<table>
<tr><td>Mayank Kumar</td><td>1. Invert a binary tree.
a. Using DFS uses more memory
b. Optimise memory (Hint: level order traversal)
2. Sort an array of 0s and 1s in a single pass.
3. What are threads in Java ?
4. What are the differences between Python and C++ ?
5. What is a deadlock ?</td></tr>
<tr><td>Varsha Tumburu</td><td>First 15 mins: Project discussion
DSA: 3 questions (35 mins)
1. Print right view of binary tree
2. Asked if I knew about linux tail command. (Retrieves the last k lines from file). Asked to give an approach and implement it.
3. Given a nested list of integers, return the sum of all integers in the list weighted by their depth. Each element is either an integer, or a list
-- whose elements may also be integers or other lists.

Input: the list [[4,[4,5],4],2,[1,1]],
Output: 4*2+4*3+5*3+4*2+2*1+1*2+1*2=49.

/**
* // This is the interface that allows for creating nested lists.
* // You should not implement it, or speculate about its implementation
* class NestedInteger {
* public:
* // Return true if this NestedInteger holds a single integer,
* // rather than a nested list.
* bool isInteger() const;
*
* // Return the single integer that this NestedInteger holds,
* // if it holds a single integer
* // The result is undefined if this NestedInteger holds a nested list
* int getInteger() const;
*
* // Return the nested list that this NestedInteger holds,
* // if it holds a nested list
* // The result is undefined if this NestedInteger holds a single integer
* const vector<NestedInteger> &getList() const;
* };
*/
int depthSum(const vector<NestedInteger>& nestedList) {
//complete this function given the above class exists
}

OOPS:
Asked about the 4 main concepts and to explain them. Then we talked about virtual functions and how they work. Asked about the diamond problem and how to resolve it.
Databases:</td></tr>
</table>

	Asked about the various table joins. Difference between cartesian join and inner join. Which one is more costly? Asked about NoSQL,etc but told him we didn't get to the course.
Shreya Dimri	Project discussion : 10 minutes DSA : 40 minutes , 2 questions Question 1 : Implement pow(x,n) function. I had to come up with the most optimal solution Question 2 : Beautiful arrangement - Suppose you have n integers labeled 1 through n. A permutation of those n integers perm (1-indexed) is considered a beautiful arrangement if for every i (1 <= i <= n), either of the following is true: perm[i] is divisible by i. i is divisible by perm[i]. Given an integer n, return all the beautiful arrangements that you can construct. The interviewer wanted me to write optimal codes and go through their dry run. https://leetcode.com/problems/beautiful-arrangement/ Next 20 minutes were spent on OOPs and basic C++ concepts Questions : State Principles of OOPs and the four features. What is method overloading and overriding When functions have different return types but same name is it still overloading A few questions on data structures and their working. Advantages of using Binary Trees. Gave me different scenarios and asked which data structure I would use and why.
Tanushree	The interviewer first asked me about the applications of trees and graphs. After this he asked what happens in topological sort and its applications and also about disjoint set union and its applications. The first question was a simple implementation problem. A string of the form a5b2c3 was given. The output in the form of aaaaabbccc was to be displayed. The second question was based on a linked list. You have been given a linked list, let's say 1 2 5 7 3 5. You have to rearrange the linked list and return 1 5 2 3 5 7. This had to be done in place and in O(n) time complexity. In my implementation I needed to find out the midpoint of the linked list and he expected me to find that without finding the actual length of list=> I used Floyd's fast and slow pointer. Then space and time complexity was discussed. After this he asked me about various types of sort operations I knew.

	Then he asked me about the time complexity of different quick sort Algorithms and also about the worst case condition of a quick sort Algorithm. Then he asked me to explain some concepts of oops-> friend function, polymorphism and examples of polymorphism. Then he gave me a small snippet in which inheritance was used to implement polymorphism and asked me what the output will be. At the end he asked me about what differences I noticed while coding in JAVA and C++. all the codes had to be written without using any function from STL.
Ishita Singh	I was given an array which was sorted in ascending order, but it was rotated about a point. Eg- for 1,3,5,7,8 we might be given 5,7,8,1,3. I was also given a number x, for which I had to return the index of x, if it was present in the array, or else -1. After solving it I was asked questions on the time and space complexity. It followed general questions on hashing, binary search trees, comparison of different operations on various data structures like linked list, stack, queue, map, and wanted me to explain how hashtables are implemented.
Kavya Goyal	Project Discussion : 10 mins Trees traversals Was asked about basics of LRU cache. What does it do, and how to implement. Asked to implement everything in O(1) operations.
Gaurav Rai	I was asked these questions: 1. Write the code of Random string generator , I wrote the code using rand() function then he asked code for rand() function. 2. If a stream of strings is there ,he asked to find if the given string is repeated or not and if it's repeated then return it's frequency. I told the solution using unordered_map then he asked about it's time complexity and asked to implement unordered_map and write code for it. 3. Ifa array consisting of only 0,1,2 values is given , sort it without using the sort() function and extra space. I gave the solution using two variables a and b to store the frequency of 0's and 1's while passing the array first time and in 2nd pass will update the array .Then he asked me to sort it in 1 pass only . (solution: for doing in 1 pass/iteration only we will take 3 pointers low ,mid and high , explanation link : https://www.youtube.com/watch?v=oaVa-9wmpns&list=PLgUwDviBIf0rPG3Ictpu74YWBQ1CaBkm2&index=2).
Shivam	Real life applications of Stacks and queues. Asked in the stacks are implemented in the browsers and if yes..how?? OOPs concepts When functions have different return types but same name is it still overloading and some followup State Principles of OOPs and the four features.

	Types of polymorphism and meaning and some followup
	
	DSA :-
	
	Input: the list [[4,[4,5],4],2,[1,1]],
	Output: 4*2+4*3+5*3+4*2+2*1+1*2+1*2=49.
	
	/**
	* // This is the interface that allows for creating nested lists.
	* // You should not implement it, or speculate about its implementation
	* class NestedInteger {
	* public:
	* // Return true if this NestedInteger holds a single integer,
	* // rather than a nested list.
	* bool isInteger() const;
	*
	* // Return the single integer that this NestedInteger holds,
	* // if it holds a single integer
	* // The result is undefined if this NestedInteger holds a nested list
	* int getInteger() const;
	*
	* // Return the nested list that this NestedInteger holds,
	* // if it holds a nested list
	* // The result is undefined if this NestedInteger holds a single integer
	* const vector<NestedInteger> &getList() const;
	* };
	*/
	int depthSum(const vector<NestedInteger>& nestedList) {
	//complete this function given the above class exists
	}
	
	Reverse a linked list using recursion in optimal time
	
	Print right view of a binary tree
	
	NO PROJECT DISCUSSION IN MY CASE

Shreeyans Jain	1. Tell me about yourself and a brief explanation about the projects. 2. There were n people in a row and they all have a specific immunity they can go out when the sum of their immunity is greater than a threshold value tell total no of different groups which can go out and all the people in the group should be adjacent to each other.(Basically total no of subarray which has sum less than k) 3. Difference between binary tree and BST, how to know if a binary tree is a BST or not

	4. What is recursion, tell if two trees are mirrors of each other or not :) 5. Time complexity of various sorting algorithms(best, average, and worst-case) 6. Oops questions like: a. What is polymorphism and explain type of polymorphism b. Difference between method overloading and overriding c. What is abstraction d. What is Hierarchical inheritance e. What is abstract class and pure virtual function
S K Touheed	1. Find square root of n without using sqrt() function 2. What is string immutability 3. A map whose keyword is string and stores list of strings is given and a paragraph is given if a word matches with any of the words in map of list of strings replace it with the keyword. 4. If a nxn matrix whose rows and columns are sorted. Can we make a sorted 1D array of size n^2 in time complexity O(n^2).
Mayank	https://www.google.com/amp/s/www.geeksforgeeks.org/assembly-line-scheduling-dp-34/amp/ https://leetcode.com/problems/word-ladder/description/