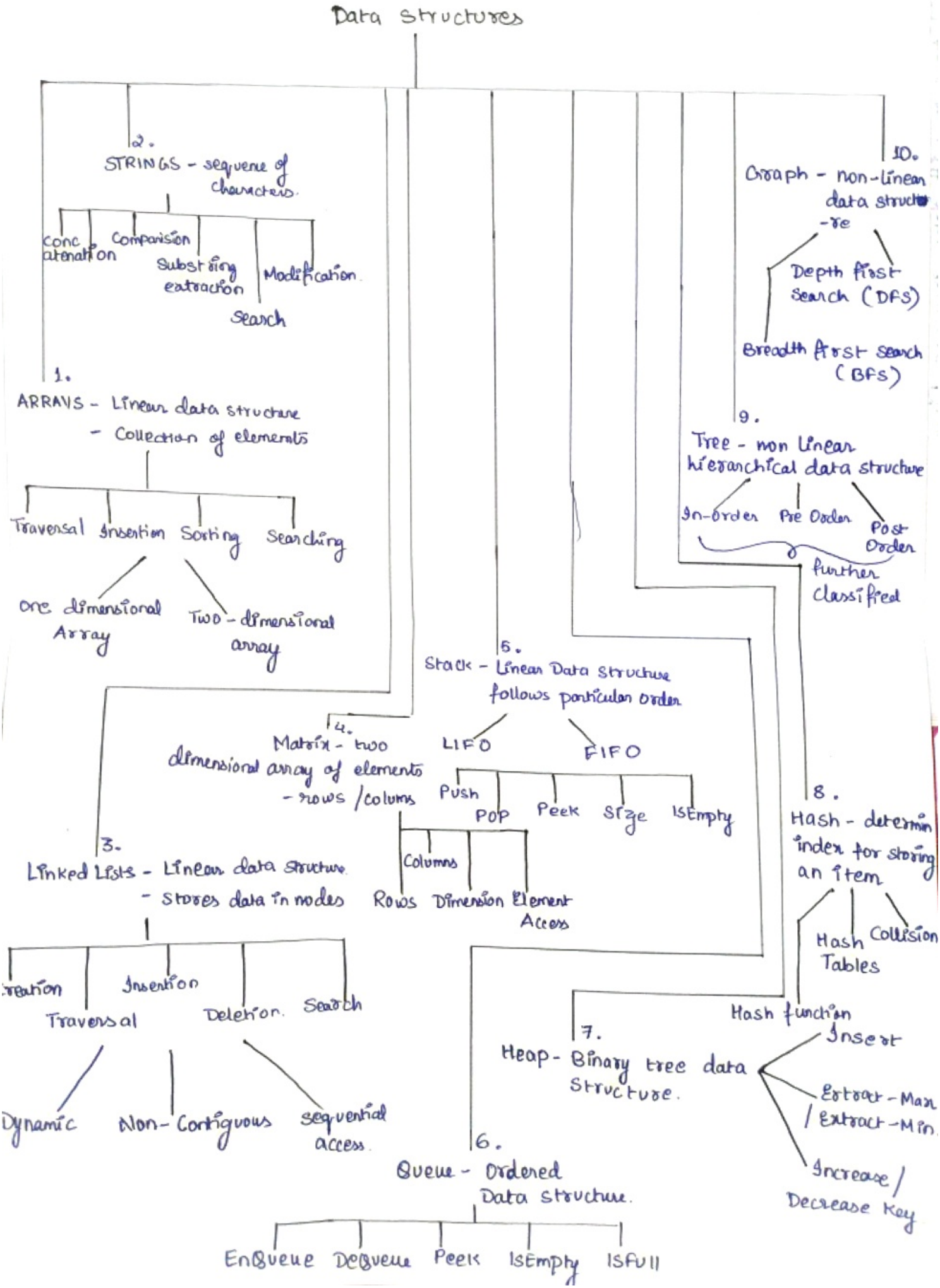
★ DSA consists of two parts :-

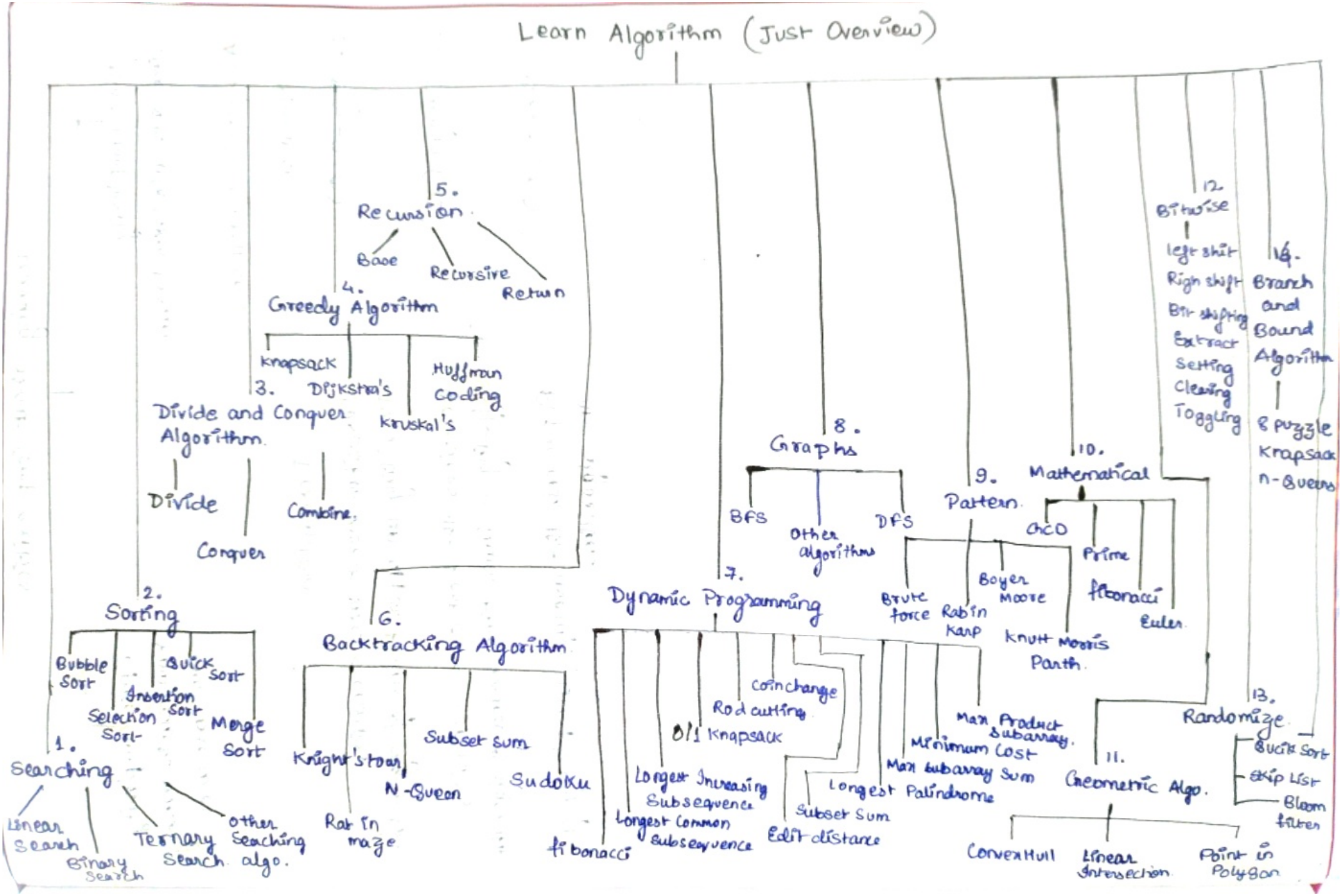
- Data structures
- Algorithms

We will move on with data structure first.

1. Learn Data structures :- They help organize and store data efficiently in computer memory one of the essential components.

- Common structures include array, Linked Lists, Stacks, Queues, trees and graphs.


Data Structures
1.
ARRAYS - Linear data structure
- Collection of elements
Traversal
Insertion
Sorting
Searching
One dimensional Array
Two-dimensional array
2.
STRINGS - sequence of characters.
Concatenation
Comparision
Substring extraction
Modification
Search
3.
Linked Lists - Linear data structure.
- Stores data in nodes
Creation
Traversal
Insertion
Deletion
Search
Dynamic
Non-Contiguous
sequential access.
4.
Matrix - two dimensional array of elements - rows / columns
Rows
Columns
Dimension
Element Access
5.
Stack - Linear Data Structure follows particular order
LIFO
FIFO
Push
Pop
Peek
Size
IsEmpty
6.
Queue - Ordered Data structure.
EnQueue
DeQueue
Peek
IsEmpty
IsFull
7.
Heap - Binary tree data structure.
8.
Hash - determine index for storing an item
Hash Tables
Collision
Hash function
Insert
Extract-Max / Extract-Min
Increase / Decrease Key
9.
Tree - non Linear hierarchical data structure
In-order
Pre Order
Post Order
further classified
10.
Graph - non-linear data structure
Depth first Search (DFS)
Breadth first search (BFS)


Learn Algorithm (Just Overview)
1. Searching
Linear Search
Binary Search
Ternary Search
other Searching algo.
2. Sorting
Bubble Sort
Selection Sort
Insertion Sort
Quick sort
Merge Sort
3. Divide and Conquer Algorithm
Divide
Conquer
Combine.
4. Greedy Algorithm
knapsack
Dijkstra's
Kruskal's
Huffman coding
5. Recursion
Base
Recursive
Return
6. Backtracking Algorithm
Knight's tour
Rat in maze
N-Queen
Subset sum
Sudoku
7. Dynamic Programming
fibonacci
Longest Increasing Subsequence
Longest Common Subsequence
0/1 knapsack
Rod cutting
coinchange
Subset Sum
Edit distance
Longest Palindrome
Minimum Cost
Max subarray Sum
Max Product Subarray.
8. Graphs
BFS
Other algorithms
DFS
9. Pattern.
Brute force
Rabin Karp
Boyer Moore
Knuth Morris Parth.
10. Mathematical
GCD
Prime
fibonacci
Euler
11. Geometric Algo.
ConvexHull
Linear Intersection
Point in Polygon
12. Bitwise
left shift
Right shift
Bit shifting
Extract
Setting
Clearing
Toggling
13. Randomize
Quick Sort
Skip List
Bloom filter
14. Branch and Bound Algorithm
8 puzzle
Knapsack
n-Queens

Learning about Complexities

Time Complexity

Time our code takes to run

Space Complexity

how much memory our code uses

Asymptotic Notation :-

Notation	Description
1. Big-O (O)	Describes worst case scenario, Providing an upper bound of algorithm.
2. Omega (Ω)	Describes the best case scenario, Offering a lower time bound of algorithms
3. Theta (Θ)	Represents the average complexity of an algorithm.