DSA

Daya Shankar

Resources

Books

CRACKING the · CODING INTERVIEW (PDFDrive).pdf
Data Structures and Algorithms Made Easy in Java.pdf

COMPLETE ROADMAAP.pdf

Videos & Courses

CN
Striver
College Wallah

using c++
Babbar

using Java
Mosh
Apna College Alpha

Sheets

- leetcode.com
- interviewcake.com

Basics TCS NQT - Coding Sheet by Arsh

Pratik
250+DSA QUESTIONS FOR PLACEMENTS.pdf Ashwani
Apna College 375 Problems DSA Sheet Alpha

Sriver AtoZ
Striver on cn

self

Geeks for Geeks SDE Sheet:-
https://www.geeksforgeeks.org/sde-sheet-a-complete-guide-for-sde-preparation/

Love Babbar 450 Problems DSA Sheet:-
https://drive.google.com/file/d/1FMdN_OCfOI0iAeDlqswCiC2DZzD4nPsb/view

Striver 180 Problems DSA Sheet:-
https://takeuforward.org/interviews/strivers-sde-sheet-top-coding-interview-problems/

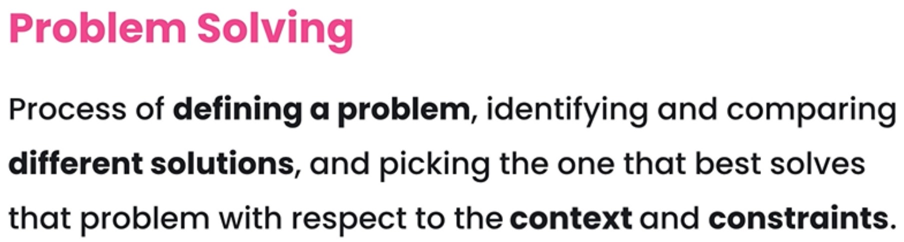
Siddharth Singh 450 Problems DSA Sheet:-
https://docs.google.com/spreadsheets/u/0/d/11tevcTIBQsIvRKIZLbSzCeN4mCO6wD4O5meyrAIfSXw/htmlview

Fraz 250 Problems DSA Sheet:-
https://docs.google.com/spreadsheets/u/0/d/1-wKcV99KtO91dXdPkwmXGTdtyxAfk1mbPXQg81R9sFE/htmlview

Arsh Goyal 280 Problems DSA Sheet:-
https://docs.google.com/spreadsheets/d/1MGVBJ8HkRbCnU6EQASjJKCqQE8BWng4qgL0n3vCVOxE/htmlview?usp=sharing&pru=AAABgKkdtIE*rPv8dPkWyOpfwjprKvKSeA

The Code Skool DSA Sheet:-
https://docs.google.com/document/u/0/d/1RxKKXJtErQFJjMfAh1kV-DyQsZoiESayimFx6PPIhVE/mobilebasic

Path



Problem Solving

Process of **defining a problem**, identifying and comparing **different solutions**, and picking the one that best solves that problem with respect to the **context** and **constraints**.

S1

- Array
- Multi-Dim Array
- String
- HashTable

S2

- LinkedList
- Stack
- Queue

S3 Approach

- Complexity analysis- time and space
- Recursion
- BackTracking
- DP

S4 Algorithm

- Binary Search
- Sorting
- String Manipulation

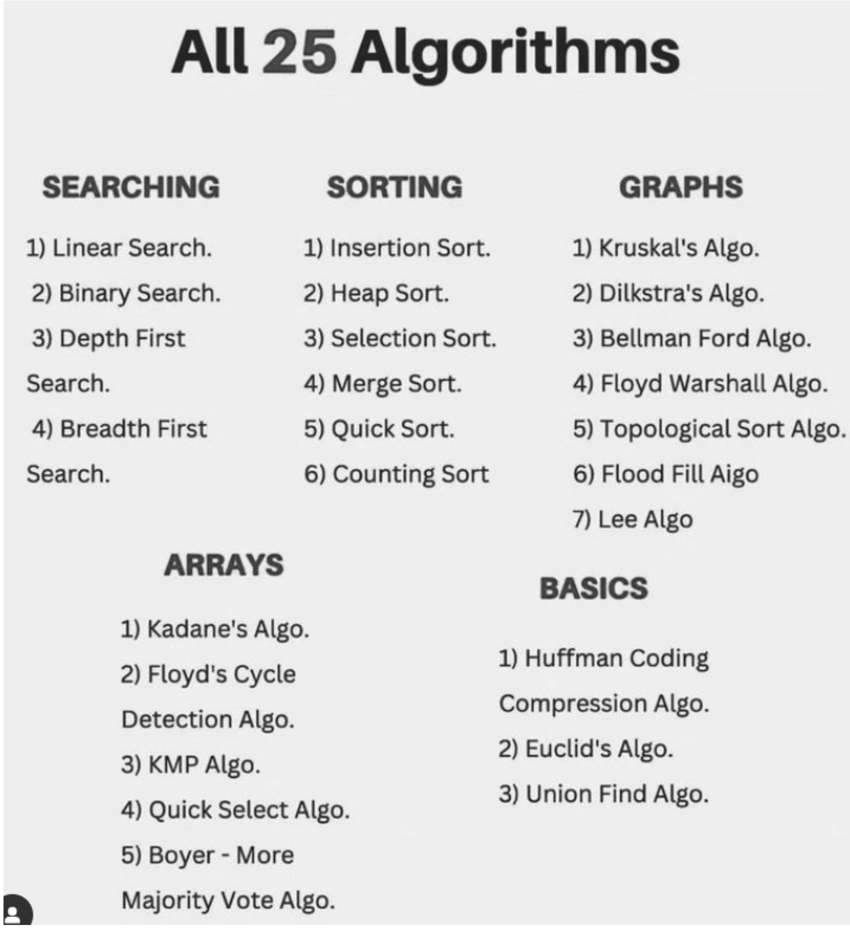
S5

- Tree
    - BT
    - BST
- Graph

S6

- Priority Queue / Heap
- Tries

S7 (Advanced)



All 25 Algorithms

SEARCHING

1) Linear Search.
2) Binary Search.
3) Depth First Search.
4) Breadth First Search.

SORTING

1) Insertion Sort.
2) Heap Sort.
3) Selection Sort.
4) Merge Sort.
5) Quick Sort.
6) Counting Sort

GRAPHS

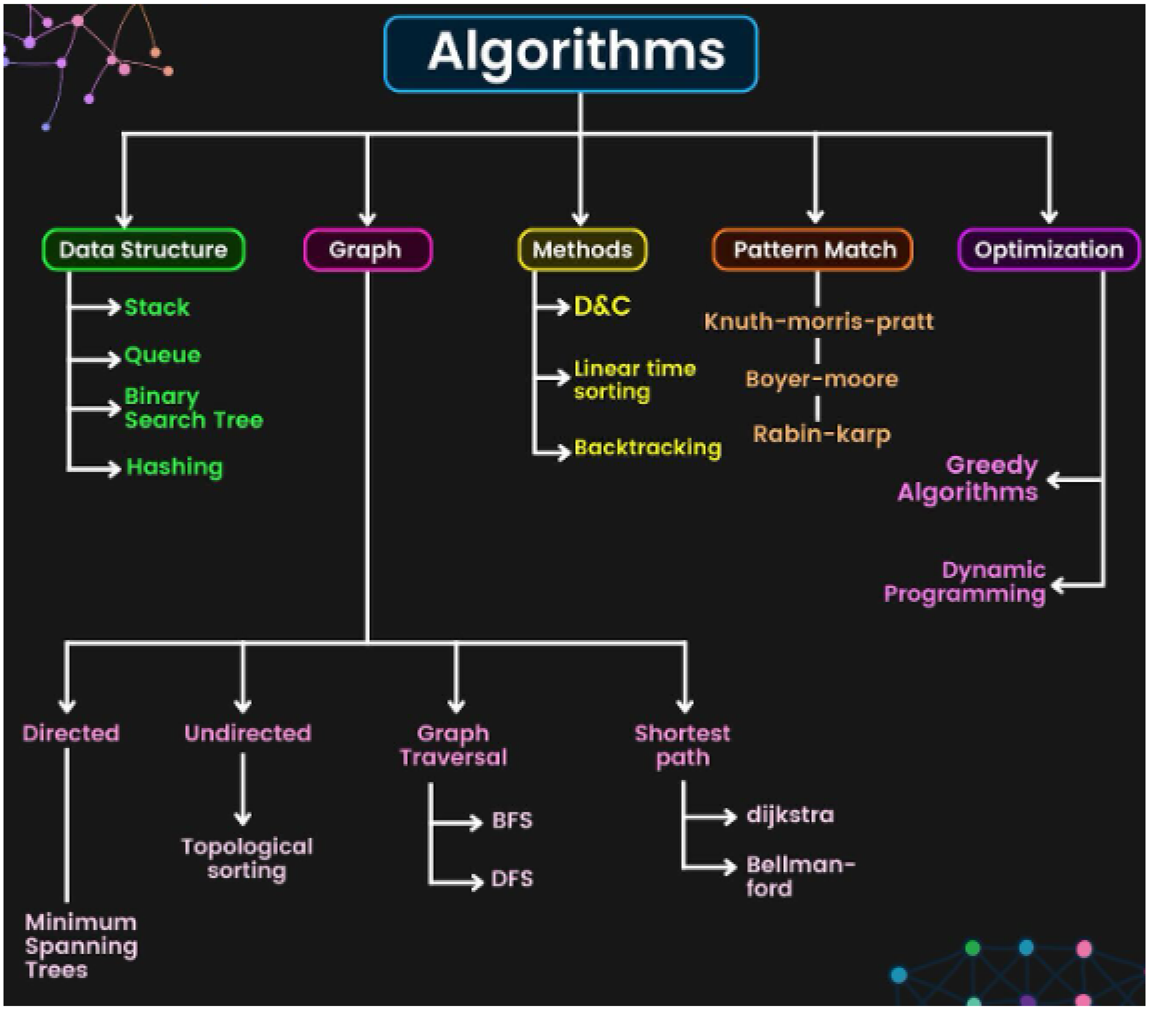
1) Kruskal's Algo.
2) Dilkstra's Algo.
3) Bellman Ford Algo.
4) Floyd Warshall Algo.
5) Topological Sort Algo.
6) Flood Fill Aigo
7) Lee Algo

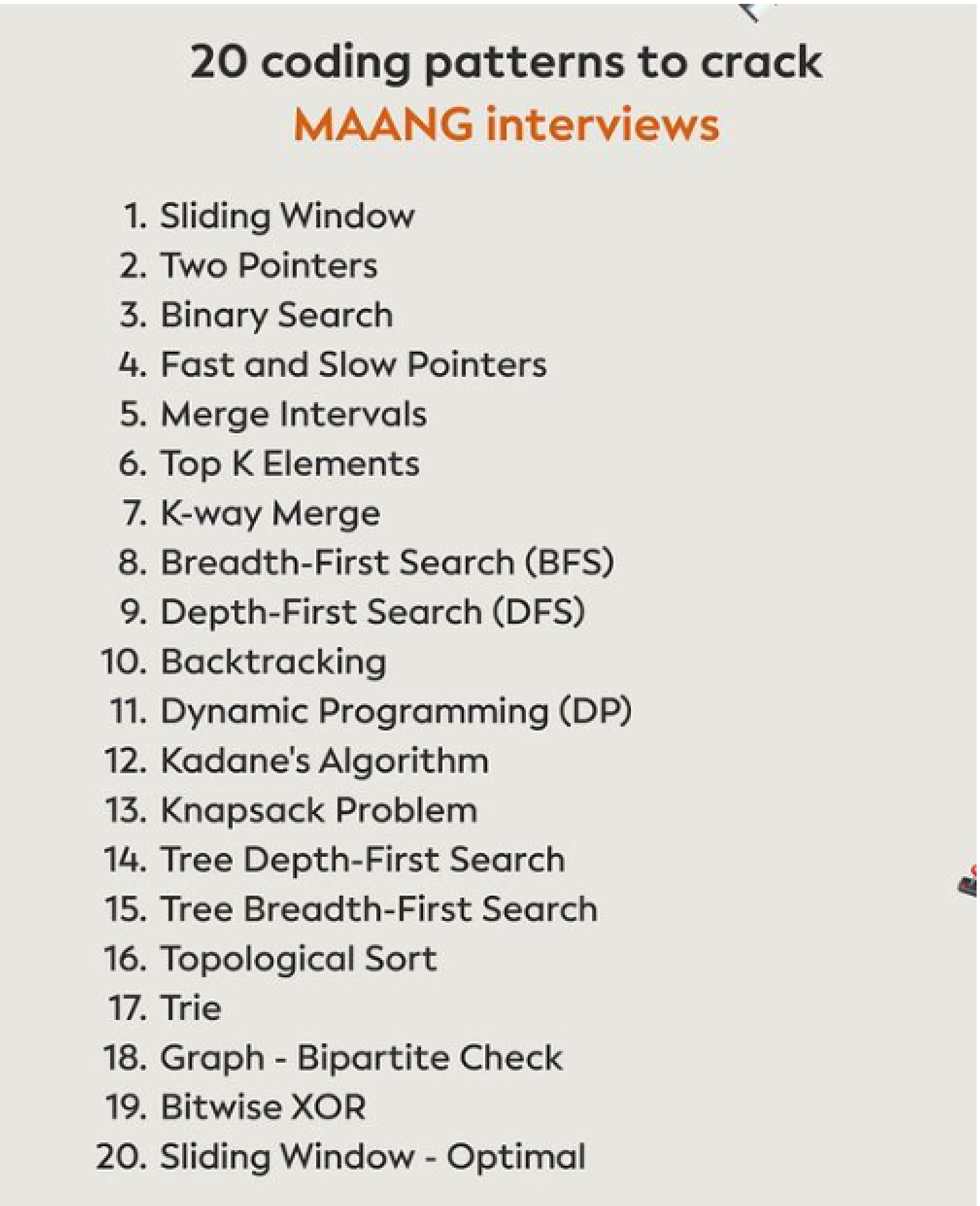
ARRAYS

1) Kadane's Algo.
2) Floyd's Cycle Detection Algo.
3) KMP Algo.
4) Quick Select Algo.
5) Boyer - More Majority Vote Algo.

BASICS

1) Huffman Coding Compression Algo.
2) Euclid's Algo.
3) Union Find Algo.


Algorithms
Data Structure
Stack
Queue
Binary Search Tree
Hashing
Graph
Methods
D&C
Linear time sorting
Backtracking
Pattern Match
Knuth-morris-pratt
Boyer-moore
Rabin-karp
Optimization
Greedy Algorithms
Dynamic Programming
Directed
Minimum Spanning Trees
Undirected
Topological sorting
Graph Traversal
BFS
DFS
Shortest path
dijkstra
Bellman-ford



20 coding patterns to crack
MAANG interviews

1. Sliding Window
2. Two Pointers
3. Binary Search
4. Fast and Slow Pointers
5. Merge Intervals
6. Top K Elements
7. K-way Merge
8. Breadth-First Search (BFS)
9. Depth-First Search (DFS)
10. Backtracking
11. Dynamic Programming (DP)
12. Kadane's Algorithm
13. Knapsack Problem
14. Tree Depth-First Search
15. Tree Breadth-First Search
16. Topological Sort
17. Trie
18. Graph - Bipartite Check
19. Bitwise XOR
20. Sliding Window - Optimal

Must Do Coding Questions for Companies like Google, Amazon, Microsoft, Adobe, … (Part 1)

1) **Arrays -**
Subarray with given sum
Count the triplets
Kadane’s Algorithm
Missing number in array
Merge two sorted arrays
Rearrange array alternatively
Number of pairs

Inversion of Array
Sort an array of 0s, 1s and 2s
Equilibrium point
Leaders in an array
Minimum Platforms
Reverse array in groups
K’th smallest element
Trapping Rain Water
Pythagorean Triplet
Chocolate Distribution Problem
Stock buy and sell
Element with left side smaller and right side greater
Convert array into Zig-Zag fashion
Last Index of 1
Spirally traversing a matrix
Largest Number formed from an Array

2) **String**
Reverse words in a given string
Permutations of a given string
Longest Palindrome in a String
Recursively remove all adjacent duplicates
Check if string is rotated by two places
Roman Number to Integer
Anagram
Remove Duplicates
Form a Palindrome
Longest Distinct Characters in the string
Implement Atoi
Implement strstr
Longest Common Prefix

3) **Linked List**
Finding middle element in a linked list
Reverse a linked list
Rotate a Linked List
Reverse a Linked List in groups of given size
Intersection point in Y shaped linked lists
Detect Loop in linked list
Remove loop in Linked List
n’th node from end of linked list
Flattening a Linked List
Merge two sorted linked lists
Intersection point of two Linked Lists

Pairwise swap of a linked list
Add two numbers represented by linked lists
Check if Linked List is Palindrome
Implement Queue using Linked List
Implement Stack using Linked List
Given a linked list of 0s, 1s and 2s, sort it
Delete without head pointer

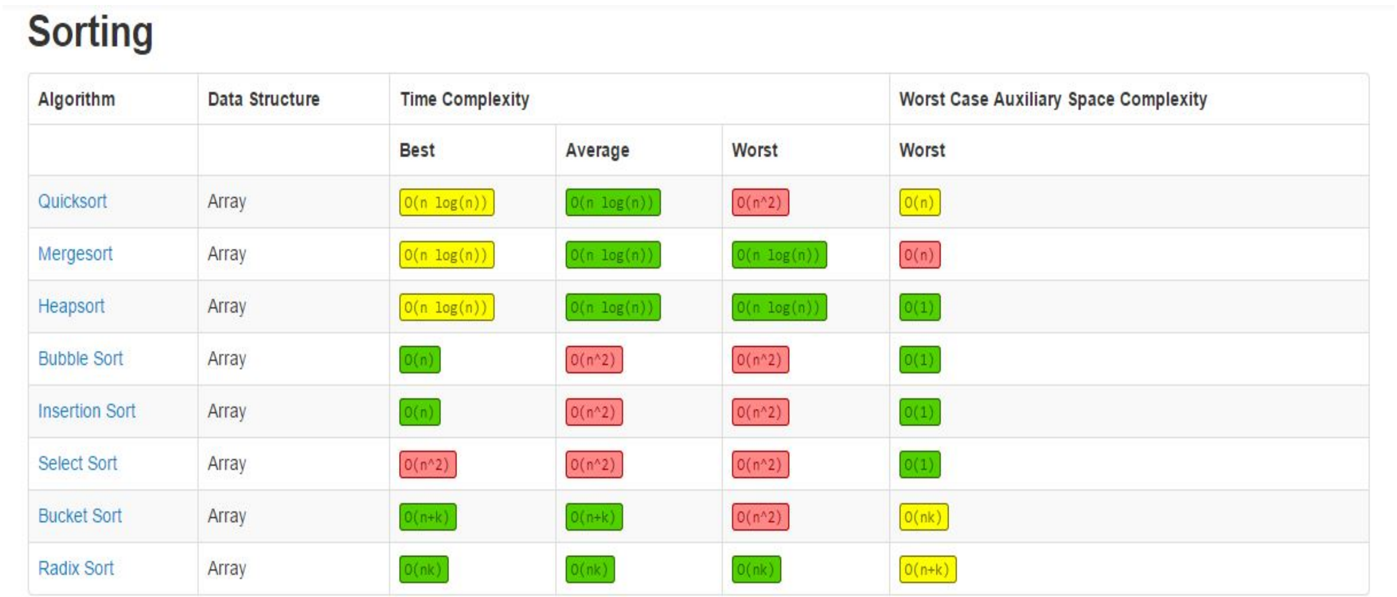
4) **Stack and Queue**
Parenthesis Checker
Next larger element
Queue using two Stacks
Stack using two queues
Get minimum element from stack
LRU Cache
Circular tour
First non-repeating character in a stream
Rotten Oranges
Maximum of all subarrays of size k

5) **Tree**
Print Left View of Binary Tree
Check for BST
Print Bottom View of Binary Tree
Print a Binary Tree in Vertical Order
Level order traversal in spiral form
Connect Nodes at Same Level
Lowest Common Ancestor in a BST
Convert a given Binary Tree to Doubly Linked List
Write Code to Determine if Two Trees are Identical or Not
Given a binary tree, check whether it is a mirror of itself
Height of Binary Tree
Maximum Path Sum
Diameter of a Binary Tree
Number of leaf nodes
Check if given Binary Tree is Height Balanced or Not
Serialize and Deserialize a Binary Tree

6) **Heap**
Find median in a stream
Heap Sort
Operations on Binary Min Heap
Rearrange characters
Merge K sorted linked lists

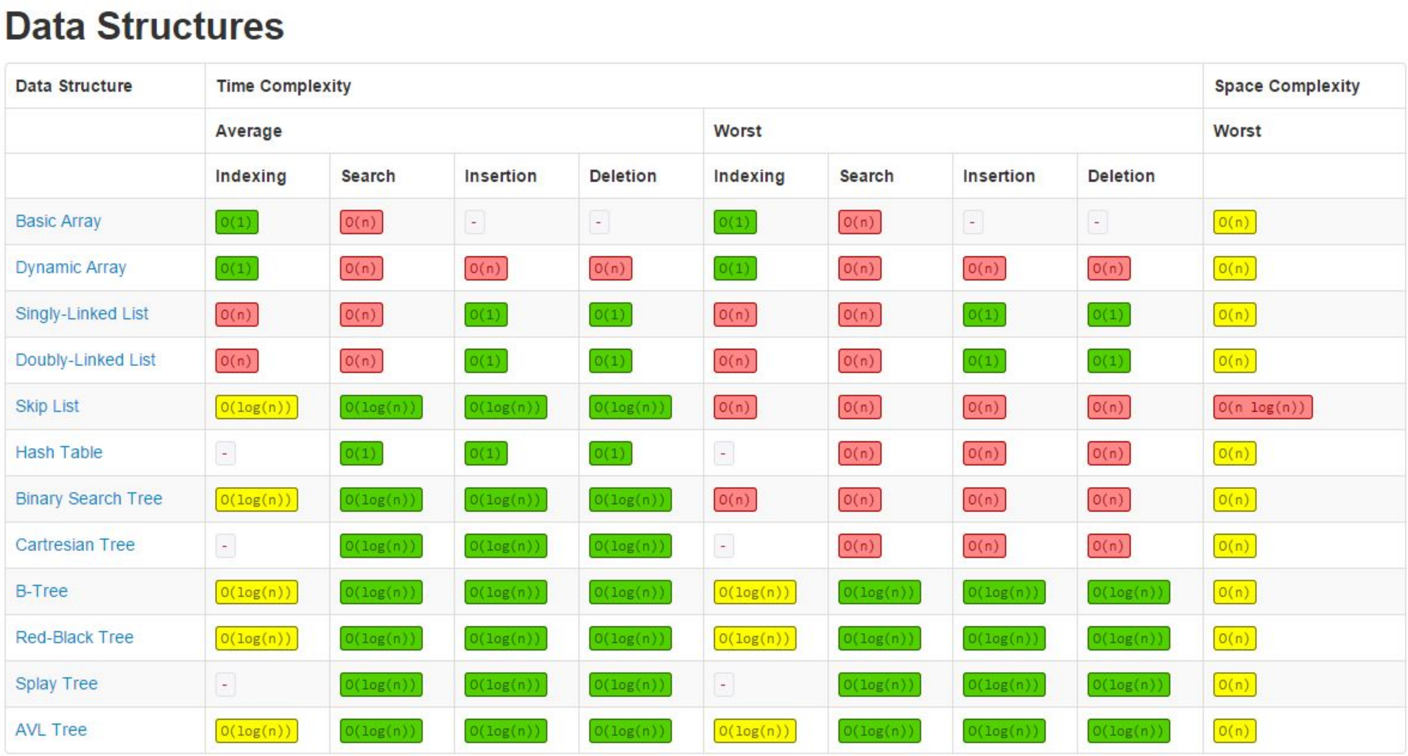
Kth largest element in a stream

7) **Recursion**
Flood fill Algorithm
Number of paths
Combination Sum – Part 2
Special Keyboard
Josephus problem



Sorting

Algorithm	Data Structure	Time Complexity			Worst Case Auxiliary Space Complexity
		Best	Average	Worst	Worst
Quicksort	Array	O(n log(n))	O(n log(n))	O(n^2)	O(n)
Mergesort	Array	O(n log(n))	O(n log(n))	O(n log(n))	O(n)
Heapsort	Array	O(n log(n))	O(n log(n))	O(n log(n))	O(1)
Bubble Sort	Array	O(n)	O(n^2)	O(n^2)	O(1)
Insertion Sort	Array	O(n)	O(n^2)	O(n^2)	O(1)
Select Sort	Array	O(n^2)	O(n^2)	O(n^2)	O(1)
Bucket Sort	Array	O(n+k)	O(n+k)	O(n^2)	O(nk)
Radix Sort	Array	O(nk)	O(nk)	O(nk)	O(n+k)



Data Structures

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Indexing	Search	Insertion	Deletion	Indexing	Search	Insertion	Deletion	
Basic Array	O(1)	O(n)	-	-	O(1)	O(n)	-	-	O(n)
Dynamic Array	O(1)	O(n)	O(n)	O(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Singly-Linked List	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table	-	O(1)	O(1)	O(1)	-	O(n)	O(n)	O(n)	O(n)
Binary Search Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartresian Tree	-	O(log(n))	O(log(n))	O(log(n))	-	O(n)	O(n)	O(n)	O(n)
B-Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Red-Black Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Splay Tree	-	O(log(n))	O(log(n))	O(log(n))	-	O(log(n))	O(log(n))	O(log(n))	O(n)
AVL Tree	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)