D.E. SHAW

Date : 22 July 2025

OA Round - 1 [9 am - 10 : 30 am]

The OA round consisted of 3 sections - Coding , Technical , Aptitude.

Coding Section

Question 1 [20 Mins]

We are given an array of integers A of size N. We are allowed to do two type of operations on it.
1. Pick some prefix of the given array and increment all elements by 1.
2. Pick some prefix of the given array and decrement all elements by 1.

We want to make all elements of the array equal to zero. We need to determine the minimum number of operations needed for that.

Constraints :
1 <= N <= 2e5
-1e9 <= Ai <= 1e9

Solution :

Question 2 [26 min]

Given a set of bus stations connected with each other using bidirectional roads , some stations are marked black and some marked white. The whole connection formed a tree with some black nodes and some white nodes. We want to add one edge between two nodes such that they are at least two edges apart. The addition of one edge will eventually form the cycle. We want to ensure that all nodes in the cycle are of the same color. We have to report in how many ways we can add this one edge.

Number of nodes = N

1 <= N <= 2e5

Solution :

Section 2 (Technical Section) [24 Min]

The section consisted of CS fundamentals , data structures , error detection of code , sql queries , WEB technology questions. There were a total of 12 questions in this section from which I attempted 9 - 10 questions as there was negative marking.

Section 3 (Aptitude)[20 Min]

This section consisted of math , physics and logical thinking questions. The questions were doable if someone maintain there pace

Technical Issue

During the OA there was some technical issue from college's side (Power cut had occured) due to which many were not able to submit their tests as many were using ethernet. When the time was up , the D.E. Shaw team decided to take another test to give everyone a fair chance. So the next test was scheduled right after the first test was over.

OA Round 2 [12 : 30 pm - 2 : 00 pm]

Coding section

Question 1 (20 Min)

You are given an array of floats A where 1.01 <= Ai <= 3.00 . We can put some elements in a container where the weight of the container is the sum of elements in it.
This weight must be smaller than or equal to 3.00. All the elements have to be in some container , we have to report the minimum number of containers required for that.

Constraint :

1 <= N <= 2e5
1.01 <= Ai <= 3.00

Solution :

Question 2 (30 Min)

You are given an array A of integers of size N. We want to create an array B of size N such that following condition holds true :

a. 1 <= B[i] <= A[i].
b. B[i] != B[i + 1]

We have to report the number of such array B that can be created.

Constrains :

1 <= N <= 2e5
1 <= A[i] <= 1e9

Solution :

Section 2 and Section 3 were similar as in OA 1 but they were relatively harder and even some topics were from out of syllabus.

Post OA

Within half an hour I got the call for my interview that was scheduled within 10 minutes.

Technical Round 1

I was called in panel 1 , there were 2 panellists. First they asked me about how the OA was and asked for some feedback related to it. Then they straight went to asking the questions

Question 1

We are given a n - array tree denoting the manager - employee structure at D.E. Shaw. The parent of the node is its immediate manager. We are given a single API which can take node id (Every node has a unique id) and will return the id of its immediate manager. The root is the CEO and for that node this API will return null. We have to implement another API which will take the node id of two nodes and return their first common manager (Basically LCA).

Solution :

One optimal solution to solve this would be to find the depth of both nodes from root and then lift the node at greater depth from till it reaches at the same level as other. Say node A and node B. Where depth(A) >= depth(B)
So we will lift node A till it reaches the same depth as of B. **i.e** depth(A) == depth(B)
Now as both node are at same depth the following two case may arise :

I. Node A == Node B

In this case node B is the LCA of A and B

II.Node A != Node B

In this case we will lift both nodes till we reach a common node and that node will be the LCA

Time Complexity : O(N) , Where N is the size of tree
Space Complexity : O(1)

Question 2

We are given a grid of 0's and 1's where 1 denotes land and 0 denotes water. All the land cells can be connected to each other if they share a common side (i.e. they are adjacent). One such connected group of land is an island , we have to find the number of islands present in the grid.

Solution :

We will consider all cells of the grid as nodes of a graph and each land cell is connected to another land cell by a bidirectional edge if they share a common side. We will keep a track of all the cells that are visited or not in some data structure and we will have our ans variable initialised to zero at the start. Then we will traverse the grid and whenever we reach a land cell which is not visited as of yet then we will start traversal from that cell to all other land cells that are connected directly or indirectly and increment the ans variable by 1 when traversal ends. We will keep doing this until all land cells are visited. At the end ans will store the count of number of islands

Time complexity : O(N * M)
Space complexity : O(N * M)

Question 3 (Puzzle)

There is a room (with no light) which has 20 shoes (10 of left leg and 10 of right leg). We can enter the room only once and take some amount of shoes with us and then check if there is a valid pair of shoes to wear or not (i.e at least 1 left leg shoe and 1 right leg shoe). How many minimum number of shoe we have to carry to guarantee

There will be at least one valid pair.

Solution :
Answer = 11
By Pigeonhole principle

Now we have 40 shoes (20 of black color and 20 of right color) among those 20 shoes there are 10 of left leg and 10 of right leg. Now how many minimum shoes we have to carry of guarantee at least one valid pair (At least one of left leg and one of right leg plus of same color)

Answer : 21

Now we have a total of **m** shoes in a room of **n** different color (m is divisible by 2 * **n**). There are (m / n) numbers of shoes of the same color among which (m / 2n) is of left and right leg. Now what is the general equation of the min number of shoes to carry in order to guarantee at least one valid pair.

Answer : (m / 2) + 1

Question 4 (Puzzle)

Just a childhood puzzle where there are 2 islands and you along with 3 Policeman and 3 Criminals and one boat are on one island. You need to take all of them to the other island. The boat can carry only at most 2 people. On any island, the number of Policemen >= number of Criminals . Min number of usage of the boat should be found. I have heard the puzzle but was not aware of the solution beforehand , and the panelist gave enough time to solve this (Around 15 mins) Later they found that I had misunderstood the question and then they clarified that , and finally I was able to solve it using the BFS kind of approach.

Then they asked me for an overview of the project in my resume and some OS questions . I was not able to answer that OS question as I was not aware how to do that (It was related to file handling)

After this they said the interview was over and if I have any questions for them I can now ask them . I asked some company related questions and some related to the SDET role and culture of the office.

HR Round (22 July , 7 PM)

I was called for an HR round on the same day in the evening and went to CCC for the same at around 7PM. Then the interview started and it began with my introduction. Then she asked me why I wanted to join D.E. Shaw and then she asked if you are aware of the SDET role or not. She was also focused on why I want to join the SDET role rather than the SDE role. Then she highlighted that they want people that are willing to be part of the firm for the long term and all such info that are there in the HR round. This round was chill and she was just checking my behaviour and reason to join the firm. (It was just a normal conversation round).

Technical Round 2 (23 July , 10 AM)

This round started with a formal introduction then they asked me some questions.

Question 1

Given an array of stack of positive integers we have to pick at most k integers . We have to report the maximum sum we can generate. For picking integers we have to pick a stack and take the topmost element of that stack.
If total elements in whole array is **N ,** then **N * K <= 1e7**

Solution :

In this question the array was given to us in the form of vector<vector<int>> .
In this question any greedy will not work in all cases so we need to check all possible ways to get max sum. This can be done using DP where the state will be defined as follows :
dp[i][j][k] -> Max sum we can generate from subarray from index i to n - 1 , such that we have picked j elements from current stack and we can pick k elements more

Transitions :
Dp[i][j][k] = max(arr[i][j] + dp[i][j + 1][k - 1] , dp[i + 1][0][k])

Time Complexity : O(N * K)
Space Complexity : (N * K)

Question 2

Given a chocolate bar of length N , we know the sweetness of every single unit of it which is given to us in the form of an array of integers of size N. I have k friends. I have to divide that bar into k + 1 segments , where the tastiness of a segment will be the sum of its elements. Among those segments I will take the one with minimum

tastiness. We have to determine what is the maximum tastiness I can get provided everyone gets exactly one segment.
Constraints :
Size of Array -> **N <= 1e5**
A[i] <= 1e9
Solution :

In this problem we just need to partition the array into k + 1 number of segments such that the sum of segment with minimum sum gets maximised.
Let X denote the sum of the segment with minimum sum. Then we just need to find the max value of X

Here lets say we found for some value of X (say 6) that it is possible to partition the array into k + 1 segments such that no segment has sum less than 6. Then we can for sure say that we will be able to do this partitioning for all values of X that are less than 6.
Similarly of there is no partitioning for some value of X (Say 100) then no value greater than 100 can satisfy the condition

Thus as the search space of X is monotonic thus we can apply **Binary Search** over it.

In the predicate function (Say we are checking for some value Y) we will start with the left end of the array and start iteration by maintaining the sum of the current segment and once this becomes greater than or equal to Y , we will partition the array here and make sum of current segment equal to zero. At last if the count of segments is greater than or equal to K + 1 return True else False.

Time Complexity : O(N * log(Sum of elements of array))
Space Complexity : O(1)

Question 2.1

Now in the same question we know with every friend how much friendship I have. For every friend **i** and **j** , if friendship[**i**] >= friendship[**j**] , then the following condition has to hold -

Tastiness of segment given to friend **i** >= Tastiness of segment given to friend **j.**

Now we have to report which segment will be allotted to which friend provided I will get the segment with minimum tastiness which has to be maximised.

Solution :

In this modification of question 2 the same Binary search will work. At last we will just have to find what is the partition of the array and sort the segments according to their sum . Then allocate the segment with min sum to ourselves then allocate the next segment to friend with min friendliness and so on.

Projects :

After these questions they asked me about the project in detail ,like why I did that , how I did that , which techstack was used. Also they asked me how I would scale this project for actual users. This part will be specific to the ones project and they will definitely deep dive into projects , so one has to be prepared for in depth questions.

Additional Round

They called three of us and said the seat is limited so they will be conducting two more additional rounds for us and then they will declare the final result. They said this round is going to be the hardest of all and said who is going to be the first one.
(I raised my hand).

Then they said you guys have heard from yours seniors about the last round
Which traditionally is the **Prank** round. Then they said all three are selected.