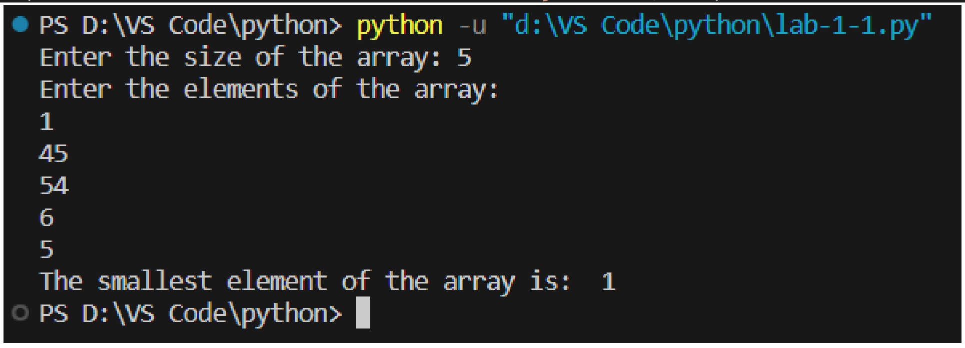
Lab Sheet 1

Name: Vinayak V Thayil
Roll No: AM.EN.U4CSE21161

1. Write a program to find the smallest element of an unsorted array of size N. Don't use any built-in functions/ methods like min() supported by certain programming languages.

```
arr=[]
n=int(input("Enter the size of the array: "))
print("Enter the elements of the array: ")
for i in range(n):
    arr.append(int(input()))
min=arr[0]
for i in range(n):
    if arr[i]<min:
        min=arr[i]
print("The smallest element of the array is: ",min)
```

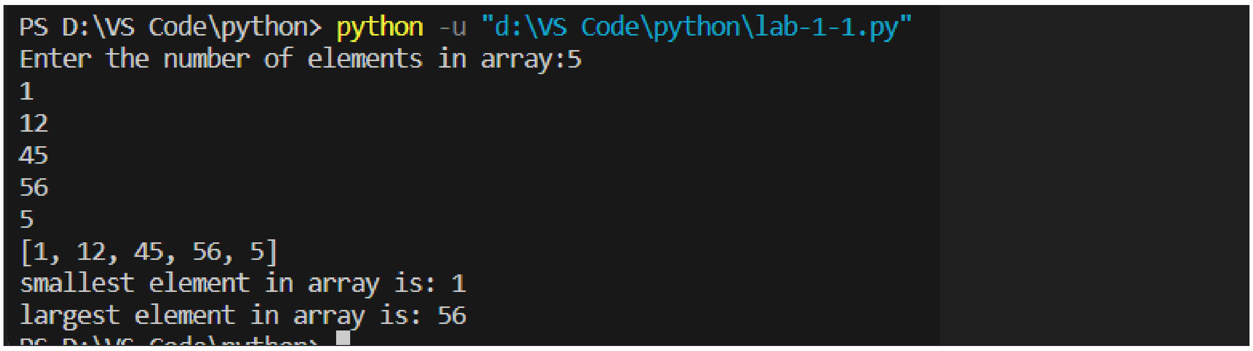


```
● PS D:\VS Code\python> python -u "d:\VS Code\python\lab-1-1.py"
  Enter the size of the array: 5
  Enter the elements of the array:
  1
  45
  54
  6
  5
  The smallest element of the array is:  1
○ PS D:\VS Code\python> 
```

   a. How many times does your loop execute?
      n-1 times
   b. As the elements in the array changes (the size of the array remains same), will there be any change in the number of times the loop executes? What is the minimum and maximum number of times the loop executes?
      No Change and the minimum and maximum number is n-1
   c. What is the time complexity of your program?
      O(n)

2. Write a program to find the smallest and the largest element in a sorted array.

```
arr=[]
n=int(input("Enter the number of elements in array:"))
for i in range(0,n):
    ele=int(input())
    arr.append(ele)
print(arr)
print("smallest element in array is:",min(arr))
print("largest element in array is:",max(arr))
```



```
PS D:\VS Code\python> python -u "d:\VS Code\python\lab-1-1.py"
Enter the number of elements in array:5
1
12
45
56
5
[1, 12, 45, 56, 5]
smallest element in array is: 1
largest element in array is: 56
PS D:\VS Code\python>
```

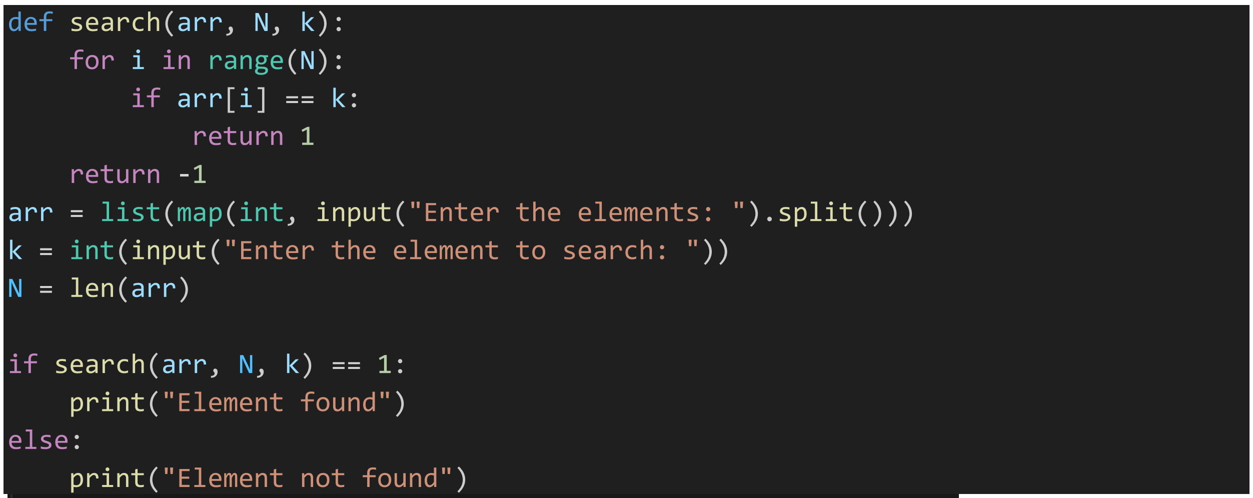
a. need any loops in this program?

No

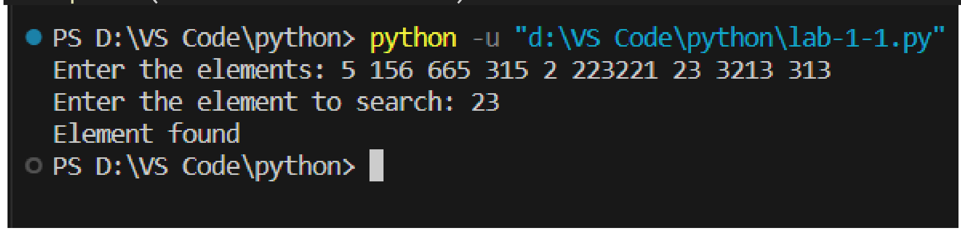
b. What is the time complexity of your program?

O(n)

3. Write a program to search an element ‘k’ in an unsorted array of size N.



```python
def search(arr, N, k):
    for i in range(N):
        if arr[i] == k:
            return 1
    return -1
arr = list(map(int, input("Enter the elements: ").split()))
k = int(input("Enter the element to search: "))
N = len(arr)

if search(arr, N, k) == 1:
    print("Element found")
else:
    print("Element not found")
```



```
● PS D:\VS Code\python> python -u "d:\VS Code\python\lab-1-1.py"
  Enter the elements: 5 156 665 315 2 223221 23 3213 313
  Enter the element to search: 23
  Element found
○ PS D:\VS Code\python> █
```

a. As the elements in the array changes (the size of the array remains same), will there be any change in the number of times the loop executes? What is the minimum and maximum number of times the loop executes?

Yes.

Maximum→O(n)

Minimum→1(1)

b. What is the time complexity of your program?

O(n)

4. We need to search an element ‘k’ in a sorted array of size N.

a. Will your program for Qn. No 3 work for this case?

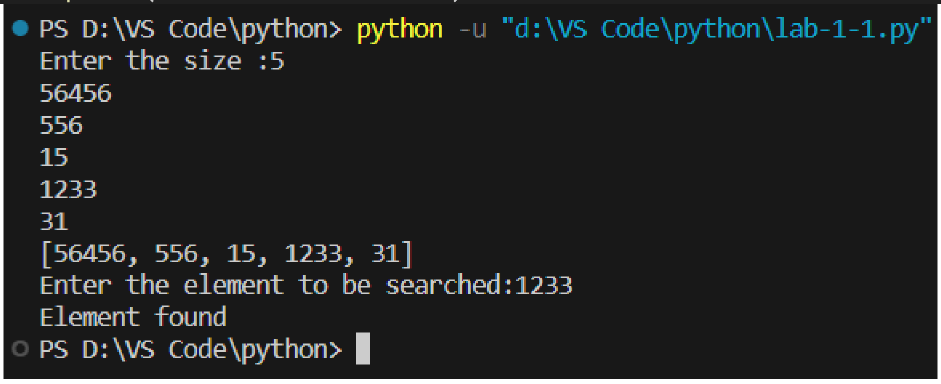
Yes

b. Is the program for Qn. No 3 the most efficient one for this? (Hint: There exists a Binary search algorithm)

No

c. Write an iterative program to implement Binary search.

```
arr=[]
n=int(input("Enter the size :"))
for i in range(0,n):
    ele=int(input())
    arr.append(ele)
print(arr)
k=int(input("Enter the element to be searched:"))
low=0
high=n-1
mid=0
while(low<=high):
    mid=(low+high)//2
    if(arr[mid]<k):
        low=mid+1
    elif(arr[mid]>k):
        high=mid-1
    else:
        print("Element found at index:")
        break
if(low>high):
    print("Element not found")
```



```
PS D:\VS Code\python> python -u "d:\VS Code\python\lab-1-1.py"
Enter the size :5
56456
556
15
1233
31
[56456, 556, 15, 1233, 31]
Enter the element to be searched:1233
Element found
PS D:\VS Code\python>
```

d. As the elements in the array changes (the size of the array remains same), will there be any change in the number of times the loop executes? What is the minimum and maximum number of times the loop executes?

Yes.

Minimum→O(1)

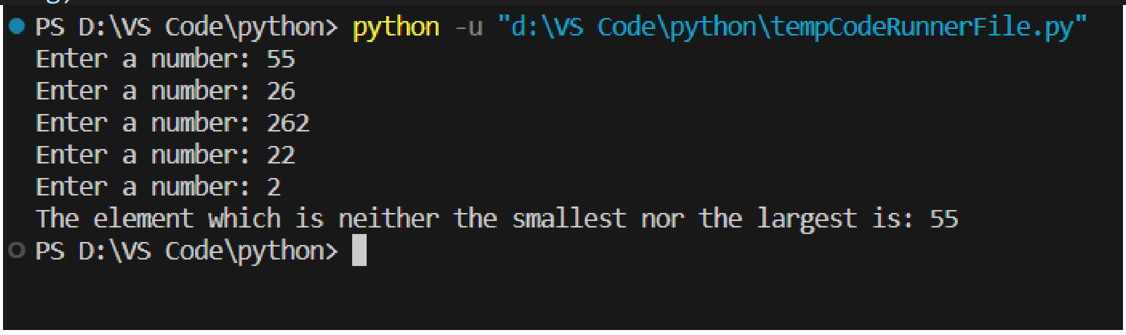
Maximum→1(log n)

e. What is the time complexity of your program?

O(log n)

5. Write an efficient program to find an element in an array which neither the smallest nor the largest. (Hint: you can do this without a loop.)

```
arr = []
for i in range(5):
    num = int(input("Enter a number: "))
    arr.append(num)
smallest = min(arr)
largest = max(arr)
arr.remove(smallest)
arr.remove(largest)
remaining = arr[0]
print("The element which is neither the smallest nor the largest is:",
remaining)
```

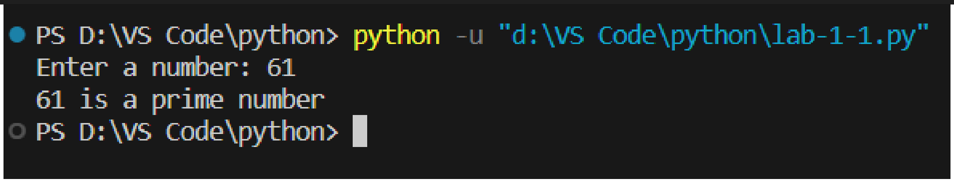


```
PS D:\VS Code\python> python -u "d:\VS Code\python\tempCodeRunnerFile.py"
Enter a number: 55
Enter a number: 26
Enter a number: 262
Enter a number: 22
Enter a number: 2
The element which is neither the smallest nor the largest is: 55
PS D:\VS Code\python>
```

a. What is the time complexity of your program?
   O(1)

6. Write an efficient program to check if a given number is prime or not.

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, n):
        if n % i == 0:
           return False
    return True
n = int(input("Enter a number: "))
if is_prime(n):
    print(n, "is a prime number")
else:
    print(n, "is not a prime number")
```



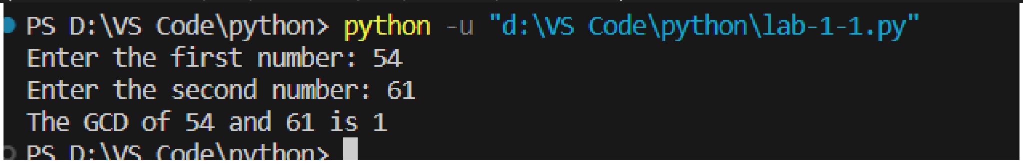
```
PS D:\VS Code\python> python -u "d:\VS Code\python\lab-1-1.py"
Enter a number: 61
61 is a prime number
PS D:\VS Code\python>
```

a. What is the time complexity of the algorithm?
   O(sqrt(n))
b. Show that the problem can be solved in root(n) time.

```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, n):
        if n % i == 0:
           return False
    return True
n = int(input("Enter a number: "))
if is_prime(n):
    print(n, "is a prime number")
else:
    print(n, "is not a prime number")
```

7. Write an efficient program to find the GCD (also called HCF) of two given numbers.

```python
def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
result = gcd(a, b)
print("The GCD of", a, "and", b, "is", result)
```



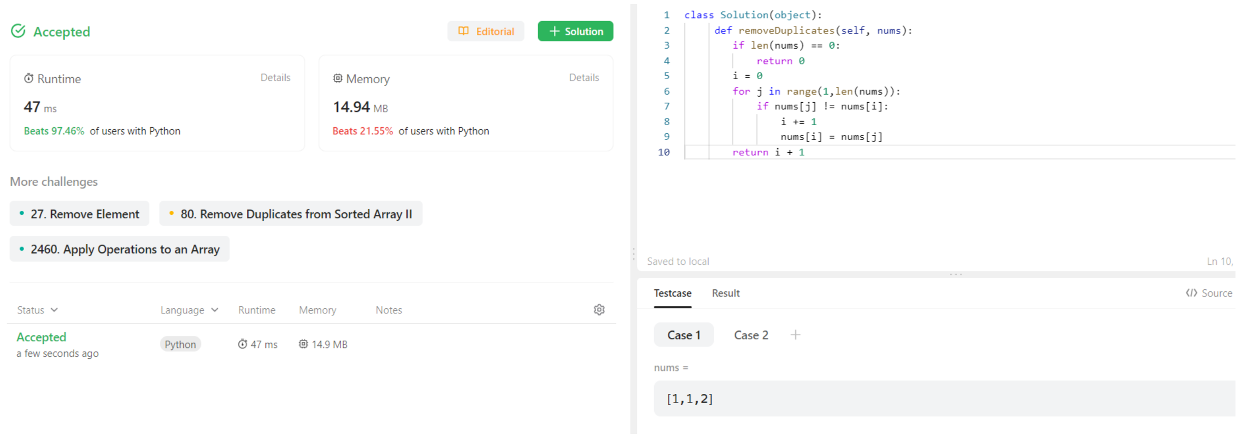
```
PS D:\VS Code\python> python -u "d:\VS Code\python\lab-1-1.py"
Enter the first number: 54
Enter the second number: 61
The GCD of 54 and 61 is 1
PS D:\VS Code\python>
```

   a. What is the time complexity of the algorithm?
      O(log(min(a,b)))
   b. Find an input that requires maximum number of iterations to solve.
      The input that requires the maximum number of iterations to solve using the Euclidean algorithm is when the two input numbers are consecutive Fibonacci numbers.

8. You are given a sorted array A of size n. Write an iterative program to remove the duplicates from the array. For example, if A[] = {2, 7, 7, 11, 24, 24, 24, 29, 36, 36}, your output should be B[] = {2, 7, 11, 24, 29, 36}.
   a. Count the operations to get the closed-form equation of running time (worst case).

```python
class Solution(object):
    def removeDuplicates(self, nums):
        if len(nums) == 0:
            return 0
        i = 0
        for j in range(1,len(nums)):→n-1
            if nums[j] != nums[i]:→n-1
                i += 1
                nums[i] = nums[j]
        return i + 1
```

   b. Submit the program for the problem [https://leetcode.com/problems/remove- duplicates-from-sorted-array/](https://leetcode.com/problems/remove-duplicates-from-sorted-array/) and submit the snapshot of acceptance as proof.



Accepted Editorial Solution

Runtime Details
47 ms
Beats 97.46% of users with Python

Memory Details
14.94 MB
Beats 21.55% of users with Python

More challenges
- 27. Remove Element
- 80. Remove Duplicates from Sorted Array II
- 2460. Apply Operations to an Array

Status	Language	Runtime	Memory	Notes
Accepted a few seconds ago	Python	47 ms	14.9 MB	

```python
class Solution(object):
    def removeDuplicates(self, nums):
        if len(nums) == 0:
            return 0
        i = 0
        for j in range(1,len(nums)):
            if nums[j] != nums[i]:
                i += 1
                nums[i] = nums[j]
        return i + 1
```

Saved to local Ln 10,

Testcase Result Source
Case 1 Case 2
nums =
[1,1,2]

   c. What is the time complexity?
      O(n)

9. Consider an array A of size n. Split A[] into the two arrays Low[] and High[] such that Low[] contains all elements < A[0] and High[] contains all elements >= A[0].
    a. Write an iterative algorithm and implement it.

```
def split_array(A):
    Low = []
    High = []
    pivot = A[0]
    for x in A[1:]:
        if x < pivot:
            Low.append(x)
        else:
            High.append(x)
    return Low, High
```

    b. What is the time complexity?

O(nlogn)

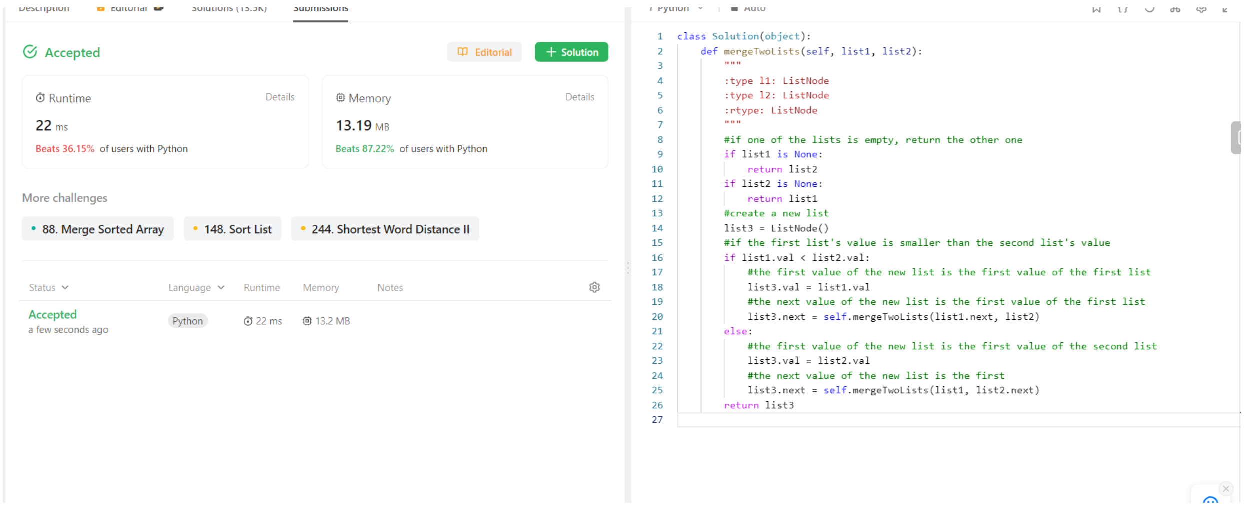
10. Given two sorted lists A[1..n] and B[1..n], write an algorithm to merge them into a single sorted list C[1..2n]. For example, if A[] = {1,3,6,7} and B[] = {2,4,5,8}, then C[] = {1,2,3,4,5,6,7,8}.

```
def merge_sorted_lists(A, B):
    i = 0
    j = 0
    n = len(A)
    m = len(B)
    C = []
    while i < n and j < m:
        if A[i] <= B[j]:
            C.append(A[i])
            i += 1
        else:
            C.append(B[j])
            j += 1
    if i < n:
        C.extend(A[i:])
    if j < m:
        C.extend(B[j:])
    return C
```

    a. Find the complexity

O(len(A) + len(B))

    b. Submit the program for the problem https://leetcode.com/problems/merge-two- sorted-lists/ and submit the snapshot of acceptance as proof



Runtime 22 ms Beats 36.15% of users with Python

Memory 13.19 MB Beats 87.22% of users with Python

More challenges: 88. Merge Sorted Array · 148. Sort List · 244. Shortest Word Distance II

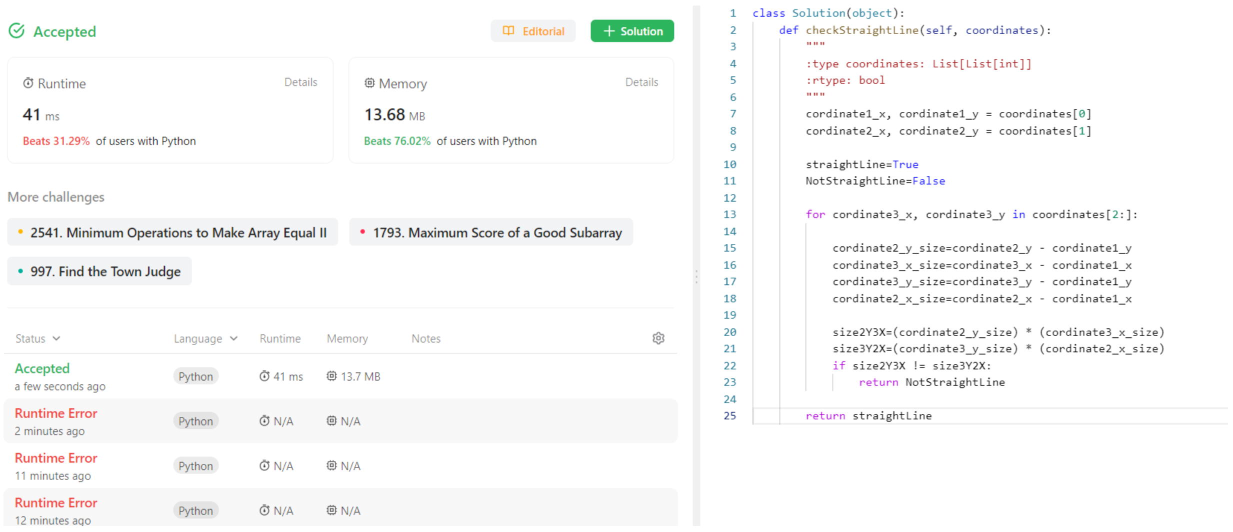
Status	Language	Runtime	Memory	Notes
Accepted a few seconds ago	Python	22 ms	13.2 MB	

```
class Solution(object):
    def mergeTwoLists(self, list1, list2):
        """
        :type l1: ListNode
        :type l2: ListNode
        :rtype: ListNode
        """
        #if one of the lists is empty, return the other one
        if list1 is None:
            return list2
        if list2 is None:
            return list1
        #create a new list
        list3 = ListNode()
        #if the first list's value is smaller than the second list's value
        if list1.val < list2.val:
            #the first value of the new list is the first value of the first list
            list3.val = list1.val
            #the next value of the new list is the first value of the first list
            list3.next = self.mergeTwoLists(list1.next, list2)
        else:
            #the first value of the new list is the first value of the second list
            list3.val = list2.val
            #the next value of the new list is the first
            list3.next = self.mergeTwoLists(list1, list2.next)
        return list3

```

11. You are given an array coordinates, cord[i] = [x, y], where [x, y] represents the coordinate of a point. Check if these points make a straight line in the XY plane.

```
     def checkStraightLine(coordinates):
    if len(coordinates) <= 2:
        return True
    x0, y0 = coordinates[0]
    x1, y1 = coordinates[1]
    for i in range(2, len(coordinates)):
        xi, yi = coordinates[i]
        if (yi - y0) * (x1 - x0) != (y1 - y0) * (xi - x0):
            return False
    return True
```

a. Find the time complexity of the algorithm.
   O (len(array))
b. Submit the program for the problem https://leetcode.com/problems/check-if-it-is-a- straight-line/ and submit the snapshot of acceptance as proof.



Runtime 41 ms Beats 31.29% of users with Python

Memory 13.68 MB Beats 76.02% of users with Python

More challenges: 2541. Minimum Operations to Make Array Equal II · 1793. Maximum Score of a Good Subarray · 997. Find the Town Judge

Status	Language	Runtime	Memory	Notes
Accepted a few seconds ago	Python	41 ms	13.7 MB	
Runtime Error 2 minutes ago	Python	N/A	N/A	
Runtime Error 11 minutes ago	Python	N/A	N/A	
Runtime Error 12 minutes ago	Python	N/A	N/A	

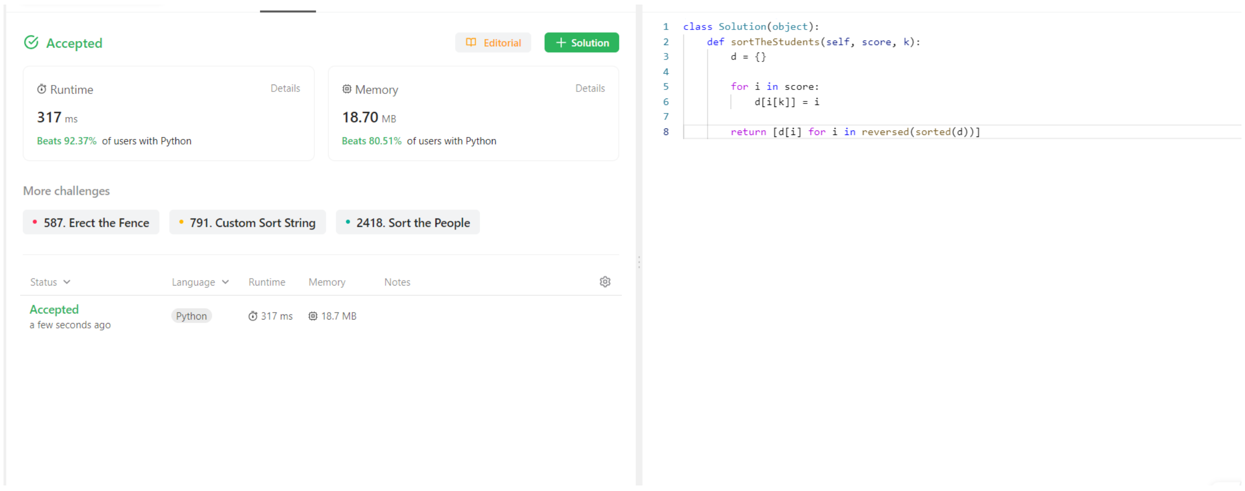
```
class Solution(object):
    def checkStraightLine(self, coordinates):
        """
        :type coordinates: List[List[int]]
        :rtype: bool
        """
        cordinate1_x, cordinate1_y = coordinates[0]
        cordinate2_x, cordinate2_y = coordinates[1]

        straightLine=True
        NotStraightLine=False

        for cordinate3_x, cordinate3_y in coordinates[2:]:

            cordinate2_y_size=cordinate2_y - cordinate1_y
            cordinate3_x_size=cordinate3_x - cordinate1_x
            cordinate3_y_size=cordinate3_y - cordinate1_y
            cordinate2_x_size=cordinate2_x - cordinate1_x

            size2Y3X=(cordinate2_y_size) * (cordinate3_x_size)
            size3Y2X=(cordinate3_y_size) * (cordinate2_x_size)
            if size2Y3X != size3Y2X:
                return NotStraightLine

        return straightLine
```

12. There is a class with m students and n exams. You are given a 0-indexed m x n integer matrix called score, where score[i][j] denotes the score the ith student got in the jth exam. The matrix score contains distinct integers only. You are also given an integer k. Sort the students (i.e., the rows of the matrix) by their scores in the kth (0-indexed) exam from the highest to the lowest. Return the matrix after sorting it.

```
def sort_scores(score, k):
    return sorted(score, key=lambda x: x[k], reverse=True)
```

a. Find the time complexity
   O(N log N)

b. Submit the program for the problem https://leetcode.com/problems/sort-the- students-by-their-kth-score/ and submit the snapshot of acceptance as proof.



Accepted

Runtime Details
317 ms
Beats 92.37% of users with Python

Memory Details
18.70 MB
Beats 80.51% of users with Python

```
Status          Language   Runtime   Memory    Notes
Accepted        Python     317 ms    18.7 MB
a few seconds ago
```

```python
class Solution(object):
    def sortTheStudents(self, score, k):
        d = {}

        for i in score:
            d[i[k]] = i

        return [d[i] for i in reversed(sorted(d))]
```