

LOGICAL

CODING QUESTIONS

For Interview Preparation

Two Sum

Given an array of integers, find two numbers such that they add up to a specific target number.

Sample Input:

`arr = [2, 7, 11, 15],`

`target_sum = 9`

Output: 2,7

Find a peak element which is larger than its neighbours

```
def find_peak_elements(nums):  
    peaks = []  
    array_len = len(nums)  
  
    #Checking edge elements  
    if(nums[0]>nums[1]):  
        peaks.append(nums[0])  
  
    if(nums[array_len-1]>nums[array_len-2]):  
        peaks.append(nums[array_len-1])  
  
    #Checking middle elements  
    for i in range(1, array_len-1):  
        if(nums[i]>nums[i-1] and nums[i]>nums[i+1]):  
            peaks.append(nums[i])  
    return peaks
```

LOGICAL

CODING QUESTIONS

For Interview Preparation

Find peak element

Given an array `arr` of `n` elements, find peak elements and return the index. A peak element is greater than its neighbours. You can imagine $\text{arr}[-1] = \text{arr}[n] = -\text{inf}$

Sample Input: [10, 20, 15, 2, 23, 90, 67]

Output: 1, 5

Given an array, move all zeros to the end while maintaining the relative order of the other elements.

Algorithm

Initial state:

input_array = [0, 2, 0, 4, 0, 1, 3, 0, 7]

index_of_zeros = 0

Iterations:

1. **i = 0:** arr[0] is 0 (zero), so we don't swap elements. Array remains unchanged.
2. **i = 1:** arr[1] is 2 (non-zero), so we swap arr[1] with arr[0]. Array becomes [2, 0, 0, 4, 0, 1, 3, 0, 7], index_of_zeros = 1.
3. **i = 2:** arr[2] is 0 (zero), so we don't swap elements. Array remains unchanged.
4. **i = 3:** arr[3] is 4 (non-zero), so we swap arr[3] with arr[1]. Array becomes [2, 4, 0, 0, 0, 1, 3, 0, 7], index_of_zeros = 2.
5. **i = 4:** arr[4] is 0 (zero), so we don't swap elements. Array remains unchanged.
6. **i = 5:** arr[5] is 1 (non-zero), so we swap arr[5] with arr[2]. Array becomes [2, 4, 1, 0, 0, 0, 3, 0, 7], index_of_zeros = 3.
7. **i = 6:** arr[6] is 3 (non-zero), so we swap arr[6] with arr[3]. Array becomes [2, 4, 1, 3, 0, 0, 0, 0, 7], index_of_zeros = 4.
8. **i = 7:** arr[7] is 0 (zero), so we don't swap elements. Array remains unchanged.
9. **i = 8:** arr[8] is 7 (non-zero), so we swap arr[8] with arr[4]. Array becomes [2, 4, 1, 3, 7, 0, 0, 0, 0], index_of_zeros = 5.

Final state:

input_array = [2, 4, 1, 3, 7, 0, 0, 0, 0]

index_of_zeros = 5

Given an array, move all zeros to the end while maintaining the relative order of the other elements.

Code

```
def move_zeros_to_end(arr):  
    index_of_zeros = 0  
  
    for i in range(len(arr)):  
        if(arr[i]!=0):  
            arr[i],arr[index_of_zeros] = arr[index_of_zeros],arr[i]  
            index_of_zeros += 1  
  
input_array = [0, 2, 0, 4, 0, 1, 3, 0, 7]  
move_zeros_to_end(input_array)  
print(input_array)
```

LOGICAL

CODING QUESTIONS

For Interview Preparation

Rotate Array

Rotate an array to the right by k steps, where k is non-negative.

Sample Input: arr:[1, 2, 3, 4, 5], k: 3

Output: [3, 4, 5, 1, 2]

Find and print the duplicate elements in an array

```
def find_duplicates(arr):  
    number_count = {}  
    for num in arr:  
        number_count[num] = number_count.get(num,0)+1  
  
    return [key for key, value in number_count.items() if value>1]  
  
print(find_duplicates(arr))
```