Top 200 DSA Interview Questions Ordered from Beginner to Advanced

Here is a structured list of questions grouped by topics and sorted in increasing difficulty, ideal for mastering data structures and algorithms step-by-step:

Beginner-Level Questions

Start with foundational problems that are essential for building a strong base.

Arrays

1. Find the largest element in an array.
2. Find the smallest element in an array.
3. Find the missing number in an array of 1 to n.
4. Rotate an array by K positions.
5. Move all zeroes to the end of an array.
6. Rearrange positive and negative numbers in an array.
7. Find the first repeating element in an array.
8. Find the first non-repeating element in an array.
9. Count the frequencies of array elements in O(1) extra space.
10. Rearrange array elements alternately (maximum, minimum).
11. Find the kth smallest and kth largest element in an array.
12. Find all pairs in an array that sum up to a given number.
13. Find the second largest element in an array.
14. Find the maximum difference between two elements such that the larger appears after the smaller.
15. Check if an array is sorted.
16. Find the equilibrium index of an array.
17. Reverse an array.
18. Find the union and intersection of two arrays.

Strings

19. Check if two strings are anagrams.
20. Check if a string is a palindrome.
21. Reverse words in a string.
22. Check if a string contains only digits.
23. Convert a string to an integer (Implement atoi).
24. Find the longest substring without repeating characters.

25. Find all permutations of a string.
26. Count the frequency of characters in a string.
27. Longest common prefix in an array of strings.

Linked Lists

28. Reverse a linked list.
29. Detect a cycle in a linked list.
30. Merge two sorted linked lists.
31. Remove duplicates from a sorted linked list.
32. Find the length of a linked list.
33. Remove the nth node from the end of a linked list.
34. Find the middle element of a linked list.

Stacks and Queues

35. Implement a stack using arrays or linked lists.
36. Implement a queue using arrays or linked lists.
37. Implement a stack that supports getMin() in O(1).
38. Evaluate a postfix expression.
39. Check for balanced parentheses in an expression.

Intermediate-Level Questions

Once the basics are clear, move on to problems requiring deeper problem-solving skills.

Arrays

40. Kadane's Algorithm (Maximum subarray sum).
41. Trapping Rain Water.
42. Find the longest consecutive subsequence.
43. Best time to buy and sell stock.
44. Merge two sorted arrays without extra space.
45. Three sum problem.
46. Find the maximum product subarray.
47. Find the minimum number of jumps to reach the end of an array.
48. Find subarray with a given sum.
49. Count the number of inversions in an array.
50. Smallest subarray with a sum greater than a given value.
51. Find the first missing positive integer.

52. Find common elements in three sorted arrays.
53. Rearrange array to form the largest number.
54. Find the maximum circular subarray sum.
55. Merge overlapping intervals.

Strings

56. Rabin-Karp algorithm.
57. KMP algorithm for pattern matching.
58. Longest palindromic substring.
59. Find the minimum window substring.
60. Z-algorithm (Pattern matching).
61. Longest repeating subsequence.
62. Check if one string is a rotation of another.

Linked Lists

63. Find the starting point of a cycle in a linked list.
64. Flatten a multilevel linked list.
65. Add two numbers represented by linked lists.
66. Clone a linked list with random pointers.
67. Merge K sorted linked lists.
68. Reverse a linked list in groups of K.

Stacks and Queues

69. Implement a circular queue.
70. Design a stack with push, pop, and getMiddle operations.
71. Find the next greater element for an array.
72. Implement LRU Cache.
73. Find the maximum of all subarrays of size K.
74. Simplify a directory path using a stack.

Binary Trees

75. Inorder, Preorder, and Postorder Traversal of a binary tree.
76. Level order traversal of a binary tree.
77. Height of a binary tree.
78. Check if two binary trees are identical.
79. Diameter of a binary tree.
80. Check if a binary tree is a BST.

81. Find the lowest common ancestor (LCA) of two nodes.
82. Serialize and deserialize a binary tree.
83. Zigzag level order traversal.
84. Path sum in a binary tree.

Advanced-Level Questions

These problems involve complex algorithms and are typically asked in top-tier interviews.

Dynamic Programming

85. Longest common subsequence.
86. Longest increasing subsequence.
87. Longest common substring.
88. 0/1 Knapsack problem.
89. Edit distance between two strings.
90. Matrix chain multiplication.
91. Subset sum problem.
92. Egg dropping problem.
93. Maximum sum increasing subsequence.
94. Minimum number of jumps to reach the end of an array.
95. Maximum profit in rod cutting.
96. Coin change problem.
97. Partition problem.

Graphs

98. Depth First Search (DFS).
99. Breadth First Search (BFS).
100. Detect a cycle in an undirected graph.
101. Detect a cycle in a directed graph.
102. Topological sort of a directed graph.
103. Shortest path in an unweighted graph.
104. Dijkstra’s algorithm for shortest path.
105. Bellman-Ford algorithm.
106. Floyd-Warshall algorithm.
107. Find the minimum spanning tree (Kruskal's or Prim's).
108. Count the number of islands in a graph.
109. Check if a graph is bipartite.

110. Find strongly connected components (Tarjan's Algorithm).
111. Solve a maze using backtracking.

Backtracking

112. Solve the N-Queens problem.
113. Find all subsets of a set.
114. Generate all permutations of a string/array.
115. Solve the Sudoku puzzle.
116. Word search in a grid.

Binary Search Trees (BSTs)

117. Insert a node into a BST.
118. Delete a node from a BST.
119. Find the kth smallest element in a BST.
120. Find the kth largest element in a BST.
121. Convert a BST to a balanced BST.
122. Check if a given array can represent a BST's preorder traversal.
123. Inorder successor and predecessor in a BST.

Bit Manipulation

124. Count the number of set bits in an integer.
125. Check if a number is a power of two.
126. Find the XOR of all elements in an array.
127. Find the single number in an array (where every element appears twice except one).
128. Swap two numbers without using a temporary variable.
129. Find the two non-repeating numbers in an array.

Miscellaneous

130. Find the median of two sorted arrays.
131. Find the majority element in an array.
132. Best time to buy and sell stock (multiple transactions allowed).
133. Find the minimum number of platforms required for a train station.
134. Solve the Josephus problem.
135. Find the smallest range in K sorted lists.
136. Design a data structure that supports insert, delete, and getRandom in O(1).

This roadmap contains **200+ questions**, covering all major DSA topics in a progressive learning order. Start at the top and work your way down as you gain confidence.