

b'

## Amazon Interview Experience | Set 231 (On-Campus)

- Difficulty Level : [Medium](#)
- Last Updated : 02 Jul, 2019

We had Amazon campus interview in our college offering an internship (performance based job conversion).

### First round(online coding round) :

20 MCQs.

2 coding questions.

1. Given three huge numbers, each represented using a linked list (where each node of a linked list represents a digit), calculate the sum of the numbers and return back the number in the form of a linked list. For example,  $923, 46$  and  $251$  respectively. The result should be  $1 \rightarrow 2 \rightarrow 2 \rightarrow 0$ . This can be solved using stacks.
2. [Given a graph, detect if it has a cycle](#). We can perform a dfs here.

### Second round(Technical Interview):

I was asked to explain my work at an internship that I did. I explained for about 10 minutes. I was then given a binary search tree and asked to form a mirror of it i.e. every parent nodes left and right child should be swapped. I wrote a simple recursive code for it.

```
\r\nvoid BST(Node root) {\r\n    if (root==null)\r\n        return;\r\n    Node temp = root.left;\r\n    root.left = root.right;
```

Given a somewhat sorted array where the numbers are in ascending order first and then follow a descending order, find the index at which the order changes and write the number at that index. For example, i/p: 1234532, o/p: 5. I wrote down a modified binary search for it (had to take care of some corner cases). The interviewer was satisfied and I was sent to the next round.

### Third round(Technical Interview):

I was asked to talk about one of the projects. I spoke about an artificial intelligence project that I did. He asked me some questions based on it. About 15 minutes.

I was asked the same question as the first one in the online round except here I had to add only two numbers. I told him that but he asked me nonetheless to write the code down and explain it. I gave him the recursive solution as well as the easier stack-based solution as a replacement of recursion.

Given positive numbers  $a_1, a_2, \dots, a_n$ , you can sum up the elements in the following way:  $1*a(i\%n) + 2*a((i+1)\%n) + \dots + n*a((i+n)\%n)$  where  $i$  ranges from 0 to  $n-1$ . Find the value of  $i$  at which this sum would be the maximum. For example, 5, 6, 7 the various sums can be  $1*5 + 2*6 + 3*7$ ,  $1*6 + 2*7 + 3*5$ , and  $1*7 + 2*5 + 3*6$ . The answer is 38.

We can solve this in  $O(n)$ . We initially calculate the sum for  $i$  being 0. We make the following observation.

$i=0$ ,  $sum_0 = 1*a_1 + 2*a_2 + 3*a_3 + \dots + n*a_n$ .

$i=1$ ,  $sum_1 = n*a_1 + 1*a_2 + 2*a_3 + \dots + (n-1)*a_n$

$sum_1 = sum_0 \times (a_1 + a_2 + \dots + a_n) + n*a_1$

Generalizing the equation:  $sum_i \times (a_0 = sum_{i-1} \times (a_1 + a_2 + \dots + a_n) + n*a_i$ , where  $i$  ranges from 1 to  $n-1$ . The maximum of the  $sum_0$  to  $sum_{n-1}$  is the result.

Next question, you need to create an application which listens to requests continuously. The application has a common file that it can access. For every request, some information from the file has to be retrieved and returned back. Would you spawn a new process for each request or a new thread. And why? I said thread since the main processing to be done is on a shared resource and by creating threads you do not need make a copy of the file as such resources can be shared between threads.

What are the scheduling algorithms that you know in operating systems? What data structure would you use for each of the algorithms. I mentioned the following: First come First Serve : Queue.

Priority based scheduling : Priority Queue.

Shortest job First: Priority Queue ( this scheduling algorithm boils down to priority based scheduling)

Round Robin : Circular Queue. Interviewer told me circular linked list suits the purpose better. I added that circular double linked list would be an even better answer because of more efficient deletion of an ith node. He told me you can do this operation efficiently in a single linked list itself. I thought for a while and agreed with him while also showing him how I would go about doing this operation in a single linked list.

Given a computer with 1Gb ram and a 2Gb ram, the latter one would perform faster. Explain why would having more amount of ram would  $\times 93$  in normal circumstances- fare better than lesser ram powered computer. Answer : paging. I also told him how in certain computations like external sorting, it would also be more efficient if you have more RAM.

[Given a matrix of  \$1 \times 99\$ s and  \$0 \times 99\$ s, where 1 represents an island and 0 water, find the number of islands formed](#). The interviewer was expecting me to come up with a dfs solution similar to this one. (Unlike the problem given in link, the  $1 \times 99$ s can be connected only horizontally or vertically). But I had a quick union find(disjoint set problem) solution in mind.

### Fourth round(Technical Interview):

I was asked about my internship work, again. I described the same one as I did in the first round. He asked me some questions based on it. We spent about 20 minutes on it.

[Print the top view of a binary tree in order](#), from left to right. I took some time to figure it out. As I was arriving at the solution I kept explaining it out loud. I had previously done a problem on printing right view of a binary tree which had a similar approach.

Given a number  $n$ , print all the squares of numbers such that the square is less than equal to  $n$ . For example, if  $n$  is 30, o/p: 1 4 9 16 25. Do this without using  $*$ ,  $^$  or division operations.

If we observe the sequence we get from the output, to arrive at the next number in the sequence, we add the next odd number. So,  $0+1=1$ ,  $1+3=4$ ,  $4+5=9$ ,  $9+7=16$  and so on.

I have covered almost all of the questions asked to me. I had to wait for more than a week for the results. I got selected  $\times 9 \times 9 \times 82$

If you like GeeksforGeeks and would like to contribute, you can also write an article and mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

[All Practice Problems for Amazon !](#)

My Personal Notes [narrow\\_drop\\_up](#)

Add your personal notes here

Save

