VISA Online Assessment practice sheet

General rounds:-

1. Online coding assessment on hackerrank (4 coding questions)
2. Technical Interview (2 rounds)
3. HR Interview (1 round)

Coding questions

CODING ROUND:

1. Maximum Score: An interviewer at HackerRank recently came up with an interesting

problem for the candidates. Given an array arr of n integers, start with a score of 0. In one operation, one element of the array is chosen, and its value is added to the score. The element is replaced by the integer ceiling of one-third of its value. For example, if the element is 10, then 10 is added to the score and the element is replaced by ceil(10/3)=4. The task is to find the maximum possible score after k operations.

2. Alex is shopping at Ozone Galleria Mall. There is a dedicated cubicle for a type of product at the shopping center. All the products sold at the ith cubicle are priced the same, denoted by prices[i]. The cubicles are arranged such that the price of the products sold at each cubicle are in non-decreasing order. Several queries would be asked about the problem. In each query, the cubicle number Alex is initially standing at, and the amount of money Alex has is given, and Alex can travel in the right direction visiting from the current cubicle to the last cubicle. Alex may buy at most one item from any cubicle visited, but the total cost of the purchase must not exceed the amount Alex has. Report the maximum number of products that can be purchased for each query. More formally, given an array of n integers, prices, where prices[i] denotes the price of the product sold in the ith cubicle. The array prices are in non-decreasing order (i.e., the price[i]≤price[i+1]), and q queries need to be processed. For each query, two integers are given:

pos: Alex's initial position
amount: the amount of money Alex has

Alex needs to visit each cubicle from number pos to n, purchasing at most one product from each cubicle. For each query, the goal is to find the maximum number of products that Alex can buy.

3. https://www.geeksforgeeks.org/inversion-count-in-array-using-merge-sort/

4. https://www.hackerrank.com/challenges/filling-jars/problem

5. https://leetcode.com/problems/longest-increasing-subsequence/

6. https://leetcode.com/problems/sell-diminishing-valued-colored-balls/

7. https://www.geeksforgeeks.org/compute-before-matrix/

8. https://leetcode.com/problems/minimum-falling-path-sum/

9. https://www.geeksforgeeks.org/largest-sum-contiguous-subarray/

CONFIDENTIAL

10. https://www.geeksforgeeks.org/count-triplets-with-sum-smaller-than-a-given-value/

11. Given a string consisting of letters that indicate directions i.e. 'L' for left, 'R' for right, 'U' for, up and 'D' for down, find the minimum length of string that can be obtained after deleting some characters of the string so that we reach the same destination as the original string.

12. Given 'n' Jars filled with 'm' number of Jellybeans. 'T' represents the number of operations performed on these jars. Given a range [a-b] and number of jelly beans to be filled in the jars lying in the range [a-b], find the number of jelly beans in each jar after these 'T' operations

13. Given a number 'N' and an array a[ ], find the number of possibilities of a[i]-a[j]=N such that i>j. (Could be solved in O(n) using a HashMap)

14. Given the number of nodes and the number of edges connecting these nodes, arrange these edges such that maximum number of nodes are strongly connected. Return the number of nodes that could be strongly connected. [Hint:- To make first node strongly connected you need n-1 edges for second you need n-2 and so on. i.e edges – (n-1) – (n-2) - ... - (n-k) >= 0, then largest k possible is the answer).

15. https://www.hackerrank.com/contests/w1/challenges/volleyball-match/ (use combinatorics using dp to solve it).

16. https://www.geeksforgeeks.org/find-the-number-of-islands-using-dfs/

17. https://www.geeksforgeeks.org/maximum-profit-by-buying-and-selling-a-share-at-most-twice/

18. https://www.naukri.com/code360/interview-bundle/visa

19. https://algodaily.com/companies/visa

20. https://workat.tech/company/visa/interview-questions/problem-solving?status=all

21. Find if two linked list intersect.

22. Akash had recently bought a puzzle book. Each page in the book has a word puzzle in which a grid of letters from the English Alphabet (uppercase or lowercase) or digits between 0-9 are given. Akash has to figure out the number of occurrences of a particular word in the given grid. The grid is always a square, and the word can be present in any direction in the grid i.e. left to right, right to left, top to bottom, bottom to top, and the diagonals. Palindromic words (words which are read the same as the original word in the opposite direction too) if present in the grid will be counted twice. Write a program to help Akash. Read the input from STDIN and print the output to STDOUT. Do not write arbitrary strings anywhere in the program, as these contribute to the standard output and test cases will fail.

Input Format:
The first line of input has N, which is the number of rows/columns in the grid.

CONFIDENTIAL

The next N lines each contain N-characters (alphabets or digits).
The last line contains the word whose number of occurrences has to be found out.

Output Format:
The output has the number of occurrences of the given word in the grid.

Constraint:
N >= 3.

Sample Input 1:

3
ctt
cat
cct
cat
Sample Output 1:
4

23. https://www.geeksforgeeks.org/maximum-length-subarray-with-difference-between-adjacent-elements-as-either-0-or-1/

24. https://www.naukri.com/code360/problems/ninja-and-the-dance-competetion_1172167

25. Imagine that you are standing at the starting point of a straight street and trying to reach the end of the street. This street is represented by a number line starting at 0 and ending at finish (finish > 0). There are electric scooters scattered along the street to help you get to the end. Specifically, the scooters are represented by an array scooters, with scooters[i] representing the location of the i^{th} scooter. Each scooter can travel upto 10 points along the number line before its battery is fully discharged and it cannot go further. For example, if a scooter is at point 5, it can travel to points 5,6,7,8 ... upto point 15 (inclusive), but it cannot get to the point q6 or further.

To get to the end point of the street, you must use the following algorithm:-

1. From the current position, travel to the nearest scooter to the right on foot. If there are no more scooters available, travel to the end point on foot.
2. Get on this scooter and use all of its battery/resources to travel as far as you can toward the end point.
3. If you still haven't reached the end point, repeat this process from step 1.

Given that you must use the algorithm described above to travel from starting point of the street 0 to the end point of the street finish, your task is to return the total distance that you will travel on scooters.

NOTE. You are not expected to provide the most optimal solution, but a solution with the time complexity not worse than O(scooters.length x finish) will fit within the execution time limit.

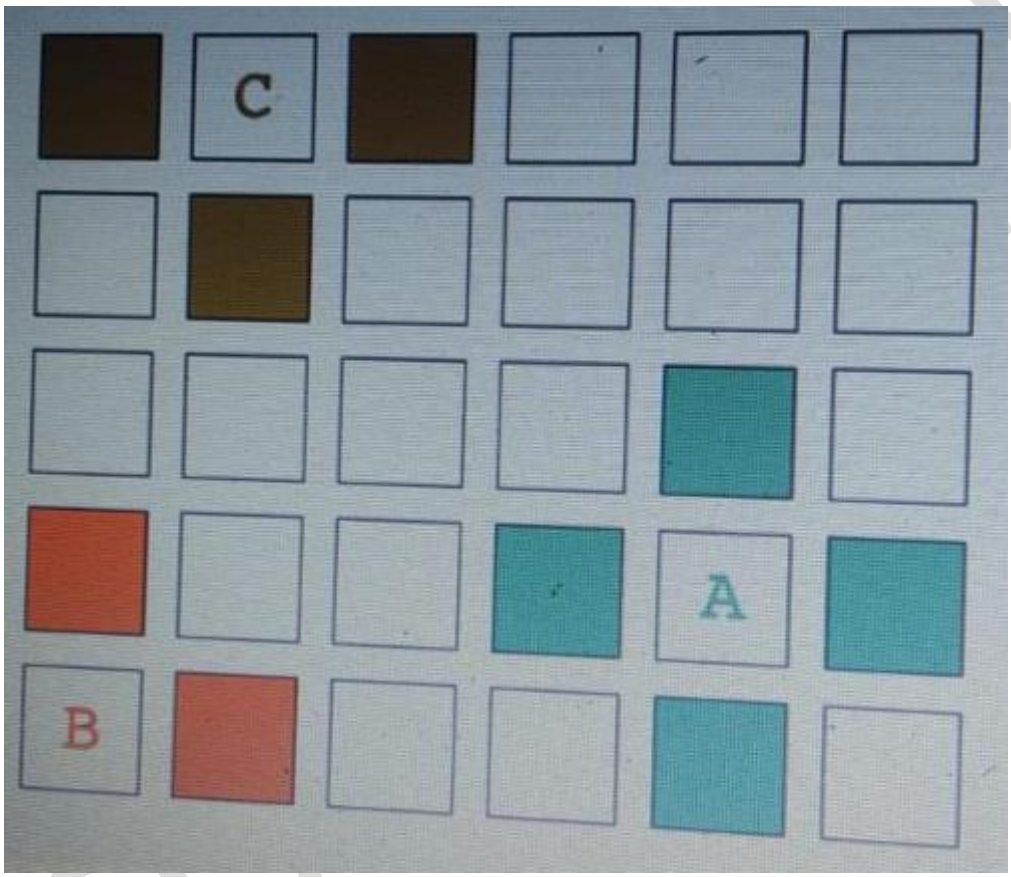
Example

1. For finish =23 and scooters = [7 ,4, 14], the output should be 19. 4->14 then 14->21 = 19
2. For finish = 27 and scooters = [15, 7,3,10], the output should be 3->13 and 15-> 25 = 20.
3. For finish =10 and scooters = [], the output should be 0.

CONFIDENTIAL

First line of input is an integer representing finish, next line has n representing the number of scooters and next line contains n integers representing the location of the scooters.

26. Imagine you are given a board of cells, each containing a bubble of a specific colour (as shown below). Neighbouring cells of the bubble are defined as adjacent cells (on either the same row or column as the given cell) which have a common side with the given cell. For example, the neighbouring cells for each of the cells A, B and C are highlighted in corresponding colour in the picture below.



Your task is to perform a bubble explosion on the board. A bubble is defined by the following rules:

- A bubble within any given cell is eligible to explode if it has the same colour as bubbles in at least 2 neighbouring cells.
- All eligible bubbles and bubbles of the same colour in neighbouring cells are marked for explosion.
- All marked bubbles explode at the same time. Exploded bubbles are removed from the board, resulting in empty cells.
- After all exploded bubbles are removed, remaining bubbles in cells above the empty cells drop down to fill all empty cells.

You are given an initial board of cells bubbles – a multidimensional array of integers representing cells containing bubbles of various colours. Return the board state after a bubble

explosion. The output should be a multidimensional array of integers with the same size as bubbles, but replacing all the empty cells (without bubbles) with 0.

NOTE: You are not expected to provide the most optimal solution, but solution with the time complexity not worse than O(bubbles.length^2. bubbles[0].length^2) will fit within the execution time limit.

For Example

Bubbles = [[3, 1, 2, 1],

[1, 1, 1, 4],

[3, 1, 2, 2],

[3, 3, 3, 4]]

The output should be

[[0, 0, 0, 0],

[0, 0, 0, 1],

[0, 0, 0, 4],

[3, 0, 2, 4]]

CONFIDENTIAL