LeetCode POTD July 2024

1. Three Consecutive Odds
Same approach as Kadane's Algorithm.

2. Intersection of Two Arrays II
Use Sorting and Two Pointers to get the solution.
Use HashMap and count the number of times each element appears.

3. Minimum Difference Between Largest and Smallest Value in Three Moves (Google OA Question)
Either we can make the largest value smallest or the smallest value largest to minimize the difference (Greedy).
Change the first three or last three, or a combination of (1 First, 2 Last) or (2 First, 1 Last) and take the minimum.

4. Merge Nodes in Between Zeros
Easy problem of Linked List. Just did what the question asked.

5. Find the Minimum and Maximum Number of Nodes Between Critical Points
Store all the local max/min (critical points).
Then max diff will be first and last, and find the minimum difference.

6. Pass the Pillow
Going from left to right, increment the index; from right to left, decrement the index.
When the time is 0, return the index.
Number of passes between n people is (n-1).
Find the number of rounds = time / (n-1), then timeLeft = time % (n-1), and based on the number of rounds, find the index.

7. Water Bottles
Difficult to understand, but it is a simulation.
Video Explanation: https://youtu.be/v2D_v3llC98?si=umaACfSTJgsz5pIv

8. Find the Winner of the Circular Game
Using array simulation {nextIndex = (i + k - 1) % n}
Using queue simulation
Josephus Problem - Important problem of Recursion

9. Average Waiting Time
Dry Run Question

10. Crawler Log Folder
Perform operations on the integer according to the conditions and find the answer.

11. Reverse Substrings Between Each Pair of Parentheses (Amazon & Meta)
Store the index of the opening parentheses in a stack and use StringBuilder.

12. Maximum Score From Removing Substrings
Using Stack
First, remove the maximum cost substring.
Second, remove the second cost substring.

13. Robot Collisions
Similar to Asteroid Collision
Sort the index of the robots in the order in which they appear on the line.

Using Stack, simulate the process.

14. Number Of Atoms
Return the number of atoms of each individual element.
Use a stack and insert elements along with their counts as a map.
Upon encountering a closing bracket, multiply the next number with the frequency of elements on the stack.
Add the frequency for the same elements.
Return the elements along with their count in lexicographical order.

15. Create Binary Tree From Descriptions
The root will be the node that is not the child of anyone.
Store the value and the node address in a map, start creating the tree step-by-step, and return the node (root) that is not the child of anyone.

16. Step-By-Step Directions From a Binary Tree Node to Another
First, find the Lowest Common Ancestor (LCA).
Then find the path from the LCA node to the source node using DFS.
Find the path from the LCA to the destination node and replace the LCA path string with 'U'.

17. Delete Nodes And Return Forest
Perform bottom-up traversal.
If nodes to delete are found, add the left and right nodes to the list and delete the current node by returning null.

18. Number of Good Leaf Nodes Pairs
Create an undirected graph or store the parent of each node, then perform BFS to find pairs with a minimum distance less than or equal to the required distance.

19. Lucky Numbers in a Matrix (Cisco)
Only one lucky number in the matrix.
Populate an array with the minimum element in each row and the maximum element in each column.
Traverse the matrix and if the same minimum and maximum are found for the given row and column, return that number as the lucky number.

20. Find Valid Matrix Given Row and Column Sums (Uber, Microsoft)
Use a greedy approach.
c1 + c2 = r1 + r2
Video Explanation: https://www.youtube.com/watch?v=9dKdLNlmxco

21. Build a Matrix With Conditions
Video Explanation: https://www.youtube.com/watch?v=QyrmIu4Vo8A

22. Sort the People
Use TreeMap or Comparable on the pair.

23. Sort Array by Increasing Frequency
Store the frequency of the numbers in a map.
Store the frequency and the number in a list of Pair objects.
Apply a comparator to sort the pair objects and return the answer.

24. Sort the Jumbled Numbers
Convert the number to their corresponding numbers and store their indices as objects.
Then sort the objects.

25. Sort an Array
Use Merge Sort.

26. Find the City With the Smallest Number of Neighbors at a Threshold Distance
Fill the matrix of minimum distances from every city to every other city using Floyd-Warshall.
Iterate through the matrix to find the required city.

27. Minimum Cost to Convert String I
Surprisingly, a graph problem.
Create an adjacency matrix of 26x26 and store the cost from every character to the next character.
Apply Floyd-Warshall on it to find the minimum cost, then solve the problem.

28. Second Minimum Time to Reach Destination
Best question for Dijkstra's Algorithm.
Store the minimum time in the first array and the second minimum time in the second array.
Also, take care of the signals using simple math.
Video Explanation: https://www.youtube.com/watch?v=_rnQKrA9xzA&t=2585s

29. Count Number of Teams
Loop from 1 to n-1, take the i-th character as the middle, and count the number of min or max numbers on the left and right.
Then solve.

30. Minimum Deletions to Make String Balanced
Find the number of 'b' characters to the left of each element.
Then find the number of 'a' characters to the right of each element.
Traverse the array and find the minimum deletions.

31. Filling Bookcase Shelves
A question of Dynamic Programming.
Try all possible ways to keep the book on the same shelf or the next shelf.
If on the same shelf, the height will be the tallest book in the shelf.
If on the next shelf, add the book's height to the current shelf height.