### Array

1. Find the missing number in a given integer array of 1 to 100.
2. Find the duplicate number on a given integer array.
3. Find the largest and smallest number in an unsorted integer array.
4. Find all pairs of an integer array whose sum is equal to a given number.
5. Find duplicate numbers in an array containing multiple duplicates.
6. Remove duplicates from an array in place.
7. Reverse an array in place.
8. Find the intersection of two arrays.
9. Find the maximum product of two integers in an array.
10. Find the minimum value in a rotated sorted array.
11. Find the maximum sum subarray (Kadane’s algorithm).
12. Rotate an array by k positions.
13. Find the subarray with the given sum.
14. Merge two sorted arrays.
15. Move all zeros to the end of an array.
16. Find the majority element in an array using the Boyer-Moore voting algorithm.
17. Find the equilibrium index of an array.
18. Rearrange an array such that even index elements are smaller and odd index elements are greater.
19. Find the largest subarray with equal number of 0s and 1s.
20. Find the smallest subarray with a sum greater than a given value.

### Strings

21. Reverse a string in place.
22. Check if two strings are anagrams of each other.
23. Find all permutations of a string.
24. Find the first non-repeating character in a string.
25. Count the occurrence of a given character in a string.
26. Remove duplicate characters from a string.
27. Check if a string is a palindrome.
28. Convert a given string into an integer.
29. Find the longest substring without repeating characters.
30. Find the longest common prefix.
31. Convert a given string into its integer equivalent (atoi).
32. Check if a string is a valid shuffle of two distinct strings.
33. Implement strstr() function to find the first occurrence of a substring.
34. Find the longest repeating subsequence in a string.
35. Remove all adjacent duplicates in a string.
36. Check if one string is a rotation of another.
37. Find the smallest window in a string containing all characters of another string.
38. Find the longest palindromic substring.
39. Convert a string to title case (first letter of each word in uppercase).
40. Count and say the sequence for a given number.

### Linked List

41. Reverse a linked list.
42. Detect a cycle in a linked list.
43. Find the middle element of a linked list.
44. Remove the nth node from the end of a linked list.
45. Merge two sorted linked lists.
46. Remove duplicates from a linked list.
47. Find the intersection point of two linked lists.
48. Detect and remove a loop in a linked list.
49. Add two numbers represented by linked lists.
50. Check if a linked list is a palindrome.
51. Implement a singly linked list.
52. Implement a doubly linked list.
53. Delete a node in a singly linked list given only access to that node.
54. Find the intersection point of two linked lists using two pointers.
55. Split a linked list into two equal halves.
56. Sort a linked list using merge sort.
57. Flatten a multilevel linked list.
58. Remove nodes from a linked list that have a greater value on the right.
59. Clone a linked list with next and random pointer.
60. Implement a skip list.

### Stack and Queue

61. Implement a stack using an array/linked list.
62. Implement a queue using an array/linked list.
63. Implement a stack using two queues.
64. Implement a queue using two stacks.
65. Sort a stack.
66. Evaluate a postfix expression using a stack.
67. Implement a priority queue.
68. Check for balanced parentheses in an expression.
69. Find the maximum element in a stack in O(1) time.
70. Implement a circular queue.
71. Design a stack that supports getMin() in O(1) time.
72. Implement efficient k stacks in a single array.
73. Implement an LFU cache.
74. Generate binary numbers from 1 to n using a queue.
75. Reverse the first k elements of a queue.
76. Check if all the leaves are at the same level in a binary tree using a queue.
77. Implement a monotonic queue.
78. Check for redundancy in an arithmetic expression.
79. Design a stack that supports push, pop, top, and retrieving the maximum element in O(1) time.
80. Implement a deque using a doubly linked list.

### Tree

81. Traverse a binary tree in preorder, inorder, and postorder.
82. Implement a binary search tree (BST).
83. Find the height of a binary tree.
84. Find the lowest common ancestor (LCA) of two nodes in a binary tree.
85. Check if a binary tree is balanced.
86. Check if two binary trees are identical.
87. Find the maximum depth of a binary tree.
88. Find the diameter of a binary tree.
89. Convert a binary tree to a doubly linked list.
90. Print all nodes at a distance k from a given node in a binary tree.
91. Convert a binary search tree to a balanced binary search tree.
92. Print the nodes of a binary tree in a vertical order.
93. Find the level of a node in a binary tree.
94. Connect nodes at the same level in a binary tree.
95. Construct a binary tree from its preorder and inorder traversal.
96. Construct a binary tree from its inorder and postorder traversal.
97. Serialize and deserialize a binary tree.
98. Find the inorder successor in a binary search tree.
99. Check if a binary tree is a subtree of another binary tree.
100. Find the sum of all left leaves in a binary tree.

### Graph

101. Implement depth-first search (DFS) and breadth-first search (BFS).
102. Find the shortest path in an unweighted graph.
103. Detect a cycle in a directed/undirected graph.
104. Find strongly connected components in a graph.
105. Implement Dijkstra’s algorithm.
106. Implement Prim’s algorithm.
107. Implement Kruskal’s algorithm.
108. Find the shortest path in a weighted graph.
109. Check if a graph is bipartite.
110. Find the number of islands in a given 2D matrix.
111. Implement the Floyd-Warshall algorithm.
112. Detect a negative cycle in a graph.
113. Find articulation points in a graph.
114. Find bridges in a graph.

115. Find the mother vertex in a graph.
116. Topologically sort a graph.
117. Implement the Bellman-Ford algorithm.
118. Implement the A* search algorithm.
119. Find the transitive closure of a graph.
120. Implement Kahn’s algorithm for topological sorting.

### Sorting and Searching
121. Implement quicksort.
122. Implement mergesort.
123. Implement heapsort.
124. Implement binary search.
125. Find the kth largest element in an unsorted array.
126. Search for a given number in a rotated sorted array.
127. Count the number of occurrences of a given number in a sorted array.
128. Search for a range in a sorted array.
129. Sort an array with many duplicated values.
130. Find the median of two sorted arrays.
131. Find the kth smallest element in a binary search tree.
132. Sort an array using heapsort.
133. Sort a nearly sorted (or K sorted) array.
134. Find the fixed point in a given array.
135. Perform an exponential search.
136. Implement the interpolation search.
137. Search in a row-wise and column-wise sorted matrix.
138. Count inversions in an array.
139. Find the peak element in an array.
140. Find the frequency of an element in a sorted array.

### Dynamic Programming
141. Implement the Fibonacci sequence using dynamic programming.
142. Solve the Knapsack problem.
143. Solve the Longest Common Subsequence problem.
144. Solve the Longest Increasing Subsequence problem.
145. Solve the Edit Distance problem.
146. Solve the Coin Change problem.
147. Solve the Maximum Subarray problem.
148. Solve the Egg Dropping problem.
149. Solve the 0/1 Knapsack problem.
150. Solve the Minimum Path Sum problem.
151. Solve the Rod Cutting problem.
152. Solve the Longest Bitonic Subsequence problem.
153. Solve the Matrix Chain Multiplication problem.
154. Solve the Subset Sum problem.
155. Solve the Count of Subsets with a Given Sum problem.
156. Solve the Target Sum problem.
157. Solve the Minimum Number of Insertions to Form a Palindrome problem.
158. Solve the Maximum Product Subarray problem.
159. Solve the Maximum Length of Pair Chain problem.
160. Solve the Word Break problem.

### Hashing
161. Implement a hash table.
162. Design a hash map.
163. Check if two strings are anagrams using a hash table.
164. Find the first non-repeating character using a hash map.
165. Find the longest substring with distinct characters using a hash map.
166. Count the frequency of elements in an array using a hash map.
167. Group anagrams using a hash table.
168. Check for pair sum in an array using hashing.
169. Find the largest subarray with 0 sum using a hash map.
170. Find the longest consecutive subsequence using a hash set.

### Advanced Topics
171. Implement a Trie (Prefix Tree).
172. Find the shortest path in a maze.
173. Solve the N-Queens problem.
174. Implement a Red-Black Tree.
175. Implement an AVL Tree.
176. Find the maximum flow in a flow network using the Ford-Fulkerson algorithm.
177. Implement the Rabin-Karp algorithm for substring search.
178. Implement the KMP algorithm for pattern matching.
179. Solve the Traveling Salesman Problem using dynamic programming.
180. Solve the Graph Coloring problem.

### System Design
181. Design a URL shortening service.
182. Design a cache system.
183. Design a messaging queue.
184. Design a file storage system like Dropbox or Google Drive.
185. Design a search autocomplete system.
186. Design a rate limiter.
187. Design a parking lot system.
188. Design a ride-sharing service like Uber or Lyft.
189. Design a social media feed system.
190. Design a hotel reservation system.

### Algorithm Analysis and Complexity
191. Analyze the time and space complexity of an algorithm.
192. Determine the Big-O notation of an algorithm.
193. Compare the time complexity of different algorithms.
194. Optimize an algorithm for better performance.
195. Identify bottlenecks in an algorithm.
196. Prove the correctness of an algorithm.
197. Find the amortized time complexity of a data structure operation.
198. Analyze the performance of recursive algorithms.
199. Use dynamic programming to optimize a recursive solution.
200. Solve problems using divide and conquer strategies.

### Miscellaneous
201. Design a vending machine.
202. Design an elevator system.
203. Design a library management system.
204. Design a hospital management system.
205. Design a movie ticket booking system.
206. Design an online food ordering system.
207. Design a document editor like Google Docs.
208. Design a photo-sharing application like Instagram.
209. Design a web crawler.
210. Design an email service like Gmail.
211. Implement a least recently used (LRU) cache.
212. Implement a most recently used (MRU) cache.
213. Implement a random access data structure.
214. Design a logging system.
215. Implement a thread-safe bounded blocking queue.
216. Design a file compression system.
217. Design a voting system.
218. Design a fraud detection system.
219. Design a recommendation system.
220. Design a blockchain.

### Coding Challenges
221. Solve the Two Sum problem.
222. Solve the Add Two Numbers problem.
223. Solve the Longest Substring Without Repeating Characters problem.
224. Solve the Median of Two Sorted Arrays problem.
225. Solve the Longest Palindromic Substring problem.
226. Solve the Container With Most Water problem.
227. Solve the Trapping Rain Water problem.
228. Solve the Multiply Strings problem.
229. Solve the Permutations problem.

230. Solve the Rotate Image problem.
231. Solve the Group Anagrams problem.
232. Solve the Maximum Subarray problem.
233. Solve the Spiral Matrix problem.
234. Solve the Jump Game problem.
235. Solve the Merge Intervals problem.
236. Solve the Insert Interval problem.
237. Solve the Subsets problem.
238. Solve the Word Search problem.
239. Solve the Binary Tree Inorder Traversal problem.
240. Solve the Validate Binary Search Tree problem.
241. Solve the Symmetric Tree problem.
242. Solve the Binary Tree Level Order Traversal problem.
243. Solve the Convert Sorted Array to Binary Search Tree problem.
244. Solve the Minimum Depth of Binary Tree problem.
245. Solve the Path Sum problem.
246. Solve the Construct Binary Tree from Preorder and Inorder Traversal problem.
247. Solve the Binary Tree Maximum Path Sum problem.
248. Solve the Longest Consecutive Sequence problem.
249. Solve the Single Number problem.
250. Solve the Linked List Cycle problem.
251. Solve the Reorder List problem.
252. Solve the Maximum Product Subarray problem.
253. Solve the Find Minimum in Rotated Sorted Array problem.
254. Solve the Search in Rotated Sorted Array problem.
255. Solve the Find Peak Element problem.
256. Solve the Word Ladder problem.
257. Solve the Clone Graph problem.
258. Solve the Course Schedule problem.
259. Solve the Implement Trie (Prefix Tree) problem.
260. Solve the Word Search II problem.
261. Solve the Maximal Rectangle problem.
262. Solve the Palindrome Partitioning problem.
263. Solve the Surrounded Regions problem.
264. Solve the Number of Islands problem.
265. Solve the Best Time to Buy and Sell Stock problem.
266. Solve the Best Time to Buy and Sell Stock II problem.
267. Solve the Best Time to Buy and Sell Stock III problem.
268. Solve the Single Number II problem.
269. Solve the Maximum Gap problem.
270. Solve the Factorial Trailing Zeroes problem.
271. Solve the Majority Element problem.
272. Solve the Excel Sheet Column Number problem.
273. Solve the Factorial Trailing Zeroes problem.
274. Solve the Rotate Array problem.
275. Solve the Binary Tree Right Side View problem.
276. Solve the Largest Number problem.
277. Solve the Basic Calculator problem.
278. Solve the Range Sum Query - Immutable problem.
279. Solve the Contains Duplicate problem.
280. Solve the Summary Ranges problem.
281. Solve the House Robber problem.
282. Solve the Implement Stack using Queues problem.
283. Solve the Implement Queue using Stacks problem.
284. Solve the Number of Digit One problem.
285. Solve the Count Primes problem.
286. Solve the Find the Duplicate Number problem.
287. Solve the Ugly Number problem.
288. Solve the Missing Number problem.
289. Solve the Product of Array Except Self problem.
290. Solve the Sliding Window Maximum problem.
291. Solve the Kth Largest Element in an Array problem.
292. Solve the Coin Change problem.
293. Solve the Combination Sum problem.
294. Solve the Permutation Sequence problem.
295. Solve the Remove Nth Node From End of List problem.
296. Solve the Longest Increasing Subsequence problem.
297. Solve the Reconstruct Itinerary problem.