

# Mircrosoft VO

---

1. [1239. Maximum Length of a Concatenated String with Unique Characters -- Medium](#)
2. [234. Palindrome Linked List -- Easy](#) **O(1)**
3. [297. Serialize and Deserialize Binary Tree -- Hard](#)
4. [428. Serialize and Deserialize N-ary Tree -- Hard](#)
5. [449. Serialize and Deserialize BST -- Medium](#)
6. [1188. Design Bounded Blocking Queue -- Medium](#)
7. [103. Binary Tree Zigzag Level Order Traversal -- Medium](#) +
8. [1197. Minimum Knight Moves -- Medium](#)
9. [23. Merge k Sorted Lists -- Hard](#)
10. [Sell Stock Series -- Easy to Hard](#)
11. [398. Random Pick Index -- Medium](#)
12. [287. Find the Duplicate Number -- Medium](#) **follow up** +
13. [146. LRU Cache -- Medium](#) +
14. [49. Group Anagrams -- Medium](#)
15. [535. Encode and Decode TinyURL -- Medium](#)
16. [438. Find All Anagrams in a String -- Medium](#)
17. [273. Integer to English Words -- Hard](#)
18. [138. Copy List with Random Pointer -- Medium](#)
19. [171. Excel Sheet Column Number -- Easy](#)
20. [348. Design Tic-Tac-Toe -- Medium](#)
21. [783. Minimum Distance Between BST Nodes -- Easy](#) **must think of In-order for BST**
22. [247. Strobogrammatic Number II -- Medium](#)
23. [21. Merge Two Sorted Lists -- Easy](#)
24. [347. Top K Frequent Elements -- Medium](#) +
25. [721. Accounts Merge -- Medium](#)
26. [new grad 001](#)
27. [80. Remove Duplicates from Sorted Array II -- Medium](#)
28. [198. House Robber -- Medium](#)
29. [489. Robot Room Cleaner -- Hard](#)
30. [295. Find Median from Data Stream -- Hard](#)
31. [183. Customers Who Never Order -- Easy](#)
32. [33. Search in Rotated Sorted Array -- Medium](#)
33. [445. Add Two Numbers II -- Medium](#)
34. check if it is BST
35. [Min distance in a graph](#)
36. [1446. Consecutive Characters -- Easy](#)

37. [650. 2 Keys Keyboard -- Medium](#)
38. [1344. Angle Between Hands of a Clock -- Medium](#)
39. [13. Roman to Integer -- Easy](#)
40. [36. Valid Sudoku -- Medium](#)
41. [37. Sudoku Solver -- Hard](#)
42. [772. Basic Calculator III -- Hard](#)
43. [53. Maximum Subarray -- Easy](#)
44. [46. Permutations -- Medium](#)
  1. [31. Next Permutation -- Medium](#)
  2. [47. Permutations II](#)
45. [560. Subarray Sum Equals K -- Medium](#)
46. [1586. Binary Search Tree Iterator II -- Medium](#)
47. [127. Word Ladder -- Hard](#)
48. [126. Word Ladder II -- Hard](#)
49. [90. Subsets II -- Medium](#) +
50. [210. Course Schedule II -- Medium](#) check III if have time
51. [207. Course Schedule -- Medium](#)
52. [340. Longest Substring with At Most K Distinct Characters -- Medium](#)
53. [239. Sliding Window Maximum -- Hard](#) +
54. Lock
55. [62. Unique Paths -- Medium](#)
56. int sqrt(x)
57. [56. Merge Intervals -- Medium](#) +
58. [source to destination, minimum cost](#)
59. [463. Island Perimeter -- Easy](#)
60. [468. Validate IP Address](#)
61. [128. Longest Consecutive Sequence -- Hard](#) **follow up**
62. [973. K Closest Points to Origin -- Medium](#)
63. [702. Search in a Sorted Array of Unknown Size](#)
64. [658. Find K Closest Elements -- Medium](#)
65. [15. 3Sum -- Medium](#)
66. [124. Binary Tree Maximum Path Sum -- Hard](#)
67. [Transfer BST to ascending doubly LinkedList](#)
68. [Find if an array of strings can be chained to form a circle](#)
  1. [1048. Longest String Chain -- Medium](#)
69. [1258. Synonymous Sentences -- Medium](#)
70. [54. Spiral Matrix -- Medium](#)
71. [75. Sort Colors -- Medium](#)
72. Given a linkedList, return K largest values in an array
73. [142. Linked List Cycle II -- Medium](#) +

74. [68. Text Justification -- Hard](#)
75. [346. Moving Average from Data Stream -- Easy](#)
76. [4. Median of Two Sorted Arrays -- Hard](#) + / kth smallest element in two sorted list
77. [179. Largest Number](#). e.g. give [23, 12, 56], return 562312.
78. [42. Trapping Rain Water -- Hard](#)
79. [258. Add Digits -- Easy](#)
80. [332. Reconstruct Itinerary -- Medium](#)
81. [check if a integer is a prime](#)
82. [460. LFU Cache -- Hard](#)
83. return node and height of the tree
84. [1780. Check if Number is a Sum of Powers of Three -- Medium](#)
85. [1429. First Unique Number](#) +
86. [23. Merge k Sorted Lists](#) +
87. [Minimum Adjacent swaps to make palindrome](#)
88. [1312. Minimum Insertion Steps to Make a String Palindrome -- Hard](#)
  1. [1143. Longest Common Subsequence](#)
  2. [516. Longest Palindromic Subsequence](#)
  3. [1062. Longest Repeating Substring](#) Binary search length
89. [Find the minimum number of deletions required to convert a string into a palindrome](#)
90. [Minimum number of characters to be replaced to make a given string Palindrome](#)
91. [Minimum number of Appends needed to make a string palindrome](#)
92. [416. Partition Equal Subset Sum -- Medium](#) / [698. Partition to K Equal Sum Subsets -- Medium](#)
93. [236. Lowest Common Ancestor of a Binary Tree -- Medium](#)
94. [48. Rotate Image -- Medium](#) clockwise: transpose + reverse row, anti: transpose + reverse col
95. given a string and a set of patterns, remove patterns from the string
  1. [205. Isomorphic Strings](#)
  2. [890. Find and Replace Pattern](#) two maps or convert string to id string abbc -> 0112
96. [17. Letter Combinations of a Phone Number -- Medium](#)
97. [692. Top K Frequent Words -- Medium](#)
98. [322. Coin Change -- Medium](#)
99. [88. Merge Sorted Array -- Easy](#) **O(N + M)**
100. [1. Two Sum -- Easy](#)
101. Quick Sort
102. [other 001](#)
  1. [79. Word Search](#)
  2. [212. Word Search II](#) backtracking + trie
  3. [makeChocolate](#)
  4. N node value.
103. [647. Palindromic Substrings -- medium](#)
104. merge to linked list / [160 intersection of linked lists](#) two pointers and HashTable
105. [20. valid parentheses -- Easy](#) finally return if `stack.size() == 0`

1. [1003. Check If Word Is Valid After Substitutions -- Medium](#)
  2. [22. Generate Parentheses -- Medium](#)
  3. [32. Longest Valid Parentheses -- Hard](#)
106. [Egg Dropping Puzzle](#)
107. Skyline
108. 155 min stack