

Visit [tenowl.com](https://tenowl.com) for Interview Tips and Placement Materials



## STRIVER SDE SHEET

### Day-1

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	1	Set Matrix Zeros	Coming Soon	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	2	Pascal Triangle	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	3.	Next Permutation	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	4.	Kadane's Algorithm	Coming Soon	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	5.	Sort an array of 0's 1's 2's	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	6.	Stock Buy and Sell	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

### Day-2

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	1	Rotate Matrix	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	2	Merge Overlapping Subintervals	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	3.	Merge two sorted Arrays without extra space	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	4.	Find the duplicate in an array of N+1 integers.	Coming Soon	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	5.	Repeat and Missing Number	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	6.	Inversion of Array (Pre-req: Merge Sort)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

### Day-3

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	1	Search in a 2d Matrix	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	2	Pow(X,n)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	3.	Majority Element (>N/2 times)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	4.	Majority Element (>N/3 times)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	5.	Grid Unique Paths	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	6.	Reverse Pairs (Leetcode)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

### Day-4

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
	1	2-Sum-Problem	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	2	4-sum-Problem	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	3.	Longest Consecutive Sequence	Coming Soon	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	4.	Largest Subarray with 0 sum	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
5.	Count number of subarrays with given Xor K	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	Longest Substring without repeat	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

### Day-5: Linked List

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓ 1	Reverse a LinkedList	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 2	Find the middle of LinkedList	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	Merge two sorted Linked List (use method used in mergeSort)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 4.	Remove N-th node from back of LinkedList	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 5.	Add two numbers as LinkedList	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 6.	Delete a given Node when a node is given. (0(1) solution)	Coming Soon	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

### Day-6

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
✓ 1	Find intersection point of Y LinkedList	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 2	Detect a cycle in Linked List	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 3.	Reverse a LinkedList in groups of size k.	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 4.	Check if a LinkedList is palindrome or not.	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

	Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
✓	5.	Find the starting point of the Loop of LinkedList	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	6.	Flattening of a LinkedList	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	7.	Rotate a LinkedList	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-7

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
	1	Clone a Linked List with random and next pointer		<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	2	3 sum	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	3.	Trapping rainwater	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	4.	Remove Duplicate from Sorted array	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	5.	Max consecutive ones	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-8

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
	1	N meeting in one room		<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	2	Minimum number of platforms required for a railway	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	3.	Job sequencing Problem		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	4.	Fractional Knapsack Problem		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	5.	Greedy algorithm to find minimum number of coins		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	6.	Activity Selection (it is the same as N meeting in one room)		<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-9: Recursion

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓ 1	Subset Sums	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 2	Subset-II		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 3.	Combination sum-1	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 4.	Combination sum-2		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 5.	Palindrome Partitioning	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	K-th permutation Sequence	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-10 : Recursion & Backtracking

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
✓ 1	Print all permutations of a string/array	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 2	N queens Problem	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 3.	Sudoku Solver	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 4.	M coloring Problem	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓ 5.	Rat in a Maze	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	Word Break (print all ways)	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-11: Binary Search

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
1	The N-th root of an integer	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
2	Matrix Median		<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	Find the element that appears once in a sorted array, and the rest element appears twice (Binary search)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	Search element in a sorted and rotated array/ find pivot where it is rotated	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
5.	Median of 2 sorted arrays	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	K-th element of two sorted arrays	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
7.	Allocate Minimum Number of Pages	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
8.	Aggressive Cows	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>

**Day-12: TRIES (Can be done at Last, but still a very very important topic) Watch this playlist -> [Link](#)**

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
1	Implement Trie (Prefix Tree)	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
2	Implement Trie – 2 (Prefix Tree)	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	Longest String with All Prefixes	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	Number of Distinct Substrings in a String	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	Power Set (this is very important)	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
5.	Maximum XOR of two numbers in an array	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	Maximum XOR With an Element From Array	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>

**Day-13 : (Stack and Queue)**

Q.No	Problem Name	Solution	Problem Link	Video Solution	C++ Code	Java Code
1	Implement Stack Using Arrays	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
2	Implement Queue Using Arrays	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	Implement Stack using Queue (using single queue)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	Implement Queue using Stack (O(1) amortized method)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
5.	Check for balanced parentheses	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>

Q.No	Problem Name	Solution	Problem Link	Video Solution	C++ Code	Java Code
6.	Next Greater Element	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
7.	Sort a Stack		<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code

## Day-14 :

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
1	Next Smaller Element		<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
2	LRU cache (IMPORTANT)	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
3.	LFU Cache		<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
4.	Largest rectangle in a histogram	<a href="#">Solution</a>	<a href="#">Click</a>	Two-Pass: <a href="#">Youtube</a>  One Pass: <a href="#">Youtube</a>	Code	Code
5.	Sliding Window maximum	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
6.	Implement Min Stack	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
7.	Rotten Orange (Using BFS)	<a href="#">Solution</a>	<a href="#">Click</a>			
8.	Stock Span Problem		<a href="#">Click</a>			
9.	Find the maximum of minimums of every window size		<a href="#">Click</a>			
10.	The Celebrity Problem		<a href="#">Click</a>			

## Day-15: String

	Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
✓	1	Reverse Words in a String	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
	2	Longest Palindrome in a string	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
✓	3.	Roman Number to Integer and vice versa	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code
✓	4.	Implement ATOI/STRSTR	<a href="#">Click</a>	<a href="#">Youtube</a>	Code	Code

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
5.	Longest Common Prefix	<a href="#">Click</a>	Youtube	<a href="#">Code</a>	<a href="#">Code</a>
6.	Rabin Karp	<a href="#">Click</a>	Youtube	<a href="#">Code</a>	<a href="#">Code</a>

## Day-16: String [Continued]

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
1	Z-Function	<a href="#">Click</a>	Youtube	<a href="#">Code</a>	<a href="#">Code</a>
2	KMP algo / LPS(pi) array	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
3.	Minimum characters needed to be inserted in the beginning to make it palindromic	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
✓ 4.	Check for Anagrams	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
5.	Count and Say	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
6.	Compare version numbers	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>

## Day-17: Binary Tree (Introduction)

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
1	Inorder Traversal	<a href="#">Morris Traversal Iterative/Recursive</a>	<a href="#">Click</a>	<a href="#">Youtube (Recursive)</a> <a href="#">Youtube (Iterative)</a> <a href="#">Youtube (Morris Traversal)</a>	<a href="#">Code (Recursive)</a> <a href="#">Code (Iterative)</a>	<a href="#">Code (Recursive)</a> <a href="#">Code (Iterative)</a> <a href="#">Code (Morris)</a>
2	Preorder Traversal	<a href="#">Morris Traversal Solution</a>	<a href="#">Click</a>	<a href="#">YouTube (Morris Traversal)</a>	<a href="#">Code (Morris)</a>	<a href="#">Code</a>
✓ 3.	Postorder Traversal	<a href="#">Solution</a>	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
✓ 4.	LeftView Of Binary Tree	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>



	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	5.	Bottom View of Binary Tree	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	6.	Top View of Binary Tree	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	7.	Preorder inorder postorder in a single traversal	<a href="#">Solution</a>				
✓	8.	Vertical order traversal	<a href="#">Solution</a>				

### Day-18: Binary Tree [Continued]

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	1	Level order Traversal / Level order traversal in spiral form	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	2	Height of a Binary Tree	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	3.	Diameter of Binary Tree	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	4.	Check if the Binary tree is height-balanced or not	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	5.	LCA in Binary Tree	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	6.	Check if two trees are identical or not	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	7.	Zig Zag Traversal of Binary Tree	<a href="#">Solution</a>				
✓	8.	Boundary Traversal of Binary Tree	<a href="#">Solution</a>				

### Day-19: Binary Tree [Continued]

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
--	------	--------------	-------------------	--------------	----------------	----------	-----------

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	1	Maximum path sum	<a href="#">Solution</a>	<a href="#">Click</a>	Youtube	<a href="#">Code</a>	<a href="#">Code</a>
✓	2	Construct Binary Tree from inorder and preorder		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
✓	3.	Construct Binary Tree from Inorder and Postorder		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
✓	4.	Symmetric Binary Tree	<a href="#">Solution</a>	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
✓	5.	Flatten Binary Tree to LinkedList		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
✓	6.	Check if Binary Tree is the mirror of itself or not		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>

## Day-20: Binary Search Tree

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
	1	Populate Next Right pointers of Tree		<a href="#">Click</a>	Youtube	<a href="#">Code</a>	<a href="#">Code</a>
	2	Search given Key in BST		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
	3.	Construct BST from given keys		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
	4.	Check is a BT is BST or not		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
	5.	Find LCA of two nodes in BST		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
	6.	Find the inorder predecessor/successor of a given Key in BST.		<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>

## Day-21: Binary Search Tree [Continued]

	Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
✓	1.	Floor in a BST		<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
✓	2.	Ceil in a BST		<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
	3.	Find K-th smallest element in BST	<a href="#">Solution</a>	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>
	4.	Find K-th largest element in BST	<a href="#">Solution</a>	<a href="#">Click</a>	YouTube	<a href="#">Code</a>	<a href="#">Code</a>

Q.No	Problem Name	Detailed Solution	Problem Link	Video Solution	C++ Code	Java Code
5.	Find a pair with a given sum in BST	<a href="#">Solution</a>	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	BST iterator		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
7.	Size of the largest BST in a Binary Tree		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
8.	Serialize and deserialize Binary Tree		<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-22: Trees [Miscellaneous]

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
1	Binary Tree to Double Linked List	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
2	Find median in a stream of running integers.	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	K-th largest element in a stream.	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	Distinct numbers in Window.	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
5.	K-th largest element in an unsorted array.	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	Flood-fill Algorithm	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-23: Graphs – Part 1

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
1	Clone a graph (Not that easy as it looks)	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
2	DFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	BFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	Detect A cycle in Undirected Graph using BFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
5.	Detect A cycle in Undirected Graph using DFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	Detect A cycle in a Directed Graph using DFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
7.	Detect A cycle in a Directed Graph using BFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
8.	Topological Sort	<a href="#">Click</a>	DFS: <a href="#">YouTube</a> BFS: <a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
9.	Number of islands (Do in Grid and Graph both)	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
10.	Bipartite Check using BFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
11.	Bipartite Check using DFS	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-24: Graphs – Part 2

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
1	Strongly Connected Component(using KosaRaju's algo)	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
2	Dijkstra's Algorithm	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	Bellman-Ford Algo	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	Floyd Warshall Algorithm	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
5.	MST using Prim's Algo	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
6.	MST using Kruskal's Algo	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

## Day-25: Dynamic Programming – Part 1

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
1	Max Product Subarray	<a href="#">Click</a>	<a href="#">Youtube</a>	<a href="#">Code</a>	<a href="#">Code</a>
2	Longest Increasing Subsequence	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
3.	Longest Common Subsequence	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
4.	0-1 Knapsack	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>
5.	Edit Distance	<a href="#">Click</a>	<a href="#">YouTube</a>	<a href="#">Code</a>	<a href="#">Code</a>

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
6.	Maximum sum increasing subsequence	<a href="#">Click</a>	YouTube	Code	Code
7.	Matrix Chain Multiplication	<a href="#">Click</a>	YouTube	Code	Code

## Day-26: Dynamic Programming – Part 2

Q.No	Problem Name	Problem Link	Video Solution	C++ Code	Java Code
1	Maximum sum path in the matrix, (count paths and similar type do, also backtrack to find the maximum path)	<a href="#">Click</a>	Youtube	Code	Code
2	Coin change	<a href="#">Click</a>	YouTube	Code	Code
3.	Subset Sum	<a href="#">Click</a>	YouTube	Code	Code
4.	Rod Cutting	<a href="#">Click</a>	YouTube	Code	Code
5.	Egg Dropping	<a href="#">Click</a>	YouTube	Code	Code
6.	Word Break	<a href="#">Click</a>	YouTube	Code	Code
7.	Palindrome Partitioning (MCM Variation)	<a href="#">Click</a>	Youtube	Code	Code
8.	Maximum profit in Job scheduling	<a href="#">Click</a>	Youtube	Code	Code

## Day-27:

1. Revise OS notes that you would have made during your sem
2. If not made notes, spend 2 or 3 days and make notes from Knowledge Gate.

## Day-28:

1. Revise DBMS notes that you would have made during your semesters.
2. If not made notes, spend 2 or 3 days and make notes from Knowledge Gate.

## Day-29:

1. Revise CN notes, that you would have made during your sem.

2. If not made notes, spend 2 or 3 days and make notes from Knowledge Gate.

### **Day-30:**

1. Make a note of how will you represent your projects, and prepare all questions related to tech which you have used in your projects. Prepare a note which you can say for 3-10 minutes when he asks you that say something about the project.

*Hurrah!! You are ready for your placement after a month of hard work without a cheat day.*

*~Striver*

Visit [tenowl.com](https://tenowl.com) for Interview Tips and Placement Materials