

Topic-Wise LeetCode Problem List for CSE Interviews in Bangladesh

May 2025

This document provides a curated list of approximately 30 LeetCode problems per topic, tailored for Computer Science and Engineering (CSE) graduates preparing for technical interviews in Bangladesh. Each problem is hyperlinked to its LeetCode page for easy access. The topics are prioritized based on their frequency in interviews at companies like Samsung RD Bangladesh, Brain Station 23, and startups like Pathao. Follow the roadmap: Arrays and Strings, Hash Tables, Linked Lists, Stacks and Queues, Trees and Binary Search Trees, Graphs, and Dynamic Programming.

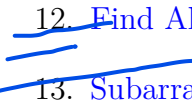
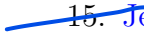

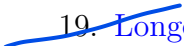

1 Arrays and Strings

1. [Two Sum](#)
2. [Best Time to Buy and Sell Stock](#)
3. [Contains Duplicate](#)
4. [Product of Array Except Self](#)
5. [Maximum Subarray](#)
6. [Maximum Product Subarray](#) Extended form of kene's algorithm
7. [Find Minimum in Rotated Sorted Array](#)
8. [Search in Rotated Sorted Array](#)
9. [3Sum](#)
10. [Container With Most Water](#)
11. [Longest Substring Without Repeating Characters](#)
12. [Longest Palindromic Substring](#)
13. [String to Integer \(atoi\)](#)

14. Regular Expression Matching
15. Merge Intervals
16. Group Anagrams
- ~~17. Valid Anagram~~
18. Valid Parentheses
19. Longest Common Prefix
- ~~20. Reverse String~~
- ~~21. First Unique Character in a String~~
22. Implement strStr()
- ~~23. Count and Say~~
- ~~24. Rotate Array~~
- ~~25. Move Zeroes~~
- ~~26. Two Sum II - Input Array Is Sorted~~
- ~~27. Minimum Size Subarray Sum~~
- ~~28. Majority Element~~
- ~~29. Next Permutation~~
- ~~30. Subarray Sum Equals K~~

2 Hash Tables

- ~~1. Two Sum~~
- ~~2. Contains Duplicate~~
- ~~3. Single Number~~
- ~~4. Intersection of Two Arrays~~
- ~~5. Happy Number~~
- ~~6. Valid Anagram~~
- ~~7. Group Anagrams~~
- ~~8. Top K Frequent Elements~~

- 
- 
- 
- 
- 
- 
9. Longest Consecutive Sequence
 10. Valid Sudoku
 11. Isomorphic Strings
 - ~~12. Find All Anagrams in a String~~
 - ~~13. Subarray Sum Equals K~~
 14. Minimum Window Substring
 - ~~15. Jewels and Stones~~
 16. Design HashMap
 17. Design HashSet
 - ~~18. First Unique Character in a String~~
 - ~~19. Longest Substring Without Repeating Characters~~
 20. Four Sum II
 21. Sort Characters By Frequency
 22. Minimum Index Sum of Two Lists
 23. Longest Word in Dictionary
 24. Find Duplicate Subtrees
 25. Number of Boomerangs
 - ~~26. Find All Duplicates in an Array~~
 27. Contiguous Array
 28. Longest Palindromic Substring
 29. K-diff Pairs in an Array
 30. Subarray Sums Divisible by K

3 Linked Lists

1. Reverse Linked List
2. Merge Two Sorted Lists
3. Palindrome Linked List

4. [Remove Nth Node From End of List](#)
5. [Linked List Cycle](#)
6. [Linked List Cycle II](#)
7. [Add Two Numbers](#)
8. [Merge k Sorted Lists](#)
9. [Swap Nodes in Pairs](#)
10. [Rotate List](#)
11. [Remove Duplicates from Sorted List](#)
12. [Remove Duplicates from Sorted List II](#)
13. [Partition List](#)
14. [Reverse Linked List II](#)
15. [Middle of the Linked List](#)
16. [Intersection of Two Linked Lists](#)
17. [Odd Even Linked List](#)
18. [Delete Node in a Linked List](#)
19. [Split Linked List in Parts](#)
20. [Next Greater Node In Linked List](#)
21. [Copy List with Random Pointer](#)
22. [Flatten a Multilevel Doubly Linked List](#)
23. [Insert into a Sorted Circular Linked List](#)
24. [Remove Linked List Elements](#)
25. [Reverse Nodes in k-Group](#)
26. [Add Two Numbers II](#)
27. [Sort List](#)
28. [Swapping Nodes in a Linked List](#)
29. [Design Linked List](#)
30. [Merge Nodes in Between Zeros](#)

4 Stacks and Queues

1. [Valid Parentheses](#)
2. [Min Stack](#)
3. [Implement Stack using Queues](#)
4. [Implement Queue using Stacks](#)
5. [Next Greater Element I](#)
6. [Next Greater Element II](#)
7. [Daily Temperatures](#)
8. [Evaluate Reverse Polish Notation](#)
9. [Simplify Path](#)
10. [Basic Calculator](#)
11. [Basic Calculator II](#)
12. [Remove All Adjacent Duplicates In String](#)
13. [Remove All Adjacent Duplicates in String II](#)
14. [Asteroid Collision](#)
15. [Decode String](#)
16. [Backspace String Compare](#)
17. [Design Circular Queue](#)
18. [Trapping Rain Water](#)
19. [Largest Rectangle in Histogram](#)
20. [Online Stock Span](#)
21. [Score of Parentheses](#)
22. [Minimum Remove to Make Valid Parentheses](#)
23. [Design a Stack With Increment Operation](#)
24. [Number of Recent Calls](#)
25. [Valid Stack Sequences](#)
26. [Make The String Great](#)

27. [Crawler Log Folder](#)
28. [Baseball Game](#)
29. [Maximum Frequency Stack](#)
30. [Remove Duplicate Letters](#)

5 Trees and Binary Search Trees

1. [Maximum Depth of Binary Tree](#)
2. [Validate Binary Search Tree](#)
3. [Symmetric Tree](#)
4. [Binary Tree Level Order Traversal](#)
5. [Binary Tree Inorder Traversal](#)
6. [Binary Tree Preorder Traversal](#)
7. [Binary Tree Postorder Traversal](#)
8. [Convert Sorted Array to Binary Search Tree](#)
9. [Lowest Common Ancestor of a Binary Search Tree](#)
10. [Path Sum](#)
11. [Path Sum II](#)
12. [Minimum Depth of Binary Tree](#)
13. [Balanced Binary Tree](#)
14. [Same Tree](#)
15. [Invert Binary Tree](#)
16. [Binary Tree Zigzag Level Order Traversal](#)
17. [Construct Binary Tree from Preorder and Inorder Traversal](#)
18. [Construct Binary Tree from Inorder and Postorder Traversal](#)
19. [Kth Smallest Element in a BST](#)
20. [Sum of Left Leaves](#)
21. [Binary Tree Right Side View](#)

22. [Count Complete Tree Nodes](#)
23. [Average of Levels in Binary Tree](#)
24. [Find Bottom Left Tree Value](#)
25. [Diameter of Binary Tree](#)
26. [Merge Two Binary Trees](#)
27. [Find Mode in Binary Search Tree](#)
28. [Second Minimum Node In a Binary Tree](#)
29. [Binary Search Tree Iterator](#)
30. [Subtree of Another Tree](#)

6 Graphs

1. [Number of Islands](#)
2. [Course Schedule](#)
3. [Course Schedule II](#)
4. [Clone Graph](#)
5. [Pacific Atlantic Water Flow](#)
6. [Rotting Oranges](#)
7. [Word Ladder](#)
8. [Word Ladder II](#)
9. [Surrounded Regions](#)
10. [Graph Valid Tree](#)
11. [Number of Connected Components in an Undirected Graph](#)
12. [Find the Town Judge](#)
13. [Accounts Merge](#)
14. [Minimum Height Trees](#)
15. [Flood Fill](#)
16. [Snakes and Ladders](#)

17. [All Paths From Source to Target](#)
18. [Keys and Rooms](#)
19. [Reconstruct Itinerary](#)
20. [Evaluate Division](#)
21. [Network Delay Time](#)
22. [Cheapest Flights Within K Stops](#)
23. [Find if Path Exists in Graph](#)
24. [Nearest Exit from Entrance in Maze](#)
25. [Shortest Path in Binary Matrix](#)
26. [Is Graph Bipartite?](#)
27. [Possible Bipartition](#)
28. [Detonate the Maximum Bombs](#)
29. [Redundant Connection](#)
30. [Time Needed to Inform All Employees](#)

7 Dynamic Programming

1. [Climbing Stairs](#)
2. [House Robber](#)
3. [House Robber II](#)
4. [Min Cost Climbing Stairs](#)
5. [Longest Increasing Subsequence](#)
6. [Longest Common Subsequence](#)
7. [Coin Change](#)
8. [Coin Change 2](#)
9. [Maximum Product Subarray](#)
10. [Palindromic Substrings](#)
11. [Decode Ways](#)

12. [Unique Paths](#)
13. [Unique Paths II](#)
14. [Minimum Path Sum](#)
15. [Partition Equal Subset Sum](#)
16. [Best Time to Buy and Sell Stock with Cooldown](#)
17. [Word Break](#)
18. [Word Break II](#)
19. [Longest Valid Parentheses](#)
20. [Triangle](#)
21. [Maximum Subarray](#)
22. [Jump Game](#)
23. [Jump Game II](#)
24. [Combination Sum IV](#)
25. [Target Sum](#)
26. [Interleaving String](#)
27. [Edit Distance](#)
28. [Distinct Subsequences](#)
29. [Arithmetic Slices](#)

Note

The 0/1 Knapsack problem is not available on LeetCode but is commonly tested in Bangladesh interviews. Practice it on platforms like GeeksforGeeks: [0/1 Knapsack Problem](#).

Study Plan

- **Weeks 1–3:** Arrays and Strings (solve 3–5 problems daily).
- **Weeks 4–5:** Hash Tables.
- **Weeks 6–7:** Linked Lists.
- **Week 8:** Stacks and Queues.

- **Weeks 9–10:** Trees and Binary Search Trees.
- **Weeks 11–13:** Graphs.
- **Weeks 14–16:** Dynamic Programming.
- Spend 30–45 minutes per problem, reviewing solutions if stuck after 20 minutes. Use LeetCode's Explore section for topic-specific practice.

Final Notes

These problems are curated based on their relevance to technical interviews in Bangladesh, focusing on easy-to-medium difficulty to match your skill level (20–22 mid-level problems solved). They cover key concepts tested by companies like Samsung RD Bangladesh, Brain Station 23, and startups like Pathao. Track progress using a spreadsheet or Notion, and practice mock interviews on platforms like Pramp. For additional resources, explore local platforms like 10 Minute School or YouTube channels like Anisul Islam.