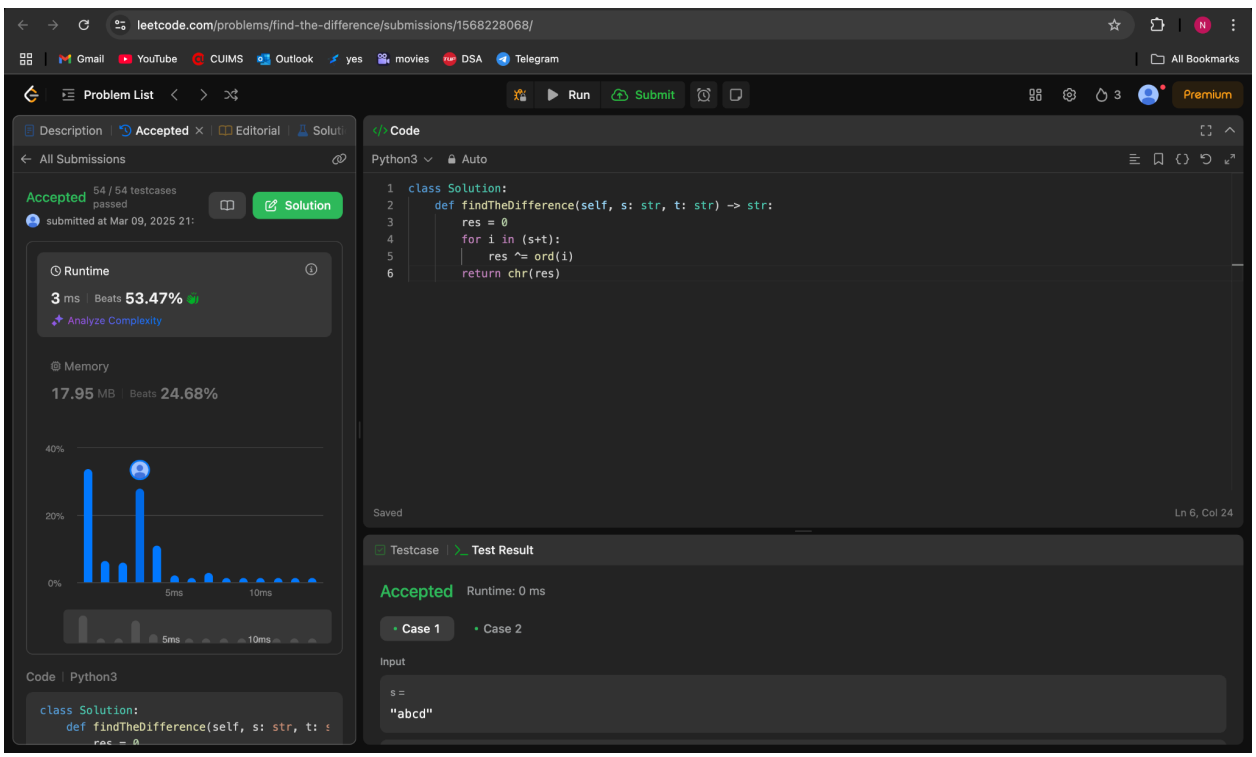
AP ASSIGNMENT - 5

Name : Naman Kumar
UID : 22ICS10001
Class : FL_IOT_604 (A)

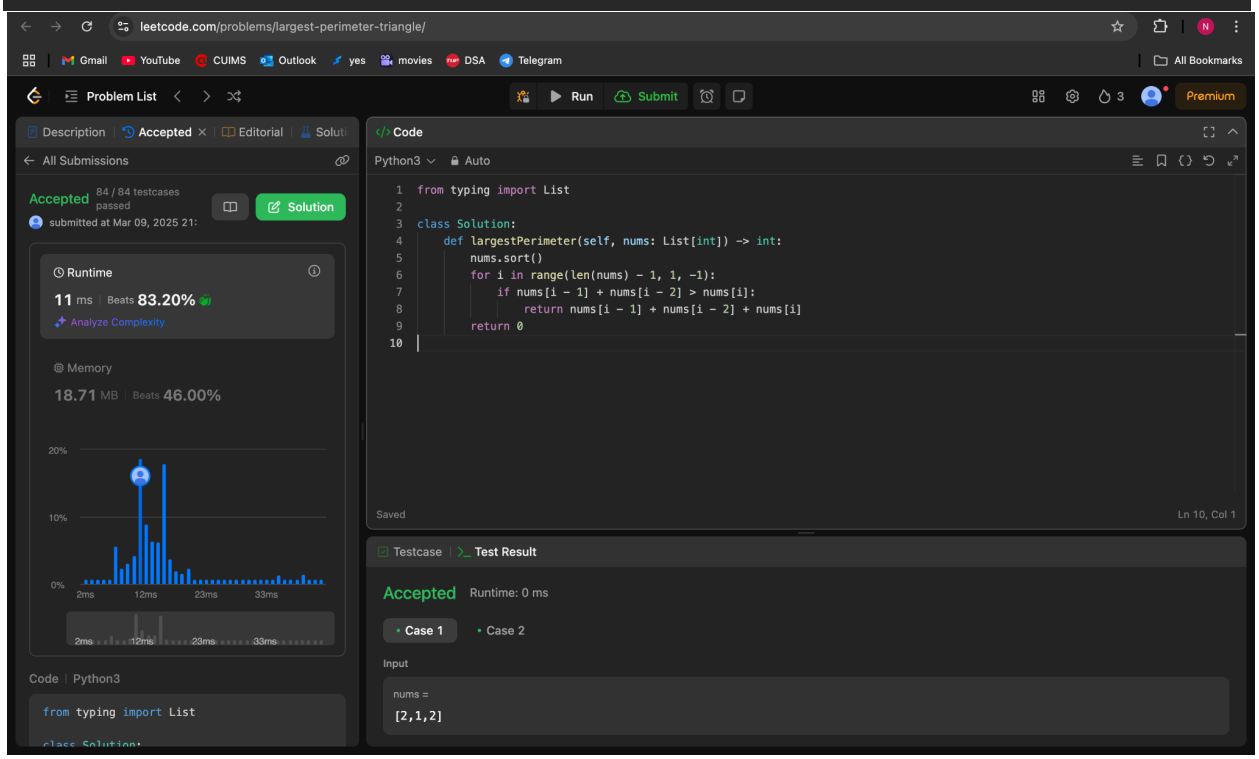
389. Find the Difference

```python
class Solution:
    def findTheDifference(self, s: str, t: str) -> str:
        res = 0
        for i in (s+t):
            res ^= ord(i)
        return chr(res)
```



976. Largest Perimeter Triangle

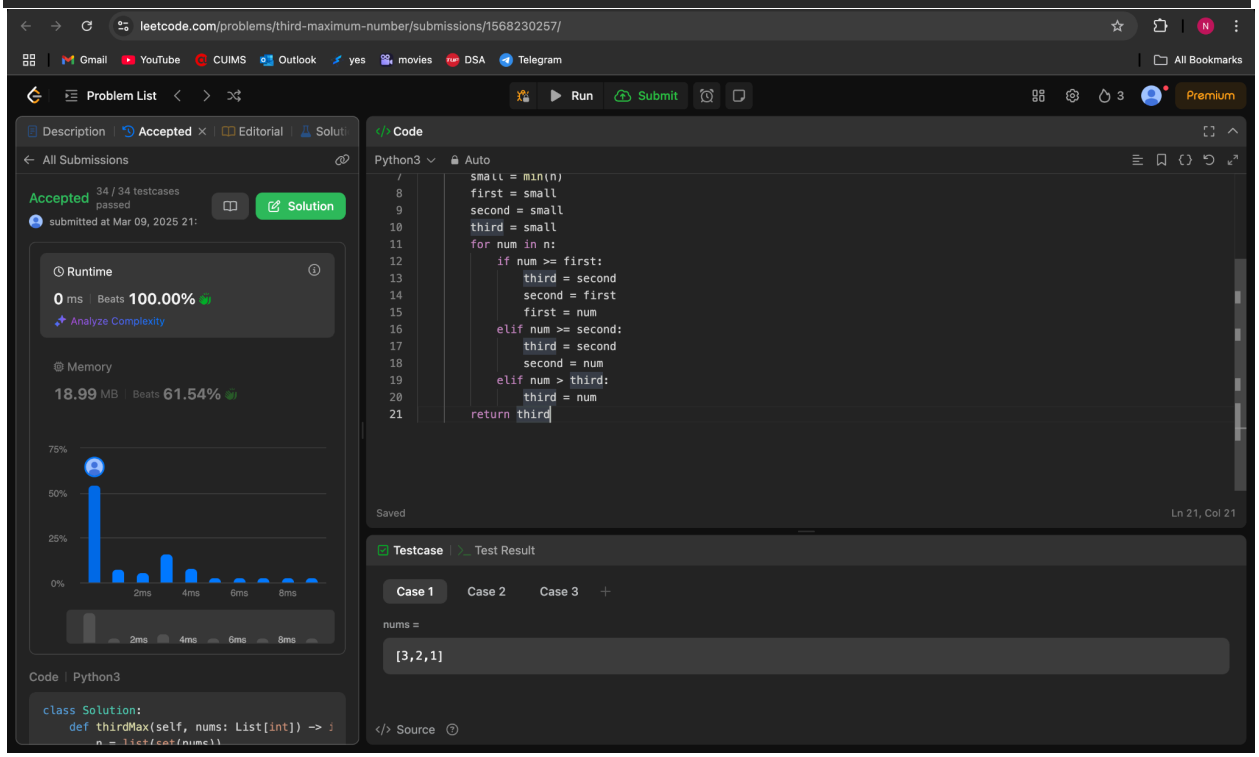
```python
from typing import List

class Solution:
    def largestPerimeter(self, nums: List[int]) -> int:
        nums.sort()
        for i in range(len(nums) - 1, 1, -1):
            if nums[i - 1] + nums[i - 2] > nums[i]:
                return nums[i - 1] + nums[i - 2] + nums[i]
        return 0
```



414. Third Maximum Number

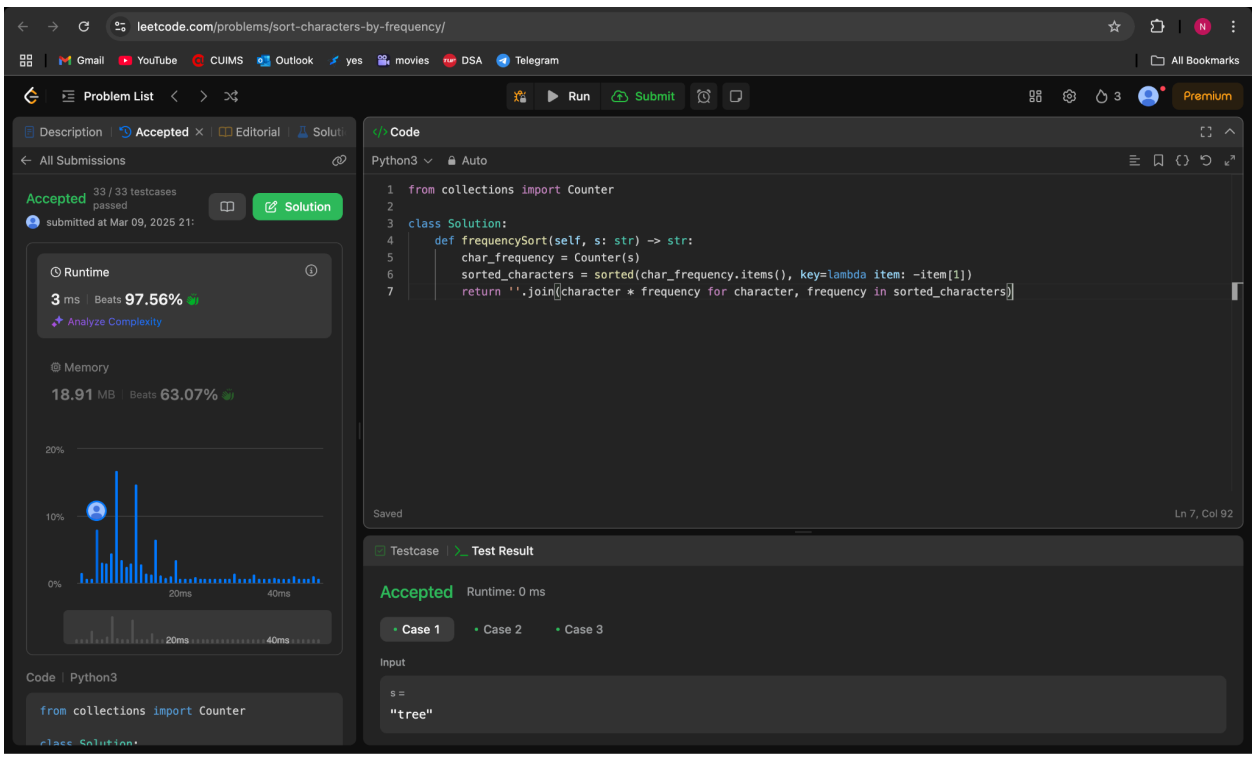
```python
cclass Solution:
    def thirdMax(self, nums: List[int]) -> int:
        n = list(set(nums))
        if len(n) == 1: return n[0]
        if len(n) == 2: return max(n)
```

```python
        small = min(n)
        first = small
        second = small
        third = small
        for num in n:
            if num >= first:
                third = second
                second = first
                first = num
            elif num >= second:
                third = second
                second = num
            elif num > third:
                third = num
        return third
```



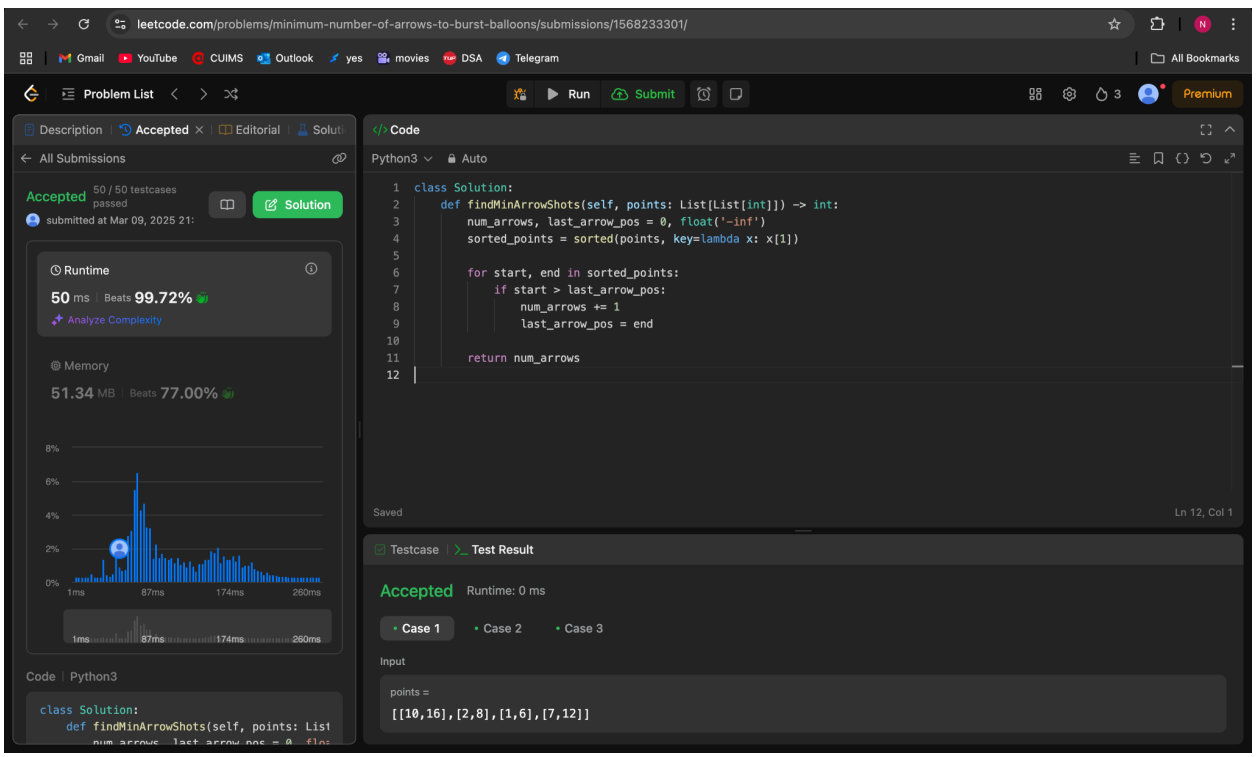
451. Sort Characters By Frequency

```python
from collections import Counter

class Solution:
    def frequencySort(self, s: str) -> str:
        char_frequency = Counter(s)
        sorted_characters = sorted(char_frequency.items(), key=lambda item: -item[1])
        return ''.join(character * frequency for character, frequency in sorted_characters)
```



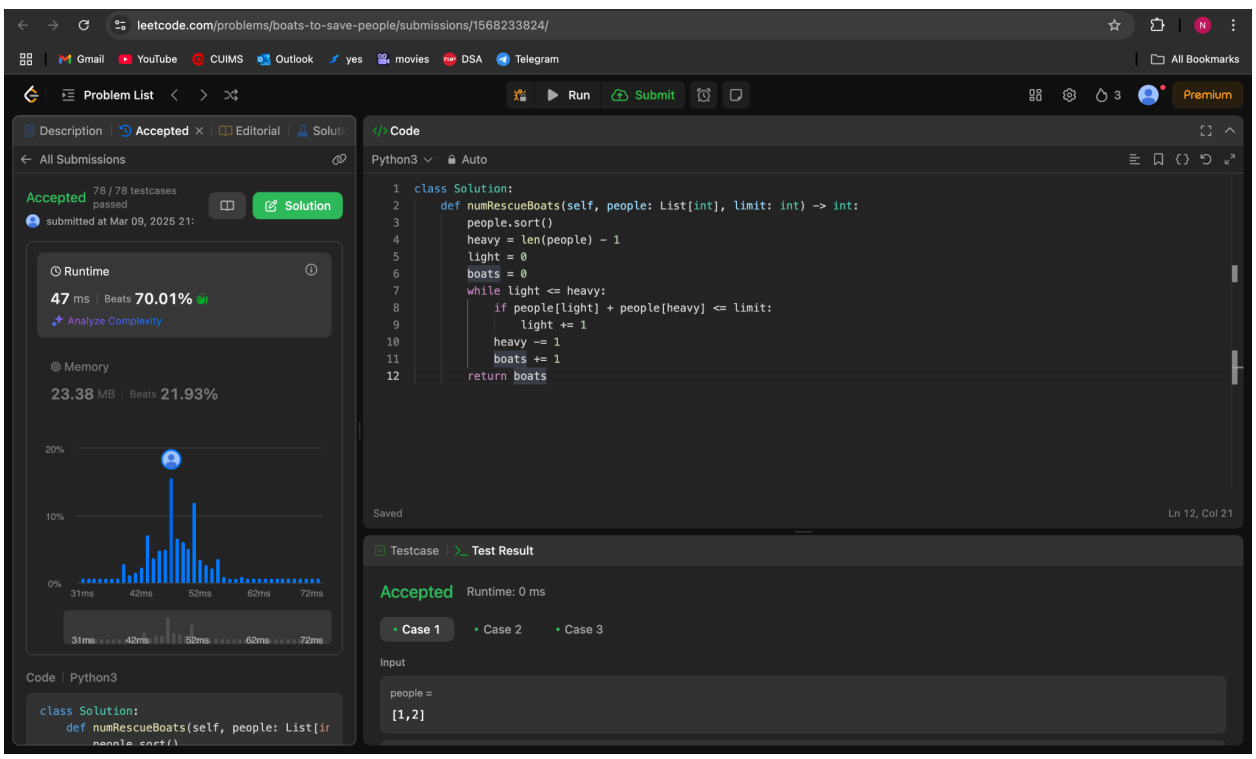
452. Minimum Number of Arrows to Burst Balloons

```python
class Solution:
    def findMinArrowShots(self, points: List[List[int]]) -> int:
        num_arrows, last_arrow_pos = 0, float('-inf')
        sorted_points = sorted(points, key=lambda x: x[1])

        for start, end in sorted_points:
            if start > last_arrow_pos:
                num_arrows += 1
                last_arrow_pos = end

        return num_arrows
```



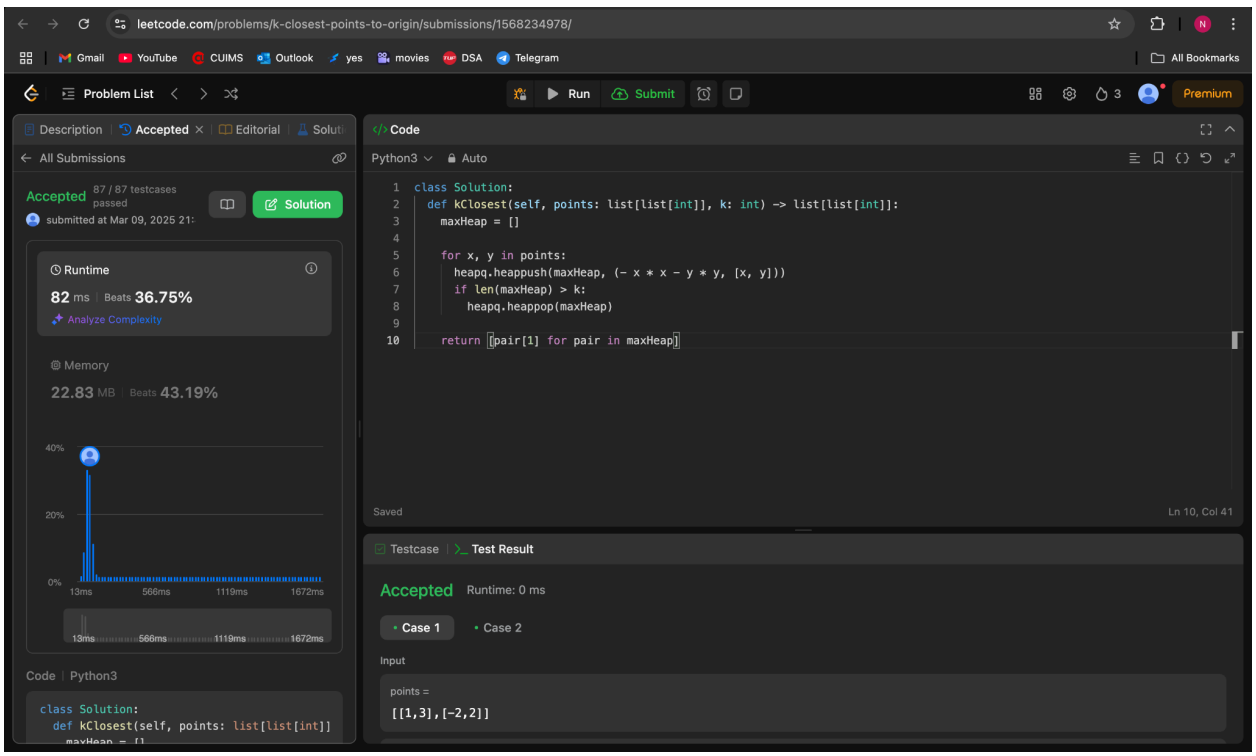
881. Boats to Save People

```python
class Solution:
    def numRescueBoats(self, people: List[int], limit: int) -> int:
        people.sort()
        heavy = len(people) - 1
        light = 0
        boats = 0
        while light <= heavy:
            if people[light] + people[heavy] <= limit:
                light += 1
            heavy -= 1
            boats += 1
        return boats
```



973. K Closest Points to Origin

```python
class Solution:
  def kClosest(self, points: list[list[int]], k: int) -> list[list[int]]:
    maxHeap = []

    for x, y in points:
      heapq.heappush(maxHeap, (- x * x - y * y, [x, y]))
      if len(maxHeap) > k:
        heapq.heappop(maxHeap)

    return [pair[1] for pair in maxHeap]
```



1338. Reduce Array Size to The Half

```python
from typing import List
from collections import Counter

class Solution:
    def minSetSize(self, arr: List[int]) -> int:
        frequency_counter = Counter(arr)
        min_set_size = 0
        removed_count = 0

        for _, frequency in frequency_counter.most_common():
            removed_count += frequency
            min_set_size += 1
            if removed_count * 2 >= len(arr):
                break

        return min_set_size
```