1. Why do we need Recursion and what is it ?
2. What is a recursive tree ?
3. Basic algorithm for recursion, how should we break the input and what is a base condition ?
4. Fibonacci series and difference between recursive and iterative method
5. Print numbers from 1 to n using recursion
6. Print numbers from n to 1 using recursion
7. Reverse an array, string using recursion
8. Reverse a stack using recursion
9. Sort an array using recursion
10. Tower of Hanoi problem
11. Generating all subsets/power sets
12. Generating all unique subsets/power sets
13. Generating all permutation with spaces
14. Generating all permutation with case change
15. Josephus Problem
16. Recursive Digit Sum
17. Number of paths
18. Special Keyboard
19. Crossword Puzzle
20. Merge two sorted lists
21. Power of two
22. Power of three
23. Power of four
24. Crypto arithmetic
25. Word Break - I
26. Memorization of recursive calls.
27. What is backtracking?

28. Why Linked List?
29. Advantages/Disadvantages
30. Properties
31. What is a node, its structure?
32. Making a linked list
33. Types of Linked List
34. Length of Linked List
35. Search in a linked list

36. Insertion Singly Linked List (start, middle, end)
37. Deletion Singly Linked List (start, middle, end)
38. Insertion Doubly Linked List (start, middle, end)
39. Deletion Doubly Linked List (start, middle, end)
40. Insertion Circular Linked List
41. Deletion Circular Linked List
42. Merge two sorted Linked List
43. Reverse a Linked List iterative
44. Reverse a Linked List recursive
45. Loop in linked list (tortoise algo, hashing)
46. Start of the loop
47. Middle of Linked List
48. Palindrome or not
49. Removing Duplicates from Linked List
50. Add two numbers represented as Linked List
51. Intersection of two Linked List
52. Nth node from the end.
53. Stack using Linked List
54. Queue using Linked List
55. Delete a given Node when a node is given
56. Flattening of a Linked List
57. Reverse a Linked List in groups of size k
58. Find length of loop present in linked list
59. Insert element in a sorted linked list
60. Remove duplicates from sorted linked list
61. Merge two linked list at alternate position
62. Delete nth node from start (similarly nth node from the end)
63. Delete nth node from end
64. Insertion deletion at (nth position)
65. Print given linked list in reverse order without reversing linked list
66. What is a tree ?
67. Types of trees ?
68. What is a node and its structure ?
69. What is a Binary Search Tree
70. What is a Binary Tree
71. Construction of a Tree ?
72. Insertion of a node

73. Deletion of a node
74. Inorder traversal iterative
75. Preorder traversal iterative
76. Postorder traversal iterative
77. Inorder traversal recursive
78. Preorder traversal recursive
79. Postorder traversal recursive
80. Level order traversal
81. Vertical order traversal
82. Search in a tree
83. Height of tree
84. Number of nodes
85. Number of leaf nodes
86. Sum of all nodes
87. Different views of tree (top, bottom, right, left)
88. Inorder successor
89. Inorder Predecessor
90. Diagonal elements of tree
91. Root to leaf paths
92. Zigzag traversal
93. Diameter of tree
94. Lowest common ancestor
95. connect nodes which are at same level
96. Print mirror image of given binary tree
97. Efficient way to print nodes present between two levels
98. Print all ancestors of given node in binary tree
99. Convert binary tree to binary search tree
100. check is given binary tree is a valid binary search tree
101. Insert node in a given binary search tree
102. check if given binary tree is subtree of another one or not
103. print all nodes that do not have siblings
104. Bubble
105. Insertion
106. Selection
107. Merge
108. Quick
109. Counting

110. Shell
111. Sorting using Linked List
112. Linear search
113. Binary search
114. Ternary Search
115. First and last occurrence of an element
116. Floor/ceil of an element in array
117. Next permutation
118. Searching in infinite sorted array
119. Why Heap ?
120. What is a heap ?
121. Types of Heap
122. Implementing Heap and its basic operations
123. Kth smallest/largest element
124. K closest numbers
125. K frequent numbers
126. Median in a stream
127. K closest points to origin
128. Connecting ropes
129. What is a stack ?
130. What is a queue ?
131. Implementing stack and its operations
132. Implementing queue and its operations
133. Types of queue
134. Parenthesis checker
135. Next greater to right
136. Next greater to left
137. Next smaller to right
138. Add min() and max () methods to return minimum and maximum value from stack and queue
139. Design a stack which keeps largest element at the top
140. Reverse given stack and queue
141. Next greater to left
142. Stock span problem
143. Generate binary numbers between 1 to N
144. Stack using queue
145. Queue using stack

146. Min Stack O(n)
147. Min stack O(1)
148. Postfix, infix,prefix conversion (all combinations)
149. Postfix, infix,prefix evaluation
150. LRU cache
151. Maximum Area Histogram
152. Rain water trapping
153. What is a greedy approach ?
154. N meeting in one room
155. Activity Selection
156. Greedy algorithm to find minimum number of coins
157. Fractional Knapsack Problem
158. Minimum number of platforms required for a railway
159. Job sequencing Problem
160. Find the minimum and maximum element in an array
161. Sort the array of 0s, 1s, and 2s.
162. Reverse the given input array.
163. Find non repeated element in aray of integers
164. Check whether given array is sorted or not.
165. Find the number of 1’s in a sorted binary array.
166. Move all the negative elements to one side of the array.
167. Find the nth largest and nth smallest number in an array.
168. Find most frequent element in array.
169. Merge two sorted array to form single array.
170. Find missing number in integer array of 1 to 100
171. Find duplicate number in array of integer.
172. Find all pairs on integer array whose sum is equal to given number.
173. Remove duplicates from array of integer.
174. Move all 0s to end of the array.
175. Find duplicate elements in array.
176. Find the first repeating element in an array of integers.
177. Find group of three elements that sum to a given value.
178. 9. Find longest subsequence in given unsorted array.
179. Arrange numbers of array to form greatest number.
180. Print all combinations of numbers from 1 to n having sum n
181. Find longest increasing subsequence in array.
182. Find the minimum and maximum element in a rotated and sorted array

183. Find if one array is subset of other array.
184. "Given a sorted integer array which is rotated any number of times, find the pivot index i.e. index
185. of the minimum element of the array.
186. "
187. Search an element in a sorted and rotated array.
188. Search an element in a sorted and rotated array.
189. Print the matrix in a Spiral manner.
190. Reverse given string.
191. Find maximum occurring character in given string.
192. Write a program to remove a given character from String.
193. Remove all duplicates from given string.
194. Print duplicate characters of given string.
195. Check if two string are rotation of each other or not.
196. Print all permutations of given string
197. Find first non-repeating character of given string.
198. check if two given string is anagram of each other.
199. Check if given string is palindrome or not.
200. Find the length of the longest substring without repeating characters.
201. Find the maximum occurring character in given String.
202. Check if given string has all unique characters
203. 4. Given two strings , check if one is permutation of the other
204. There are three types of edits that can be performed on string insert,remove,replace.Given two strings write function to check how many edits are required.
205. Write a function to perform basic string compression e.g. aabccccc would become a2b1c5a3 and vice versa
206. what is graph data structure and its examples
207. BFS Traversal and its Application
208. DFS Traversal and its application
209. Find Minimum steps to reach target