

Contributor Name: Anonymous
Company Name: Microsoft
Profile: Software Development Engineer (SDE) Intern
Job Location: Bangalore
Applied[OnCampus/OffCampus]: off-campus

Round: 1

Detailed Round Description

Online Coding Interview focused on Data Structures and Algorithms (DSA) and Object-Oriented Programming (OOPS).

Detailed Question Description (with Test Cases, if possible)

Coding Problems: 3

1. Mean, Median, and Mode of Unsorted Array

Difficulty: Easy

Average Time: 10 minutes

Success Rate: 80%

Problem:

Given an array **ARR** of **N** integers, calculate the **Mean**, **Median**, and **Mode**.

- **Mean:** Average of elements.
- **Median:** Middle value when the array is sorted (or the average of two middle values).
- **Mode:** Most frequent value (smallest element in case of ties).

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]

Code Logic Block

```
from collections import Counter
from math import gcd

def mean_median_mode(arr):
    # Mean
    mean = sum(arr) / len(arr)
    mean_fraction = f"{int(mean * len(arr))}/{len(arr)}"

    # Median
    arr.sort()
    n = len(arr)
    if n % 2 == 0:
        median = (arr[n//2 - 1] + arr[n//2]) / 2
    else:
        median = arr[n//2]

    # Mode
    freq = Counter(arr)
    mode = min([k for k, v in freq.items() if v == max(freq.values())])

    return mean_fraction, median, mode
```

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]

2.Nth Term of GP Series

Difficulty: Easy

Average Time: 15 minutes

Success Rate: 90%

Problem:

Find the Nth term of a geometric progression (GP) series given the first term A , common ratio R , and integer N . Return the result modulo $10^9 + 7$.

Code Logic Block

```
def nth_term_gp(A, R, N):  
    MOD = 10**9 + 7  
    return (A * pow(R, N - 1, MOD)) % MOD
```

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]

3.Dice Throw

Difficulty: Hard

Average Time: 35 minutes

Success Rate: 65%

Problem:

Given D dice, each with F faces numbered from 1 to F , find the number of ways to achieve a sum S . Return the answer modulo $10^9 + 7$.

Code Logic Block

```
def num_rolls_to_target(D, F, S):  
    MOD = 10**9 + 7  
    dp = [[0] * (S + 1) for _ in range(D + 1)]  
    dp[0][0] = 1  
  
    for d in range(1, D + 1):  
        for s in range(1, S + 1):  
            dp[d][s] = sum(dp[d - 1][s - f] for f in range(1, F + 1) if s - f >= 0) % MOD  
  
    return dp[D][S]
```

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]

Round:2

Detailed Round Description

Online Coding Interview focused on Data Structures and Algorithms (DSA) and Object-Oriented Programming (OOPS).

Coding Problems:2

Detailed Question Description(with Test Cases, if possible)

Problem 1: Identical Trees

Difficulty: Moderate

Average Time: 20 minutes

Success Rate: 85%

Problem:

Check if two binary trees are identical.

Code Logic Block

```
def is_identical(tree1, tree2):  
    if not tree1 and not tree2:  
        return True  
    if tree1 and tree2 and tree1.val == tree2.val:  
        return is_identical(tree1.left, tree2.left) and is_identical(tree1.right, tree2.right)  
    return False
```

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]

Round:2

Detailed Round Description

Online Coding Interview focused on Data Structures and Algorithms (DSA) and Object-Oriented Programming (OOPS).

Coding Problems:2

Detailed Question Description(with Test Cases, if possible)

Problem 2: Distinct Subsequences of an Array

Difficulty: Moderate

Average Time: 10 minutes

Success Rate: 80%

Problem:

Count distinct subsequences of a string S . Return the count modulo 10^9+7 .

Code Logic Block

```
def distinct_subsequences(S):  
    MOD = 10**9 + 7  
    last_occurrence = {}  
    dp = [1]  
  
    for i, char in enumerate(S):  
        dp.append(dp[-1] * 2)  
        if char in last_occurrence:  
            dp[-1] -= dp[last_occurrence[char]]  
        dp[-1] %= MOD  
        last_occurrence[char] = i  
  
    return dp[-1]
```

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]

Round:3

Detailed Round Description

Online Coding Interview focused on Data Structures and Algorithms (DSA) and Object-Oriented Programming (OOPS).

Coding problems:1

Detailed Question Description(with Test Cases, if possible)

Problem 2: Delete Kth Node from End of the LinkList

Difficulty: Moderate

Average Time: 15 minutes

Success Rate: 95%

Description:

Delete Kth Node of a Given LinkList

Code Logic Block

```
def remove_kth_from_end(head, k):  
    fast, slow = head, head  
    for _ in range(k):  
        fast = fast.next  
  
    if not fast:  
        return head.next  
  
    while fast.next:  
        fast = fast.next  
        slow = slow.next  
  
    slow.next = slow.next.next  
    return head
```

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]

Round 4

Type: HR Round

Difficulty: Medium

Duration: 30 minutes

Interview Date: 10 Aug 2020

Questions:

1. What do you expect from this internship?
2. Where do you see yourself 5 years down the line?
3. What are your weaknesses?

LinuxSocials Interview Experiences Prep Resource

Each experience is unique in itself. And can help other developers who look up to you.

To contribute and help the LinuxSocials' Open Developer Community, use the [format](#) to write your experience and [mail it here](#). [You can check your resource contribution on: <https://www.linuxsocials.com>]