

# Google Interview Experience For Software Engineering Internship 2022

- Difficulty Level : [Medium](#)
- Last Updated : 24 Aug, 2021

## A Short Overview:

Google came to our campus(IIT Roorkee) to select candidates for Summer Intern 2022 in August 2021. Only 8 students were selected finally after all the rounds. The entire process was virtual and was conducted in 1+2 ( 1 coding round + 2 interview rounds) rounds of 45 minutes each.

## Interview/Technical Rounds(Total 3):

### Round 1:

This round went on for nearly 55 minutes for me. The interviewer was very friendly and helpful and he first asked me to introduce myself. Then he asked me a small question which was followed by 5-6 follow-ups:

- Given an unsorted array containing duplicates and a number, find at what position that number would occur if the array is sorted. (I first told him the  $O(N \log N)$  solution of sorting and then finding the element and then I proposed to him the solution of  $O(n)$  of traversing the array and counting the number of elements that are smaller than the given number. He was satisfied and asked me to code the solution)
- Then he asked me what if the index of the element is given instead of the element.
- I don't remember other follow-ups, but they were quite similar and were based on binary search and maps.
- I was able to answer all of them and the interviewer was quite satisfied with me and told me that it ended very well.

Then I got a mail regarding my next round which was scheduled 1 hr later.

### Round 2:

This round went on for nearly 45 minutes for me. The question asked was:

Given a chat history represented in the form of a vector of strings, find the top k users who spoke the maximum number of words.

Sample Input:

Rohan: hey there how you doing

Naman: I am fine you tell

Rohan: me too

- I proposed to him the solution that first of all I will traverse the vector of strings and map the number of words said by each user to their user name(i.e. `map<string, int>`) then I can either make a vector from that map and sort it and take last k elements out or will use `priority_queue`

of max heap instead and then pop only k elements.

- He asked me to code the first part of inserting everything in the map. My code was appending a char to string. So he asked me to avoid that since that was use  $O(\text{length of the string})$  time. So I changed the solution to using indexes and not appending.
- Coding the first part with optimization and dry running took the whole of my time and wasn't able to complete the second part.

After both rounds were over, I was waiting for the results since only 2 rounds were supposed to happen. But the next day in the morning I got a call from the placement and internship cell of my college that google has demanded another interview of mine.

### Round 3:

This round extended to 75 minutes for me. The question asked was of graphs:

Given is a network of cities. A crime has occurred in city C and the criminal has fled to his hometown H. Now the police want to catch the criminal and he knows that the criminal will go from city A to city B only if the shortest path distance from city B to H is less than city A to H. We are supposed to find the number of possible routes that the criminal might have taken starting from C.

- I told him I will use the adjacency list representation to represent the undirected weighted graph. Then first of all I will calculate the shortest distance from H to each of the cities using the Dijkstra algorithm and then my answer would be the sum of the number of routes from all the cities connected to city C whose shortest distance from H is less than that of C, i.e. will use recursion.
- Then he asked me the time complexity for recursion I told him  $O(V+E)$ . But he asked if I am sure and asked me to draw an example in my notebook and find if I am missing something. Then using one of my drawn examples he helped me in getting to the conclusion that the time complexity of my solution would be exponential because of repetitive subproblems which led me to the idea of memorization.
- Then he asked me to code it. I coded my Dijkstra, recursive, and the main function connecting both of my functions. He then asked me the time complexity of both functions and asked me to optimize the code for Dijkstra. I came up with the idea of priority queues(This priority queue optimization is a standard Dijkstra algorithm but I wasn't aware of that before so I thought of that there only).

Then the next day I got my results.

**Verdict:** Selected

### Preparation:

- I used to practice only from interview bit as suggested by my seniors
- For the google interview, I read the interview experiences of other people and did company-specific questions of google from GFG.
- This much of practice is enough to develop your thinking skills if you are doing qualitatively and honestly

### Tips:

- Most important thing is to remain calm and not mess up even the things that you know just because of nervousness.
- Be confident and believe in yourself.
- Even if you are unable to think of the whole solution just say what's on your mind

and along with the help of the interviewer and his hints you would be able to reach the solution.

- Just don't give up and freak out. Think that the interviewer is there to help you in reaching the solution.

My Personal Notes

Add your personal notes here

Save