Day-16 Python DSA

Missing and Repeating

https://www.geeksforgeeks.org/problems/find-missing-and-repeating2512/1

Bruteforce

```
def findTwoElement(arr):
    repeating, missing = -1, -1
    n = len(arr)

    for i in range(1, n + 1):
        count = 0
        for j in range(n):
            if arr[j] == i:
                count += 1

        if count == 2:
            repeating = i
        elif count == 0:
            missing = i

        if repeating != -1 and missing != -1:
            break
```

```
    return [repeating, missing]



arr = [4, 3, 6, 2, 1, 1]
findTwoElement(arr)
```

TC – O(N^2)

SC- O(1)

Better

```
def findTwoElement(arr):
    repeating, missing=-1,-1
    n = len(arr)
    hash_list = [0] * (n+1)
    for num in arr:
        hash_list[num] +=1
    for i in range(1, len(hash_list)):
        if hash_list[i] ==2:
            repeating =i
        elif hash_list[i]==0:
            missing =i
        if repeating !=-1 and missing!=-1:
            return [repeating, missing]
arr = [4, 3, 6, 2, 1, 1]
findTwoElement(arr)
```

TC- O(N)

SC-O(N)

Optimal

```
class Solution: def findTwoElement(self, arr):
  n= len(arr)
  sN = (n * (n + 1)) // 2
  s2N = (n * (n + 1) * (2 * n + 1)) // 6
  s = 0
   s2 = 0
   for num in arr:
      s += num
      s2 += num**2

   val1 = s - sN  # x-y
   val2 = s2 - s2N  # x^2 - y^2

   # This is x+y
   val2 = val2 // val1  # (x^2 - y^2)// x-y
   x = (val1 + val2) // 2
   y = x - val1
   return [x,y]
```

TC- O(N)

SC-O(1)