

**Interview Round 1(60-70 Minutes):**

Technical Interview

**Question 1:** Check if a character link list is palindrome or not.

**Question 2:** A sorted array has been rotated  $r$  times to the left. Find  $r$  in least possible time.

**Question 3:** Clone a singly link list whose nodes contain, apart from next pointers, an extra pointer to any random node. The random pointer of a node  $N$  could be after  $N$ , before  $N$  or the node  $N$  itself.

**Interview Round 2(50-60 Minutes):**

Technical Interview

**Question 1:** There is a big file of words which is dynamically changing. We are continuously adding some words into it. How would you keep track of top 10 trending words at each moment?

**Question 2:** Write code for minHeapify() operation.

**Question 3:** Design a data structure for the following operations:

- I. Enqueue
- II. Dequeue
- III. Delete a given number(if it is present in the queue, else do nothing)
- IV. isNumberPresent

All these operations should take  $O(1)$  time.

**Question 4:** Write a function that returns the length of the longest leaf-to-leaf path in a binary tree.

**Interview Round 3(60-70 Minutes):**

## Technical Interview

**Question 1:** There is a binary tree of size N. All nodes are numbered between 1-N(inclusive). There is a N\*N integer matrix Arr[N][N], all elements are initialized to zero. So for all the nodes A and B, put Arr[A][B] = 1 if A is an ancestor of B (**NOT** just the immediate ancestor).

**Question 2:** Find an element in a sorted rotated integer array.

**Question 3:** There is a N\*N integer matrix Arr[N][N]. From the row r and column c, we can go to any of the following three indices:

- I. Arr[ r+1 ][ c-1 ] (valid only if c-1>=0)
- II. Arr[ r+1 ][ c ]
- III. Arr[ r+1 ][ c+1 ] (valid only if c+1<=N-1)

So if we start at any column index on row 0, what is the largest sum of any of the paths till row N-1.

**Interview Round 4(40-50 Minutes):**

## Bar Raiser Round

Interviewer asked HR Questions Initially, then a sort of puzzle.

Two robots land with their parachutes on an infinite one-dimensional number line. They both release their parachutes as soon as they land and start moving. They are allowed only to make use of the following functions.

- I. moveLeft() // robot moves to left by 1 unit in 1 unit time
- II. moveRight() // robot moves to right by 1 unit in 1 unit time
- III. noOperation() // robot does not move and takes 1 unit time
- IV. onTopOfParachute() // returns true if the robot is standing on top of either of the parachute, else false