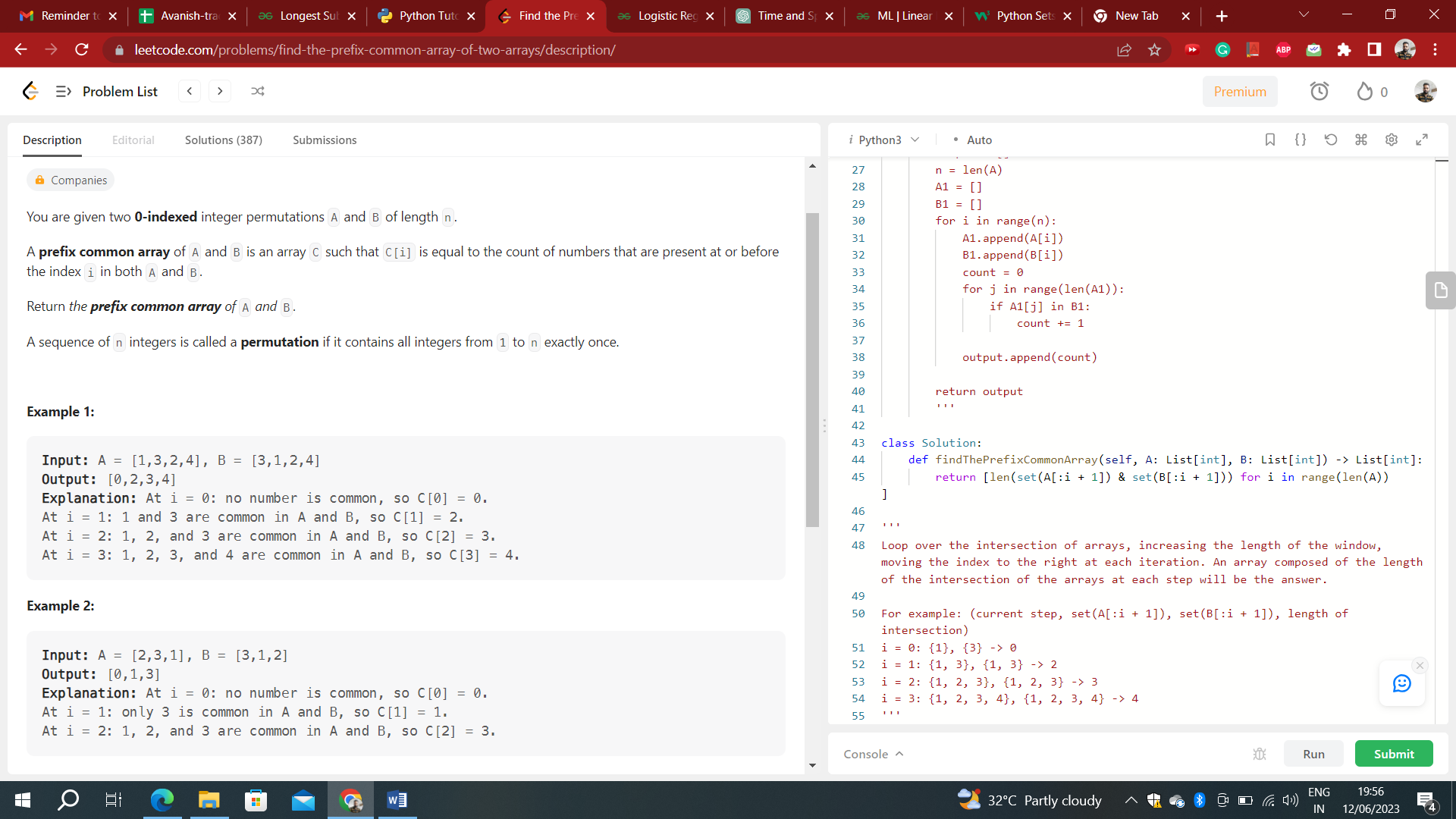
**Find the Prefix Common Array of Two Arrays**

<https://leetcode.com/problems/find-the-prefix-common-array-of-two-arrays/description/>

**

**Time Complexity:** O(N)  
**Auxiliary Space:** O(N)

'''

class Solution:

def findThePrefixCommonArray(self, A: List[int], B: List[int]) -> List[int]:

a = set()

b = set()

ans = []

count = 0

for i in range(0, len(A)):

a.add(A[i])

b.add(B[i])

if A[i] == B[i]:

count += 1

ans.append(count)

continue

if A[i] in b:

count += 1

if B[i] in a:

count += 1

ans.append(count)

return ans

'''

**Time Complexity:** O(N2)  
**Auxiliary Space:** O(N)

'''

class Solution:

def findThePrefixCommonArray(self, A: List[int], B: List[int]) -> List[int]:

output = []

n = len(A)

A1 = []

B1 = []

for i in range(n):

A1.append(A[i])

B1.append(B[i])

count = 0

for j in range(len(A1)):

if A1[j] in B1:

count += 1

output.append(count)

return output

'''

**Time Complexity:** O(N)  
**Auxiliary Space:** O(N)

class Solution:

def findThePrefixCommonArray(self, A: List[int], B: List[int]) -> List[int]:

return [len(set(A[:i + 1]) & set(B[:i + 1])) for i in range(len(A))]

'''

Loop over the intersection of arrays, increasing the length of the window, moving the index to the right at each iteration. An array composed of the length of the intersection of the arrays at each step will be the answer.

For example: (current step, set(A[:i + 1]), set(B[:i + 1]), length of intersection)

i = 0: {1}, {3} -> 0

i = 1: {1, 3}, {1, 3} -> 2

i = 2: {1, 2, 3}, {1, 2, 3} -> 3

i = 3: {1, 2, 3, 4}, {1, 2, 3, 4} -> 4