**Stack:**

|  |  |  |
| --- | --- | --- |
|  | Problem | Status |
|  | Implement Queue using Stacks |  |
|  | LRU Cache Implementation |  |
|  | Implement Stack using Queues |  |
|  | Queue | Set 2 (Linked List Implementation) |  |
|  | How to efficiently implement k Queues in a single array? |  |
|  | Implement a stack using single queue |  |
|  | Implementation of Deque using circular array |  |
|  | Circular Queue | Set 2 (Circular Linked List Implementation) |  |
|  | Breadth First Traversal or BFS for a Graph |  |
|  | Level Order Tree Traversal |  |
|  | Construct Complete Binary Tree from its Linked List Representation |  |
|  | Program for Page Replacement Algorithms | Set 2 (FIFO) |  |
|  | Check whether a given Binary Tree is Complete or not | Set 1 (Iterative Solution) |  |
|  | Reversing a Queue |  |
|  | Reversing the first K elements of a Queue |  |
|  | Interleave the first half of the queue with second half |  |
|  | Level order traversal in spiral form |  |
|  | Sliding Window Maximum (Maximum of all subarrays of size k) |  |
|  | Find the largest multiple of 3 | Set 1 (Using Queue) |  |
|  | Find the first circular tour that visits all petrol pumps |  |
|  | Iterative Method to find Height of Binary Tree |  |
|  | An Interesting Method to Generate Binary Numbers from 1 to n |  |
|  | Minimum time required to rot all oranges |  |
|  | Find maximum level sum in Binary Tree |  |
|  | Sum of minimum and maximum elements of all subarrays of size k. |  |
|  | Distance of nearest cell having 1 in a binary matrix |  |
|  | Level order traversal line by line | Set 2 (Using Two Queues) |  |
|  | First negative integer in every window of size k |  |
|  | Minimum sum of squares of character counts in a given string after removing k characters |  |
|  | Queue based approach for first non-repeating character in a stream |  |
|  | Averages of Levels in Binary Tree |  |
|  | Stack Permutations (Check if an array is stack permutation of other) |  |
|  | Check if all levels of two trees are anagrams or not |  |
|  | Check mirror in n-ary tree |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |