744. Find Smallest Letter Greater Than Target

Given a list of sorted characters letters containing only lowercase letters, and given a target letter target, find the smallest element in the list that is larger than the given target.

Letters also wrap around. For example, if the target is target = 'z' and letters = ['a', 'b'], the answer is 'a'.

Answer:

class Solution:

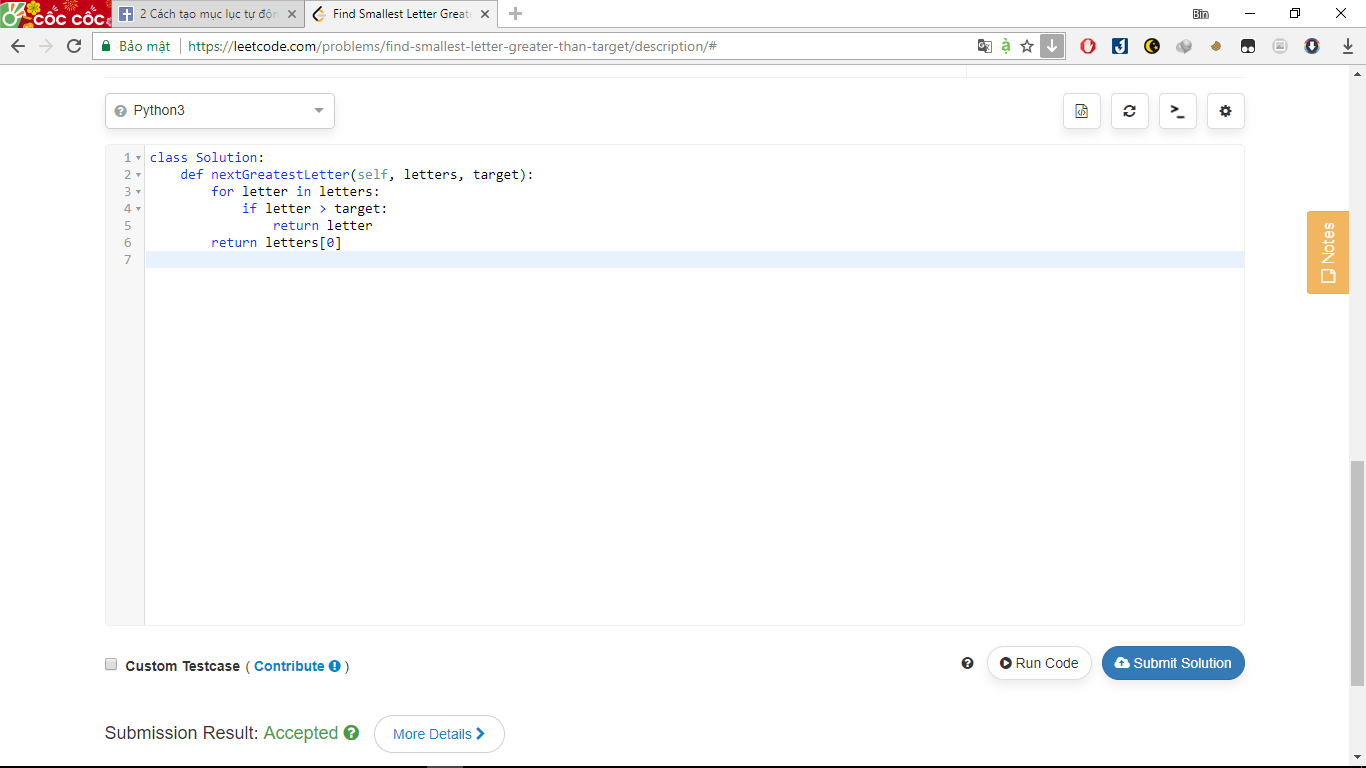
def nextGreatestLetter(self, letters, target):

for letter in letters:

if letter > target:

return letter

return letters[0]



628. Maximum Product of Three Numbers

Given an integer array, find three numbers whose product is maximum and output the maximum product.

**Example 1:**

**Input:** [1,2,3]

**Output:** 6

**Example 2:**

**Input:** [1,2,3,4]

**Output:** 24

**Note:**

1. The length of the given array will be in range [3,104] and all elements are in the range [-1000, 1000].
2. Multiplication of any three numbers in the input won't exceed the range of 32-bit signed integer.

answer:

class Solution:

def maximumProduct(self, A):

A.sort()

del A[3:-3]

return max(a \* b \* c for a, b, c in itertools.combinations(A, 3))



27. Remove Element

Given an array and a value, remove all instances of that value [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) and return the new length.

Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

The order of elements can be changed. It doesn't matter what you leave beyond the new length.

**Example:**

Given **nums** = [3,2,2,3], **val** = 3,

Your function should return length = 2, with the first two elements of *nums* being 2.

Answer :

class Solution:

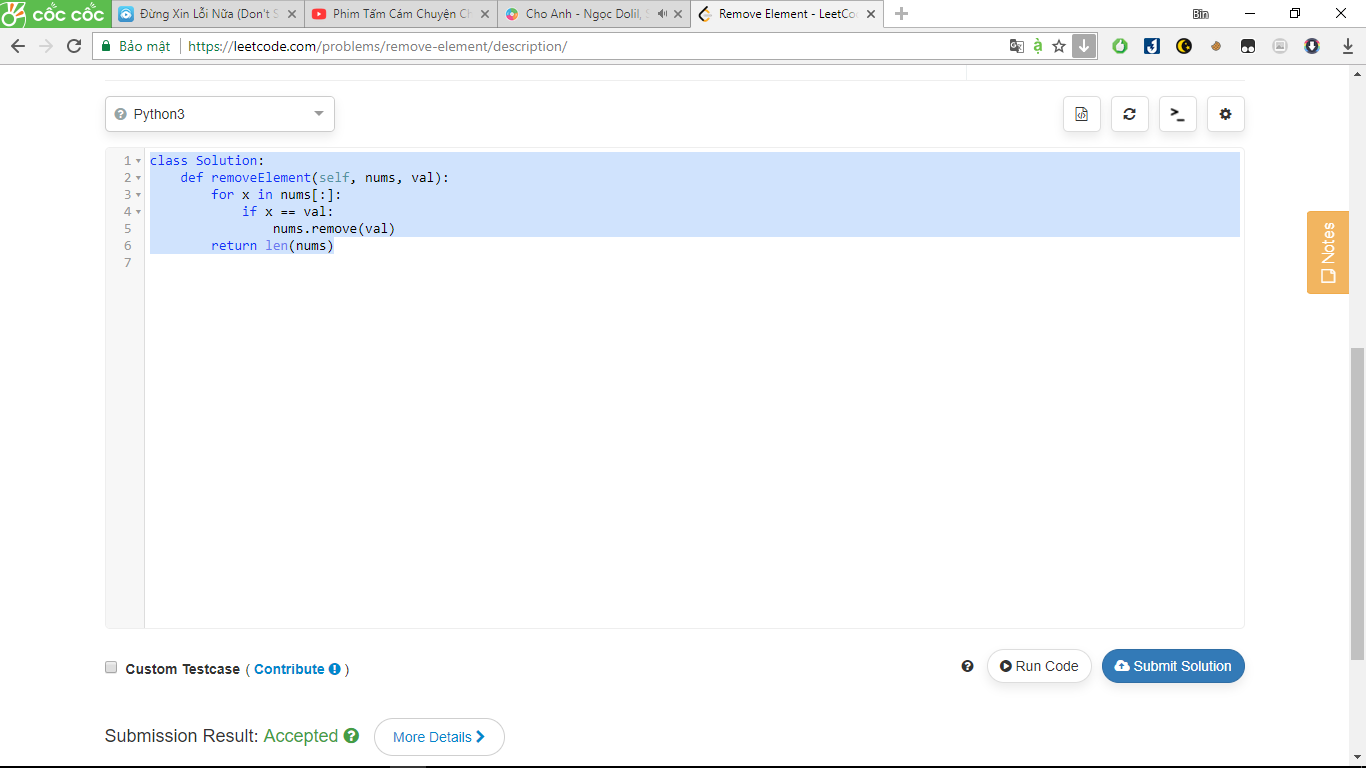
def removeElement(self, nums, val):

for x in nums[:]:

if x == val:

nums.remove(val)

return len(nums)



344. Reverse String

Write a function that takes a string as input and returns the string reversed.

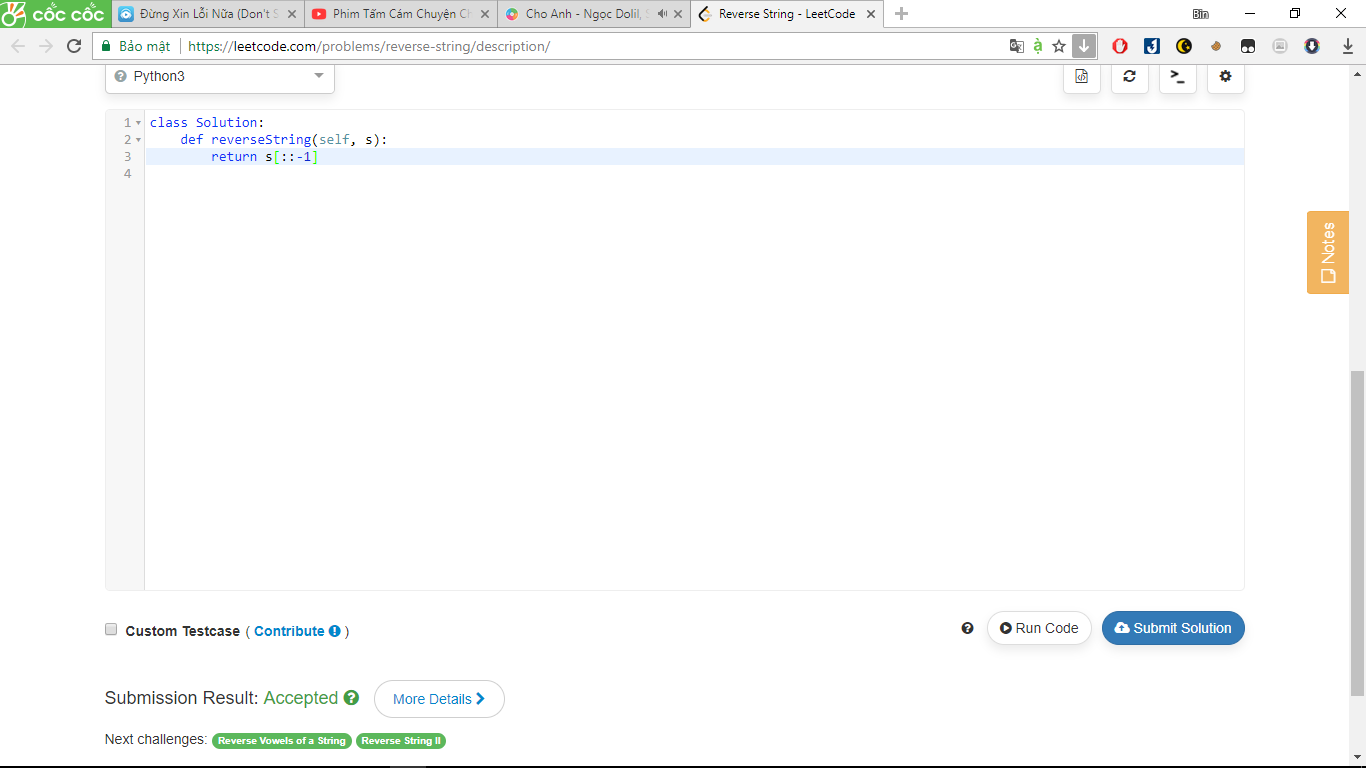
**Example:**  
Given s = "hello", return "olleh".

Answer:

Class solution:

Def reversestring(self,s):

Return s[::-1]



557. Reverse Words in a String III

Given a string, you need to reverse the order of characters in each word within a sentence while still preserving whitespace and initial word order.

**Example 1:**

**Input:** "Let's take LeetCode contest"

**Output:** "s'teL ekat edoCteeL tsetnoc"

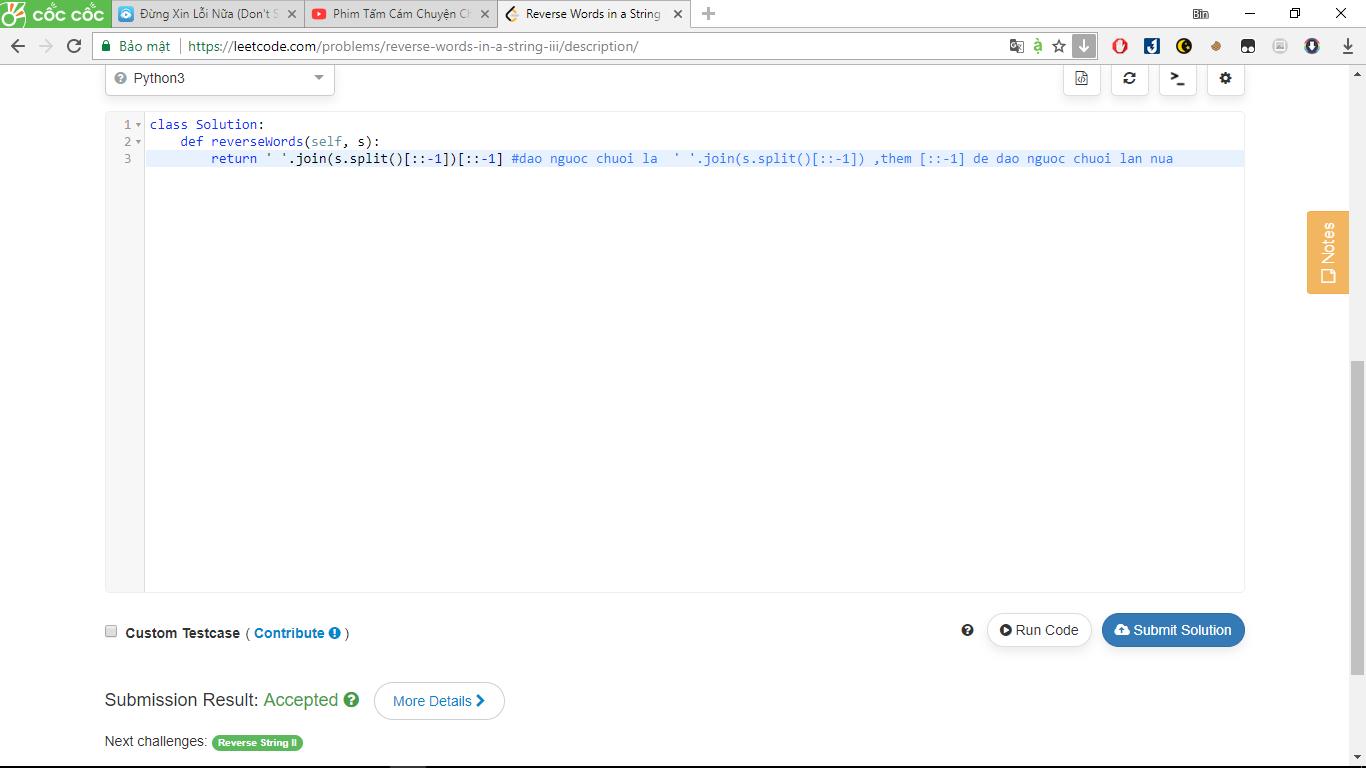
**Note:** In the string, each word is separated by single space and there will not be any extra space in the string.

Answer:

Class solution:

Def reverseWords(self,s):

Return ' '.join(a.split()[::-1])[::-1]



>>> a="hallo how aw"

>>> c=' '.join(a.split()[::-1])

>>> print(c)

aw how hallo

>>> print(c[::-1])

ollah woh wa

283. Move Zeroes

Given an array nums, write a function to move all 0's to the end of it while maintaining the relative order of the non-zero elements.

For example, given nums = [0, 1, 0, 3, 12], after calling your function, nums should be [1, 3, 12, 0, 0].

**Note**:

1. You must do this **in-place** without making a copy of the array.
2. Minimize the total number of operations.

Answer:

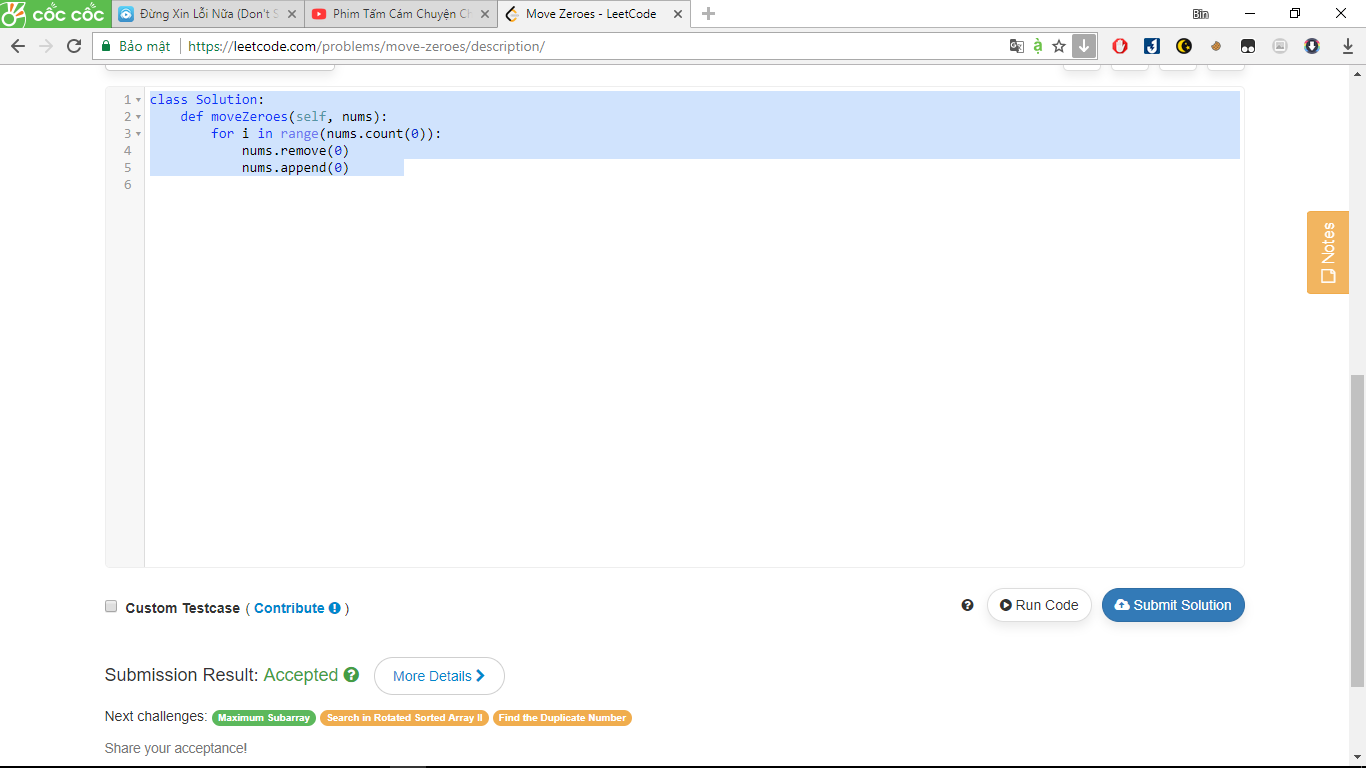
class Solution:

def moveZeroes(self, nums):

for i in range(nums.count(0)):

nums.remove(0)

nums.append(0)



389. Find the Difference

Given two strings ***s*** and ***t*** which consist of only lowercase letters.

String ***t*** is generated by random shuffling string ***s*** and then add one more letter at a random position.

Find the letter that was added in ***t***.

**Example:**

Input:

s = "abcd"

t = "abcde"

Output:

e

Explanation:

'e' is the letter that was added.

Answer:

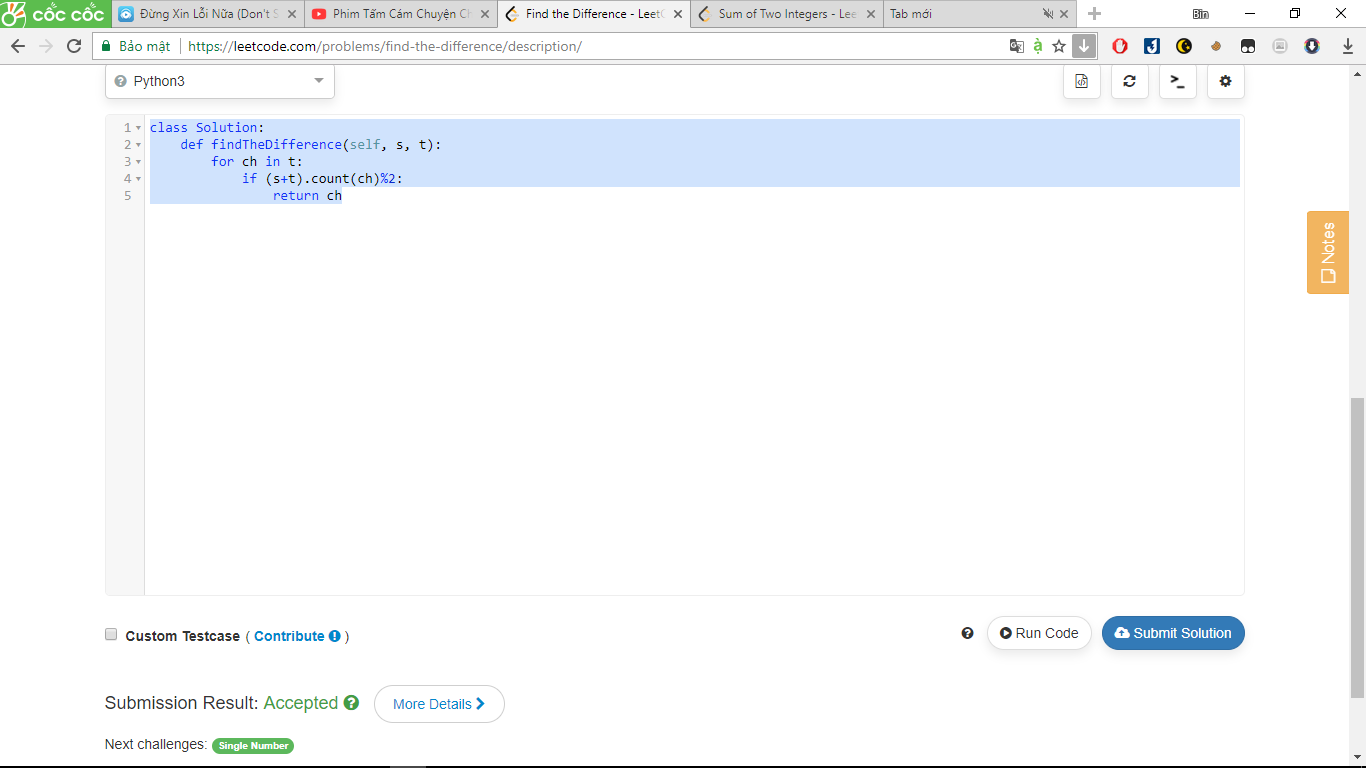
class Solution:

def findTheDifference(self, s, t):

for ch in t:

if (s+t).count(ch)%2:

return ch



371. Sum of Two Integers

Calculate the sum of two integers *a* and *b*, but you are **not allowed** to use the operator + and -.

**Example:**  
Given *a* = 1 and *b* = 2, return 3.

Answer:

class Solution:

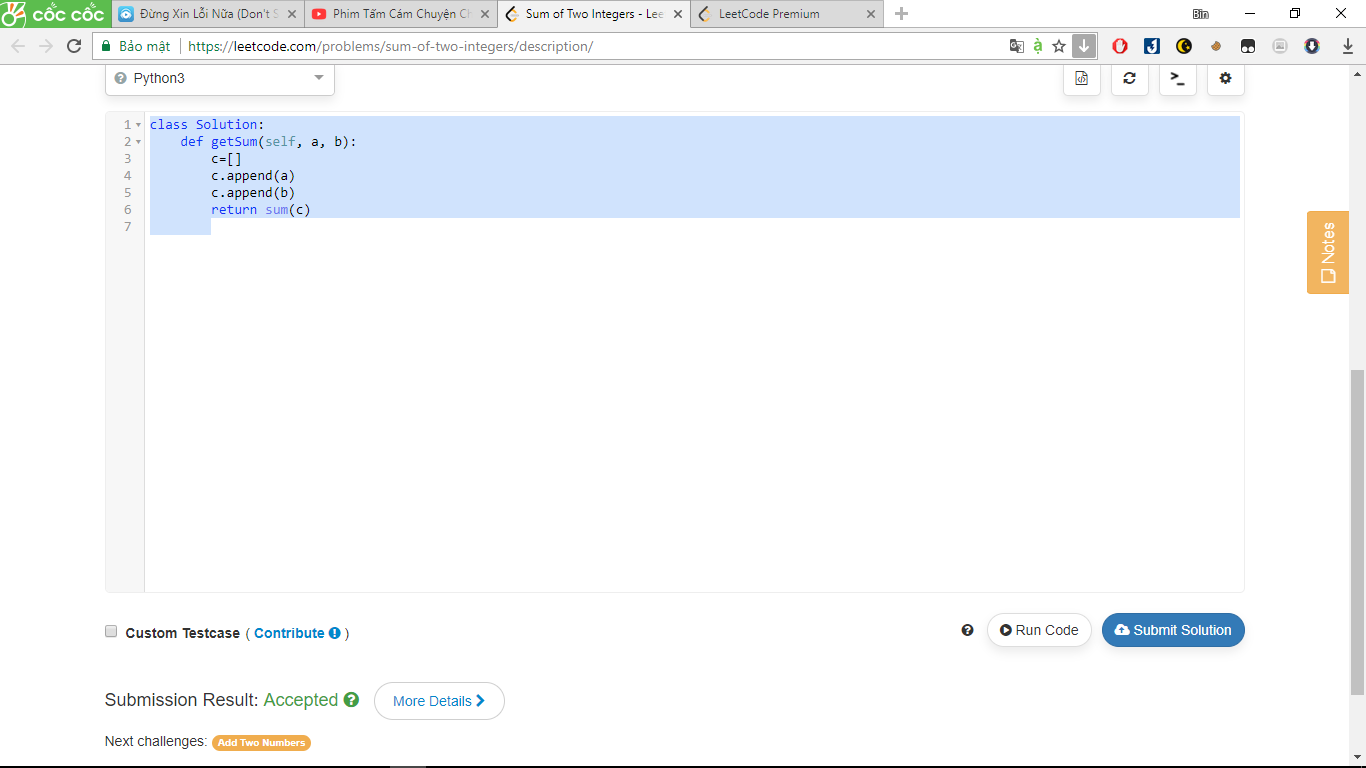
def getSum(self, a, b):

c=[]

c.append(a)

c.append(b)

return sum(c)



Fast way:

**class** **Solution**(object):

**def** **getSum**(self, a, b):

list=[a,b]

**return** sum(list)

771. Jewels and Stones

ou're given strings J representing the types of stones that are jewels, and S representing the stones you have.  Each character in Sis a type of stone you have.  You want to know how many of the stones you have are also jewels.

The letters in J are guaranteed distinct, and all characters in J and S are letters. Letters are case sensitive, so "a" is considered a different type of stone from "A".

**Example 1:**

**Input:** J = "aA", S = "aAAbbbb"

**Output:** 3

**Example 2:**

**Input:** J = "z", S = "ZZ"

**Output:** 0

**Note:**

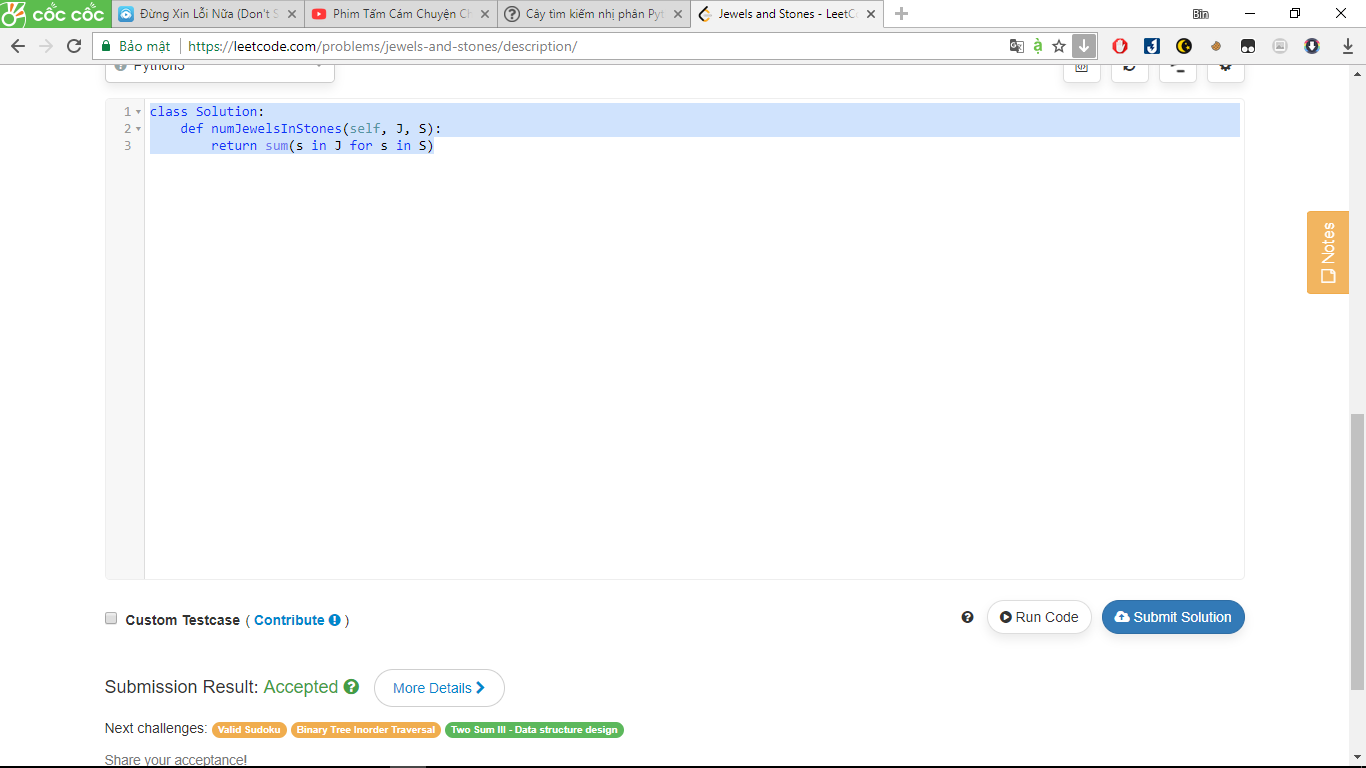
* S and J will consist of letters and have length at most 50.
* The characters in J are distinct.

Answer:

class Solution:

def numJewelsInStones(self, J, S):

return sum(s in J for s in S)



461. Hamming Distance

The [Hamming distance](https://en.wikipedia.org/wiki/Hamming_distance) between two integers is the number of positions at which the corresponding bits are different.

Given two integers x and y, calculate the Hamming distance.

**Note:**  
0 ≤ x, y < 231.

**Example:**

**Input:** x = 1, y = 4

**Output:** 2

**Explanation:**

1 (0 0 0 1)

4 (0 1 0 0)

↑ ↑

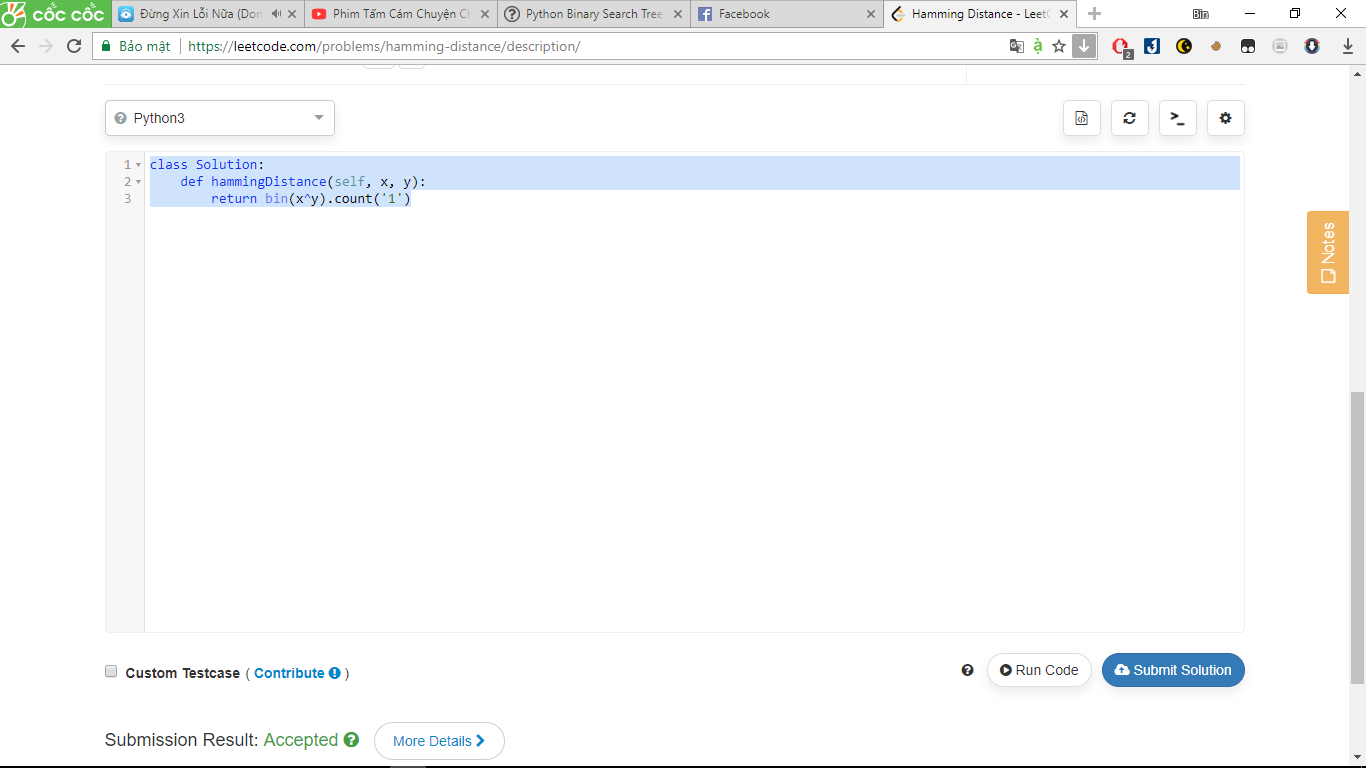
The above arrows point to positions where the corresponding bits are different.

Answer:

class Solution:

def hammingDistance(self, x, y):

return bin(x^y).count('1')



657. Judge Route Circle

Initially, there is a Robot at position (0, 0). Given a sequence of its moves, judge if this robot makes a circle, which means it moves back to **the original place**.

The move sequence is represented by a string. And each move is represent by a character. The valid robot moves are R (Right), L(Left), U (Up) and D (down). The output should be true or false representing whether the robot makes a circle.

**Example 1:**

**Input:** "UD"

**Output:** true

**Example 2:**

**Input:** "LL"

**Output:** false

Answer:

class Solution:

def judgeCircle(self, moves):

return (moves.count("U") == moves.count("D")) and (moves.count("R") == moves.count("L"))

\*chỉ cần: cân bằng giữa R và L , giữa U và D



561. Array Partition I

Given an array of **2n** integers, your task is to group these integers into **n** pairs of integer, say (a1, b1), (a2, b2), ..., (an, bn) which makes sum of min(ai, bi) for all i from 1 to n as large as possible.

**Example 1:**

**Input:** [1,4,3,2]

**Output:** 4

**Explanation:** n is 2, and the maximum sum of pairs is 4 = min(1, 2) + min(3, 4).

**Note:**

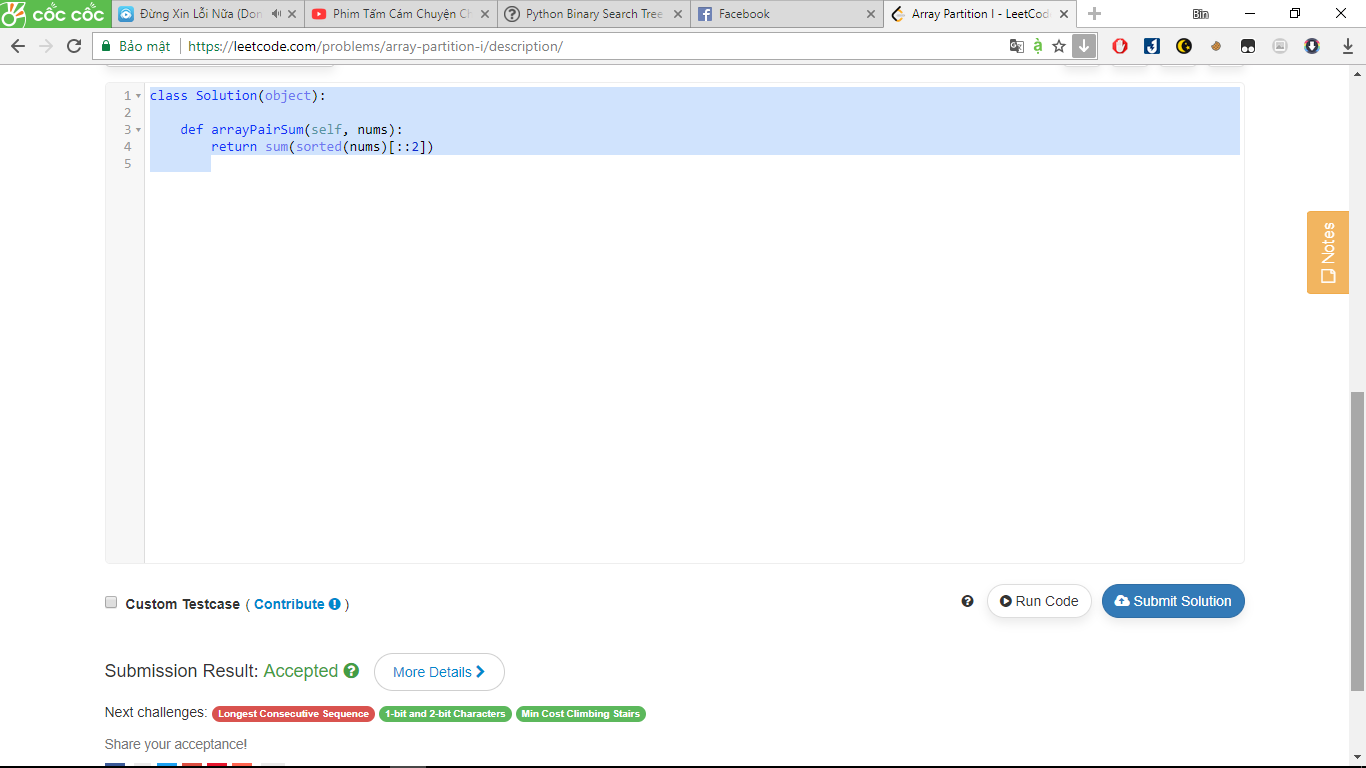
1. **n** is a positive integer, which is in the range of [1, 10000].
2. All the integers in the array will be in the range of [-10000, 10000].

Answer:

class Solution(object):

def arrayPairSum(self, nums):

return sum(sorted(nums)[::2])



191. Number of 1 Bits

Write a function that takes an unsigned integer and returns the number of ’1' bits it has (also known as the [Hamming weight](http://en.wikipedia.org/wiki/Hamming_weight)).

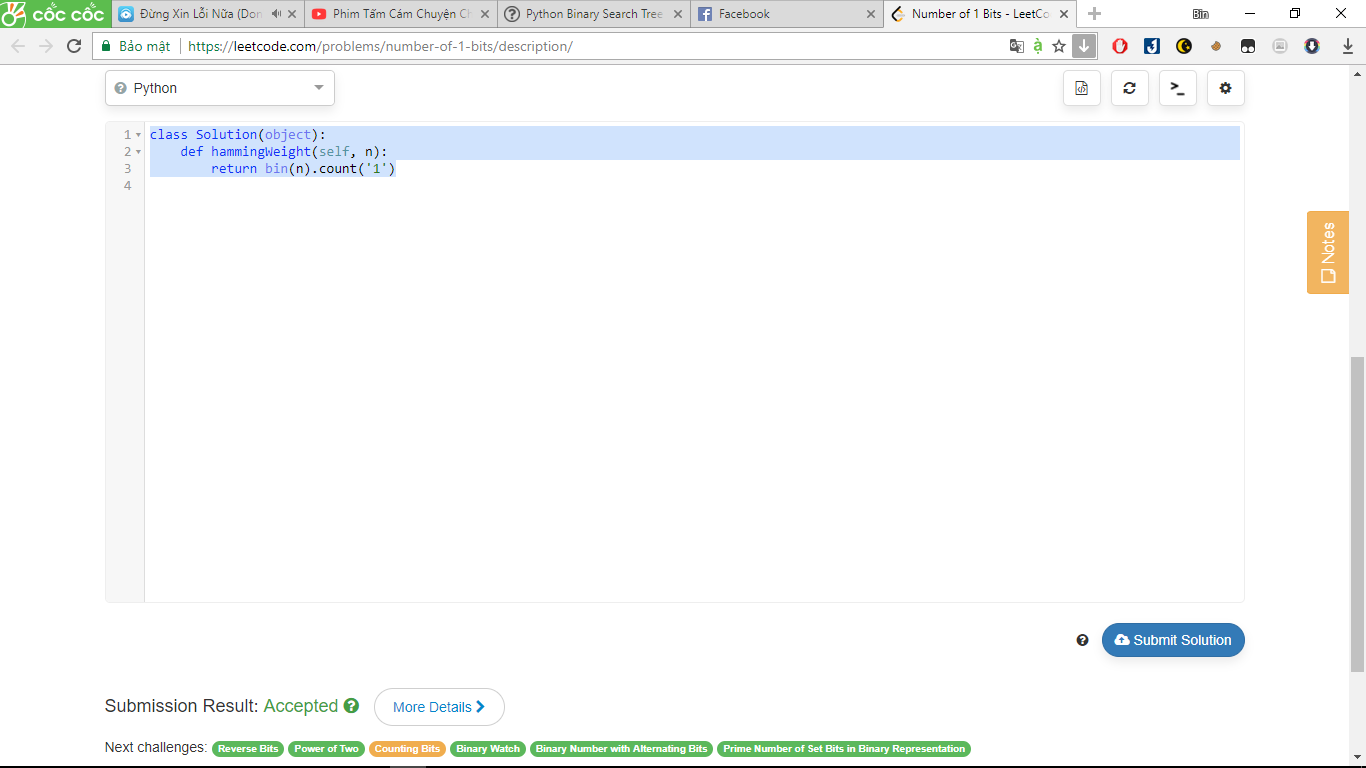
For example, the 32-bit integer ’11' has binary representation 00000000000000000000000000001011, so the function should return 3.

Answer:

class Solution(object):

def hammingWeight(self, n):

return bin(n).count('1')



35. Search Insert Position

Given a sorted array and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You may assume no duplicates in the array.

**Example 1:**

**Input:** [1,3,5,6], 5

**Output:** 2

**Example 2:**

**Input:** [1,3,5,6], 2

**Output:** 1

**Example 3:**

**Input:** [1,3,5,6], 7

**Output:** 4

**Example 1:**

**Input:** [1,3,5,6], 0

**Output:** 0

Answer:

class Solution(object):

def searchInsert(self, nums, target):

return len([x for x in nums if x<target])



724. Find Pivot Index

Given an array of integers nums, write a method that returns the "pivot" index of this array.

We define the pivot index as the index where the sum of the numbers to the left of the index is equal to the sum of the numbers to the right of the index.

If no such index exists, we should return -1. If there are multiple pivot indexes, you should return the left-most pivot index.

**Example 1:**

**Input:**

nums = [1, 7, 3, 6, 5, 6]

**Output:** 3

**Explanation:**

The sum of the numbers to the left of index 3 (nums[3] = 6) is equal to the sum of numbers to the right of index 3.

Also, 3 is the first index where this occurs.

**Example 2:**

**Input:**

nums = [1, 2, 3]

**Output:** -1

**Explanation:**

There is no index that satisfies the conditions in the problem statement.

**Note:**

 The length of nums will be in the range [0, 10000].

 Each element nums[i] will be an integer in the range [-1000, 1000].

Answer:

class Solution(object):

def pivotIndex(self, nums):

left, right = 0, sum(nums)

for index, num in enumerate(nums):

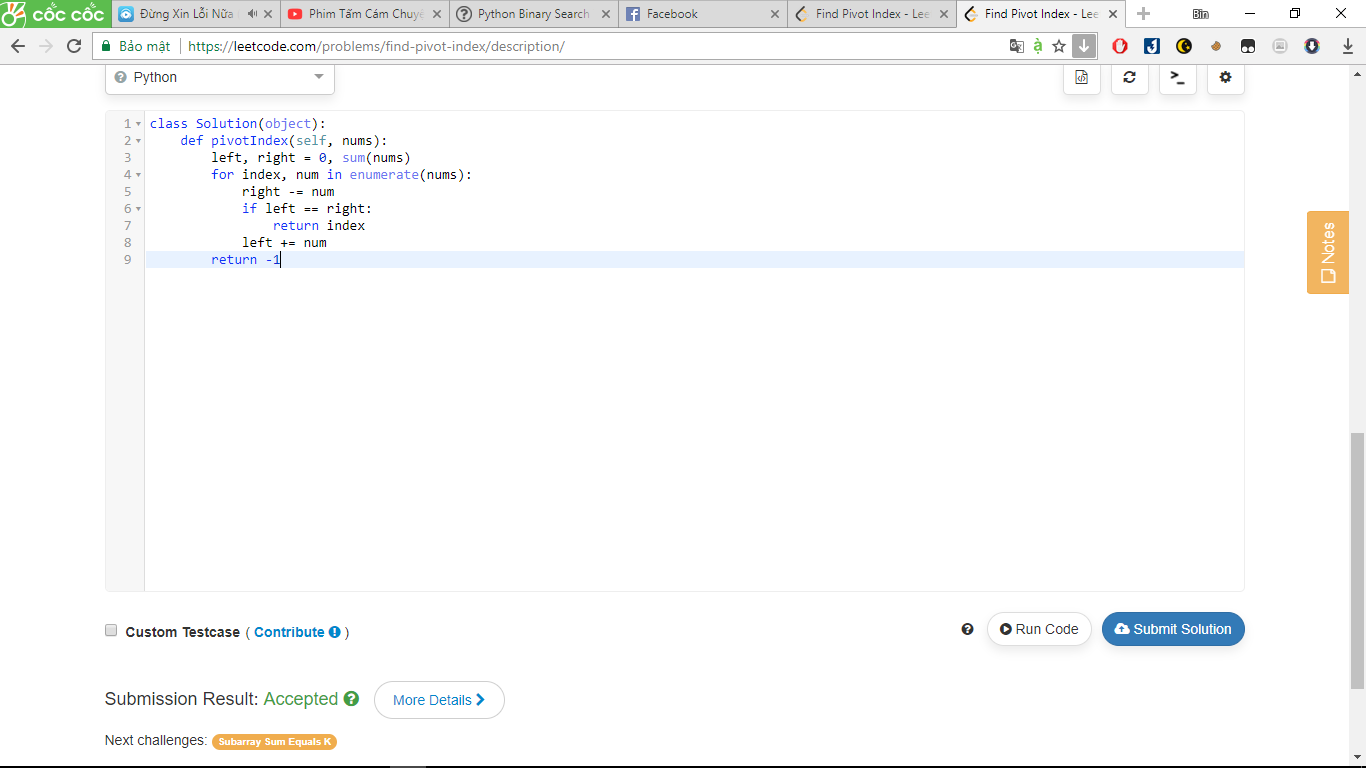
right -= num

if left == right:

return index

left += num

return -1



\*\* As we iterate through the array of numbers, we need to keep track of the sum of the values on the current number’s left and its right. The following debugger trace demonstrates the values of the variables in each loop before the left == right line

Input: [1, 7, 3, 6, 5, 6]

1. index: 0, num: 1, left: 0, right: 27
2. index: 1, num: 7, left: 1, right: 20
3. index: 2, num: 3, left: 8, right: 17
4. index: 3, num: 6, left: 11, right: 11 <-- Found!!!

198. House Robber

You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security system connected and **it will automatically contact the police if two adjacent houses were broken into on the same night**.

Given a list of non-negative integers representing the amount of money of each house, determine the maximum amount of money you can rob tonight **without alerting the police**.

Answer:

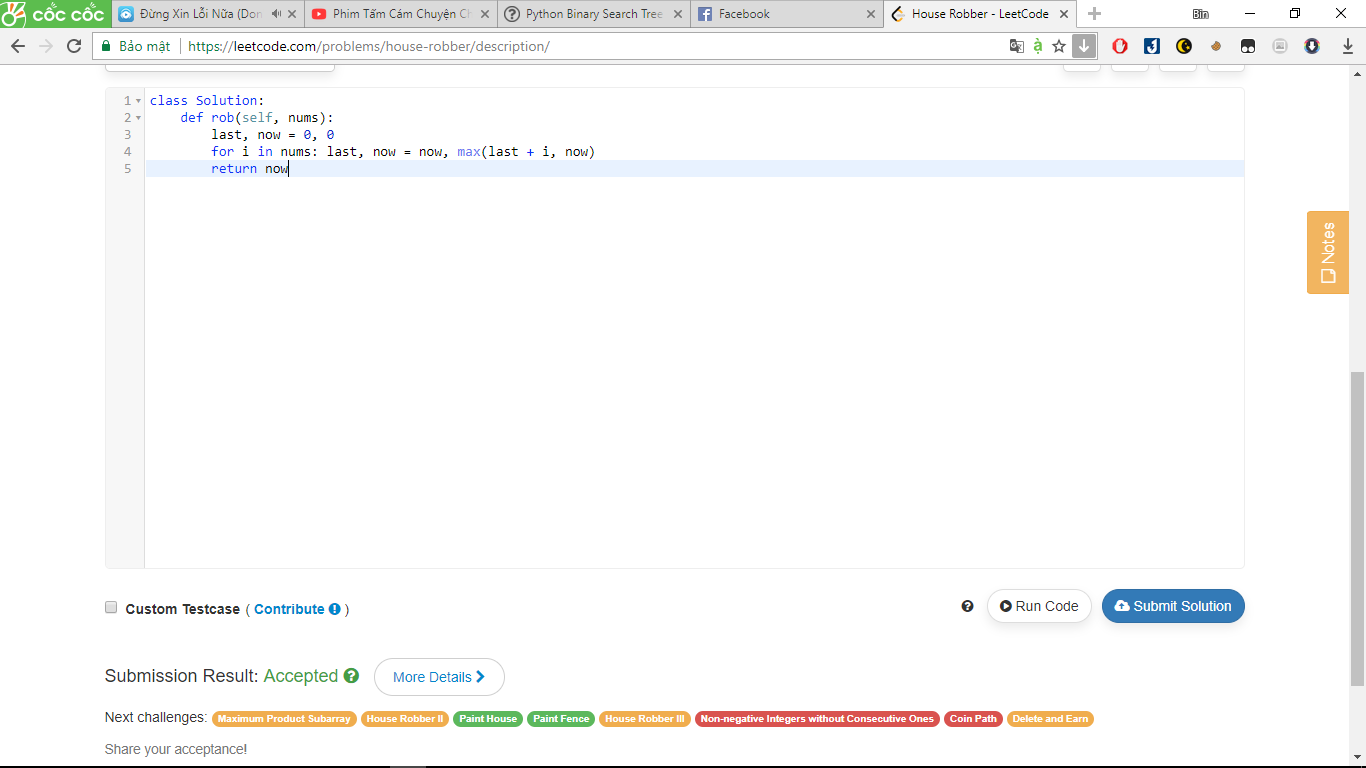
class Solution:

def rob(self, nums):

last, now = 0, 0

for i in nums: last, now = now, max(last + i, now)

return now



263. Ugly Number

Write a program to check whether a given number is an ugly number.

Ugly numbers are positive numbers whose prime factors only include 2, 3, 5. For example, 6, 8 are ugly while 14 is not ugly since it includes another prime factor 7.

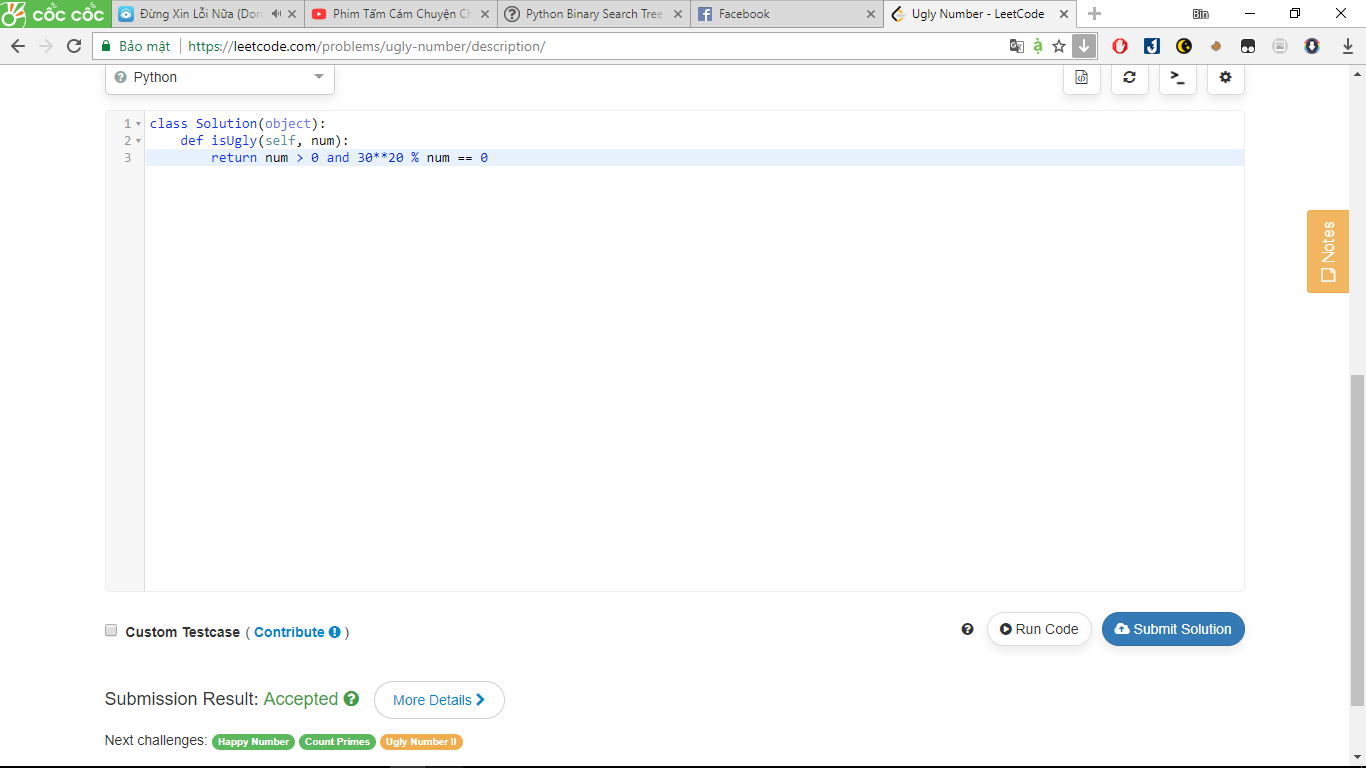
Note that 1 is typically treated as an ugly number.

Answer:

class Solution(object):

def isUgly(self, num):

return num > 0 and 30\*\*20 % num == 0



367. Valid Perfect Square

Given a positive integer *num*, write a function which returns True if *num* is a perfect square else False.

**Note:** **Do not** use any built-in library function such as sqrt.

**Example 1:**

Input: 16

Returns: True

**Example 2:**

Input: 14

Returns: False

Answer:

class Solution(object):

def isPerfectSquare(self, num):

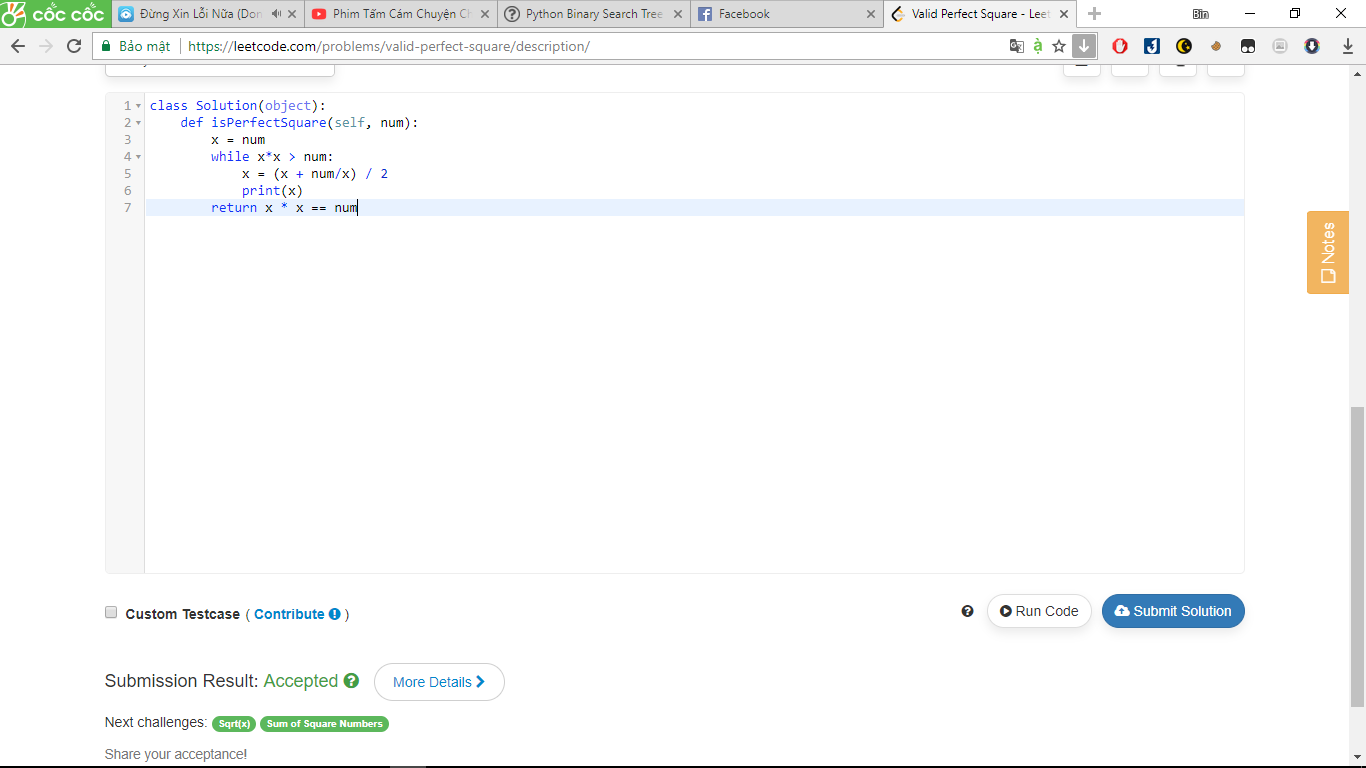
x = num

while x\*x > num:

x = (x + num/x) / 2

print(x)

return x \* x == num



459. Repeated Substring Pattern

Given a non-empty string check if it can be constructed by taking a substring of it and appending multiple copies of the substring together. You may assume the given string consists of lowercase English letters only and its length will not exceed 10000.

**Example 1:**

**Input:** "abab"

**Output:** True

**Explanation:** It's the substring "ab" twice.

**Example 2:**

**Input:** "aba"

**Output:** False

**Example 3:**

**Input:** "abcabcabcabc"

**Output:** True

**Explanation:** It's the substring "abc" four times. (And the substring "abcabc" twice.)

Answer:

class Solution(object):

def repeatedSubstringPattern(self, str):

return str in (2 \* str)[1:-1]



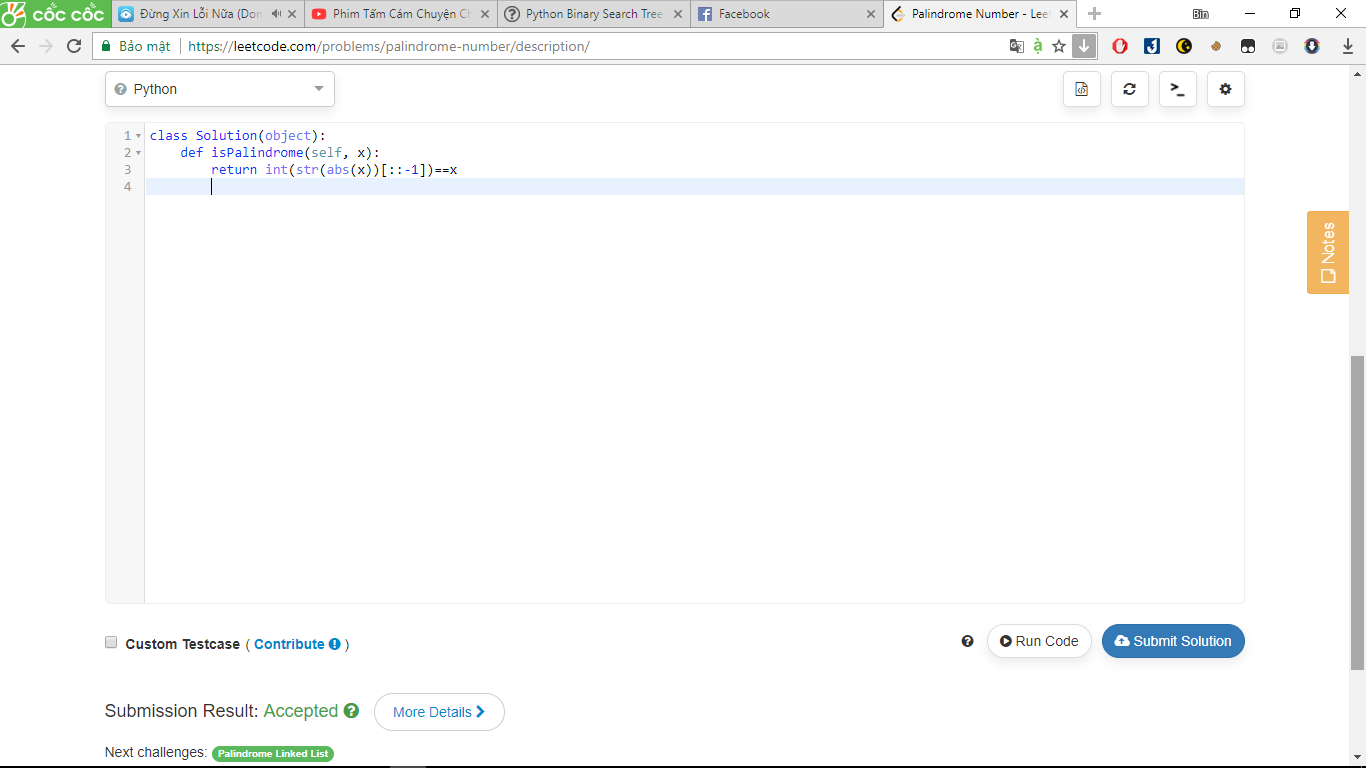
9. Palindrome Number

Answer:

class Solution(object):

def isPalindrome(self, x):

return int(str(abs(x))[::-1])==x



448. Find All Numbers Disappeared in an Array

Given an array of integers where 1 ≤ a[i] ≤ *n* (*n* = size of array), some elements appear twice and others appear once.

Find all the elements of [1, *n*] inclusive that do not appear in this array.

Could you do it without extra space and in O(*n*) runtime? You may assume the returned list does not count as extra space.

**Example:**

**Input:**

[4,3,2,7,8,2,3,1]

**Output:**

[5,6]

Answer:

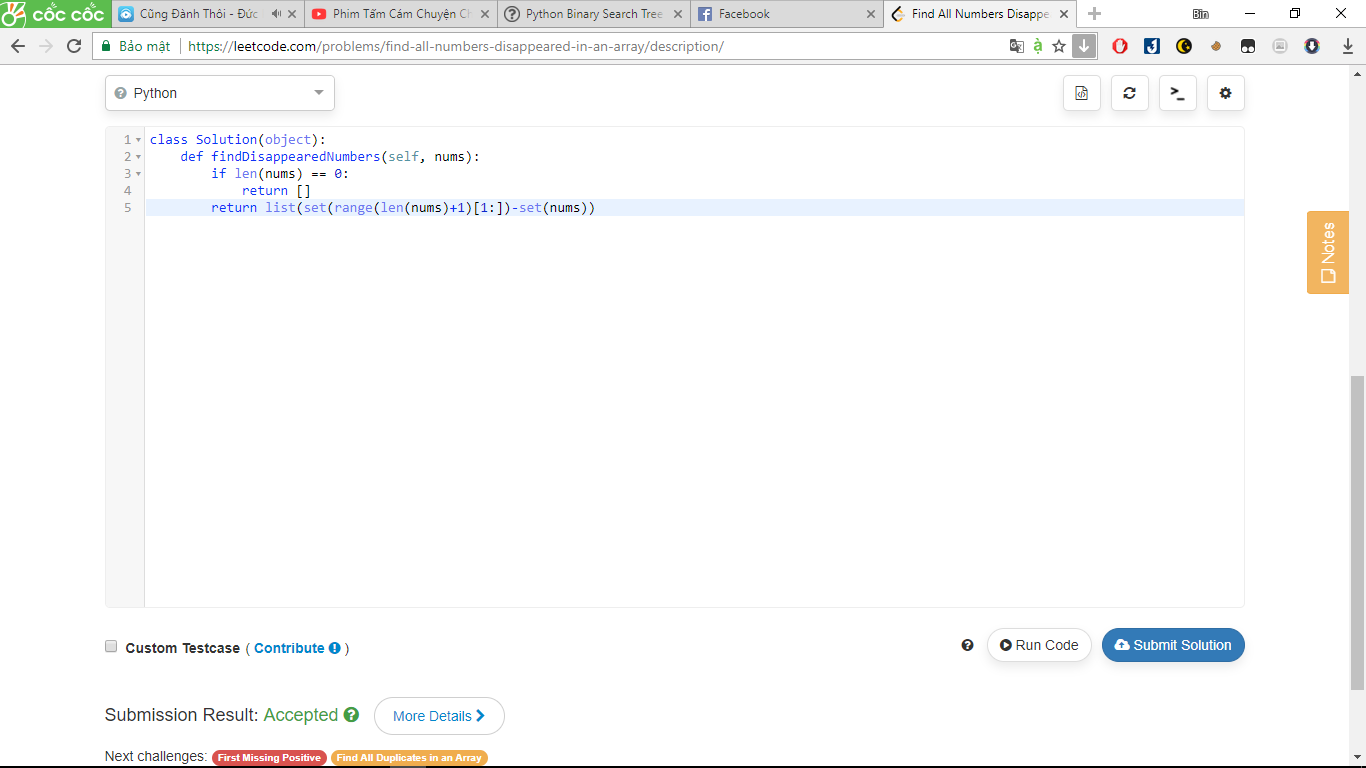
class Solution(object):

def findDisappearedNumbers(self, nums):

if len(nums) == 0:

return []

return list(set(range(len(nums)+1)[1:])-set(nums))



476. Number Complement

Given a positive integer, output its complement number. The complement strategy is to flip the bits of its binary representation.

**Note:**

1. The given integer is guaranteed to fit within the range of a 32-bit signed integer.
2. You could assume no leading zero bit in the integer’s binary representation.

**Example 1:**

**Input:** 5

**Output:** 2

**Explanation:** The binary representation of 5 is 101 (no leading zero bits), and its complement is 010. So you need to output 2.

**Example 2:**

**Input:** 1

**Output:** 0

**Explanation:** The binary representation of 1 is 1 (no leading zero bits), and its complement is 0. So you need to output 0.

Answer:

class Solution(object):

def findComplement(self, num):

"""

:type num: int

:rtype: int

"""

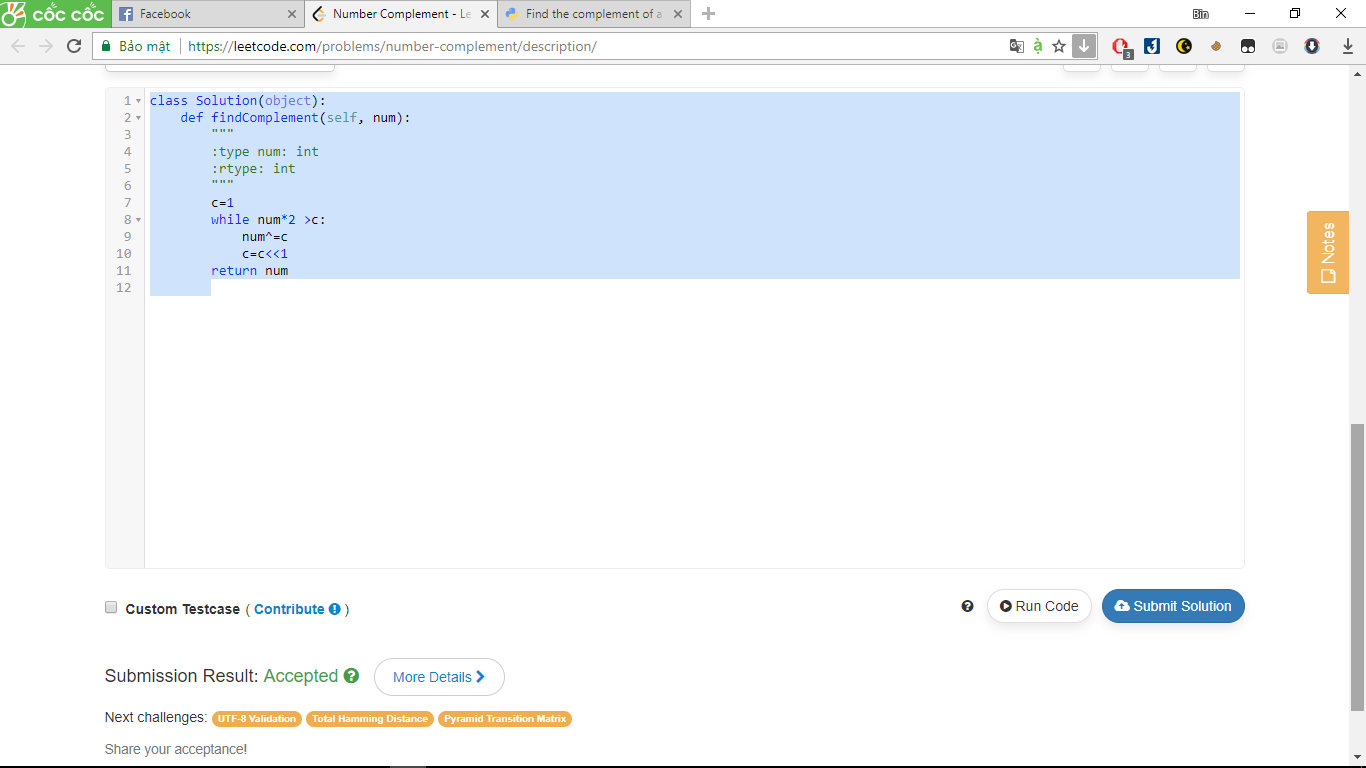
c=1

while num\*2 >c:

num^=c

c=c<<1

return num



412. Fizz Buzz

Write a program that outputs the string representation of numbers from 1 to *n*.

But for multiples of three it should output “Fizz” instead of the number and for the multiples of five output “Buzz”. For numbers which are multiples of both three and five output “FizzBuzz”.

**Example:**

n = 15,

Return:

[

"1",

"2",

"Fizz",

"4",

"Buzz",

"Fizz",

"7",

"8",

"Fizz",

"Buzz",

"11",

"Fizz",

"13",

"14",

"FizzBuzz"

]

Answer:

class Solution(object):

def fizzBuzz(self, n):

"""

:type n: int

:rtype: List[str]

"""

return [(not i%3)\*"Fizz" + (not i%5)\*"Buzz" or str(i) for i in range(1, n+1)]



575. Distribute Candies

Given an integer array with **even** length, where different numbers in this array represent different **kinds** of candies. Each number means one candy of the corresponding kind. You need to distribute these candies **equally** in number to brother and sister. Return the maximum number of **kinds** of candies the sister could gain.

**Example 1:**

**Input:** candies = [1,1,2,2,3,3]

**Output:** 3

**Explanation:**

There are three different kinds of candies (1, 2 and 3), and two candies for each kind.

Optimal distribution: The sister has candies [1,2,3] and the brother has candies [1,2,3], too.

The sister has three different kinds of candies.

**Example 2:**

**Input:** candies = [1,1,2,3]

**Output:** 2

**Explanation:** For example, the sister has candies [2,3] and the brother has candies [1,1].

The sister has two different kinds of candies, the brother has only one kind of candies.

**Note:**

1. The length of the given array is in range [2, 10,000], and will be even.
2. The number in given array is in range [-100,000, 100,000].

Answer:

class Solution(object):

def distributeCandies(self, candies):

"""

:type candies: List[int]

:rtype: int

"""

return min(len(candies) / 2, len(set(candies)))



Giải thích:

>>> candies=[1,1,2,2,3,3]

>>> c=min(len(candies) / 2, len(set(candies)))

>>> print(c)

3.0

>>> len(candies)

6

>>> set(candies)

{1, 2, 3}

\*\* người chị không thể lấy hơn nửa số nến

485. Max Consecutive Ones

Given a binary array, find the maximum number of consecutive 1s in this array.

**Example 1:**

**Input:** [1,1,0,1,1,1]

**Output:** 3

**Explanation:** The first two digits or the last three digits are consecutive 1s.

The maximum number of consecutive 1s is 3.

**Note:**

* The input array will only contain 0 and 1.
* The length of input array is a positive integer and will not exceed 10,000

Answer:

class Solution(object):

def findMaxConsecutiveOnes(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

count = 0

sclu = 0

for num in nums:

if num == 1:

count += 1

sclu = max(sclu, count)

else:

count = 0

return sclu



520. Detect Capital

Given a word, you need to judge whether the usage of capitals in it is right or not.

We define the usage of capitals in a word to be right when one of the following cases holds:

1. All letters in this word are capitals, like "USA".
2. All letters in this word are not capitals, like "leetcode".
3. Only the first letter in this word is capital if it has more than one letter, like "Google".

Otherwise, we define that this word doesn't use capitals in a right way.

**Example 1:**

**Input:** "USA"

**Output:** True

**Example 2:**

**Input:** "FlaG"

**Output:** False

Answer:

class Solution(object):

def detectCapitalUse(self, word):

"""

:type word: str

:rtype: bool

"""

return word[1:]==word[1:].lower() or word==word.upper()



258. Add Digits

Given a non-negative integer num, repeatedly add all its digits until the result has only one digit.

For example:

Given num = 38, the process is like: 3 + 8 = 11, 1 + 1 = 2. Since 2 has only one digit, return it.

**Follow up:**  
Could you do it without any loop/recursion in O(1) runtime?

Answer:

class Solution(object):

def addDigits(self, num):

"""

:type num: int

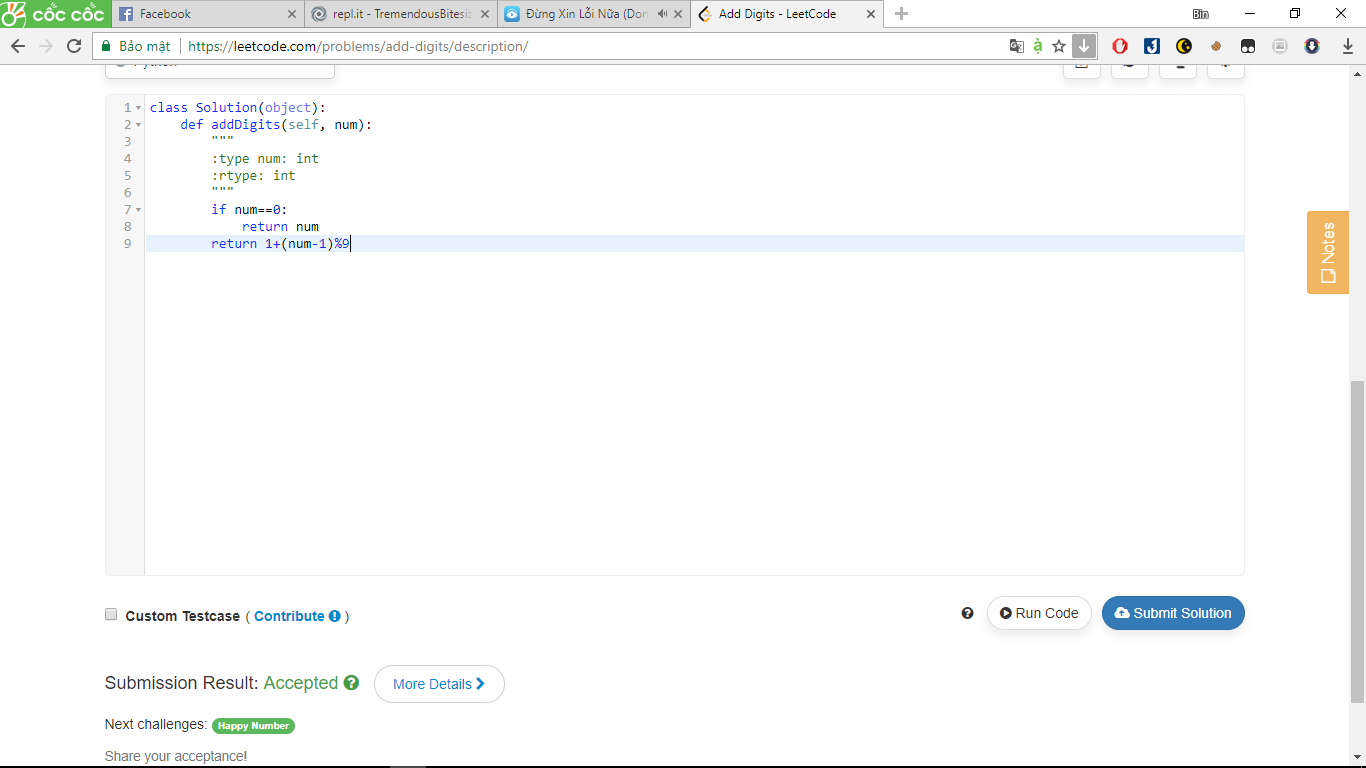
:rtype: int

"""

if num==0:

return num

return 1+(num-1)%9



492. Construct the Rectangle

For a web developer, it is very important to know how to design a web page's size. So, given a specific rectangular web page’s area, your job by now is to design a rectangular web page, whose length L and width W satisfy the following requirements:

1. The area of the rectangular web page you designed must equal to the given target area.

2. The width W should not be larger than the length L, which means L >= W.

3. The difference between length L and width W should be as small as possible.

You need to output the length L and the width W of the web page you designed in sequence.

**Example:**

**Input:** 4

**Output:** [2, 2]

**Explanation:** The target area is 4, and all the possible ways to construct it are [1,4], [2,2], [4,1].

But according to requirement 2, [1,4] is illegal; according to requirement 3, [4,1] is not optimal compared to [2,2]. So the length L is 2, and the width W is 2.

**Note:**

1. The given area won't exceed 10,000,000 and is a positive integer
2. The web page's width and length you designed must be positive integers.

Answer:

class Solution(object):

def constructRectangle(self, area):

"""

:type area: int

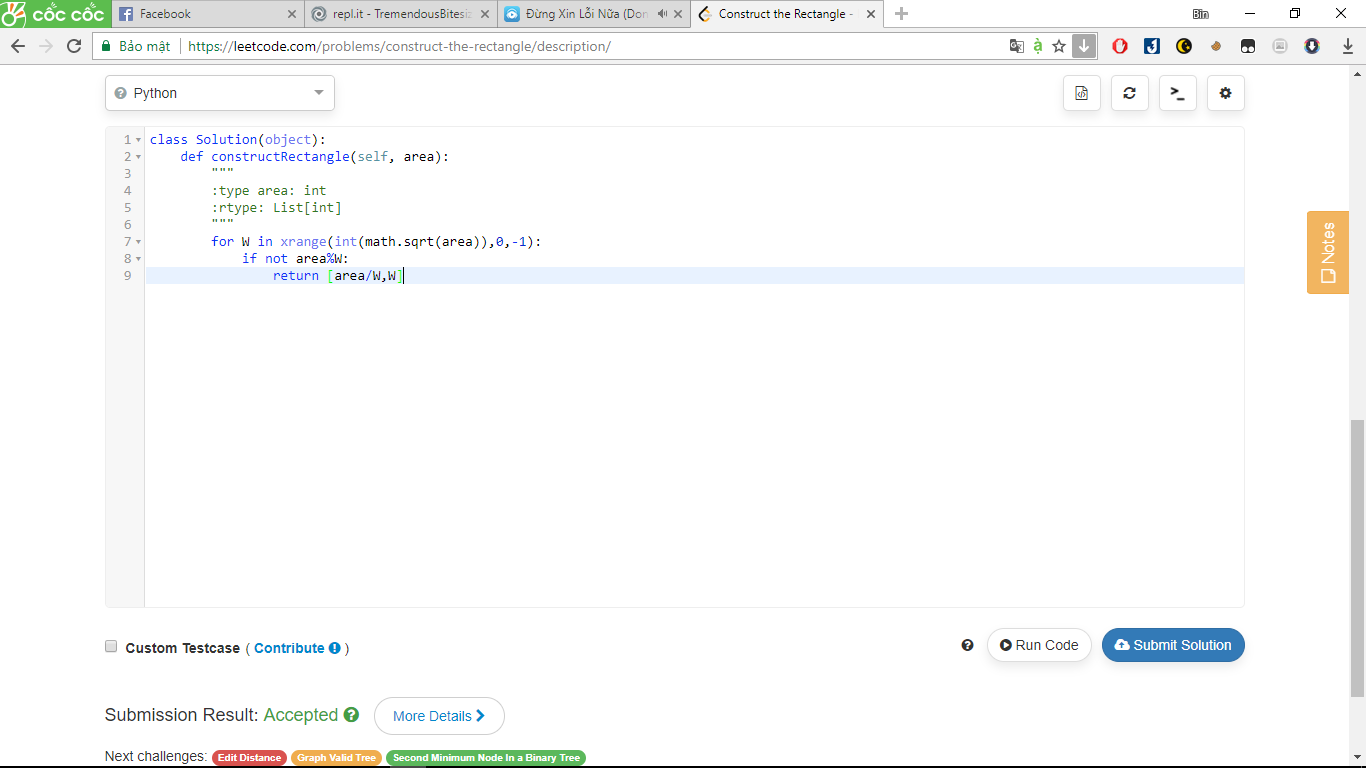
:rtype: List[int]

"""

for W in xrange(int(math.sqrt(area)),0,-1):

if not area%W:

return [area/W,W]



171. Excel Sheet Column Number

Given a column title as appear in an Excel sheet, return its corresponding column number.

For example:

A -> 1

B -> 2

C -> 3

...

Z -> 26

AA -> 27

AB -> 28

Answer:

class Solution(object):

def titleToNumber(self, s):

"""

:type s: str

:rtype: int

"""

res = 0

for i in s:

res = res\*26 + ord(i)-ord('A')+1

return res

# ham ord(i) de chuyen i ve ma ascii



453. Minimum Moves to Equal Array Elements

Given a **non-empty** integer array of size *n*, find the minimum number of moves required to make all array elements equal, where a move is incrementing *n* - 1 elements by 1.

**Example:**

**Input:**

[1,2,3]

**Output:**

3

**Explanation:**

Only three moves are needed (remember each move increments two elements):

[1,2,3] => [2,3,3] => [3,4,3] => [4,4,4]

Answer:

class Solution(object):

def minMoves(self, nums):

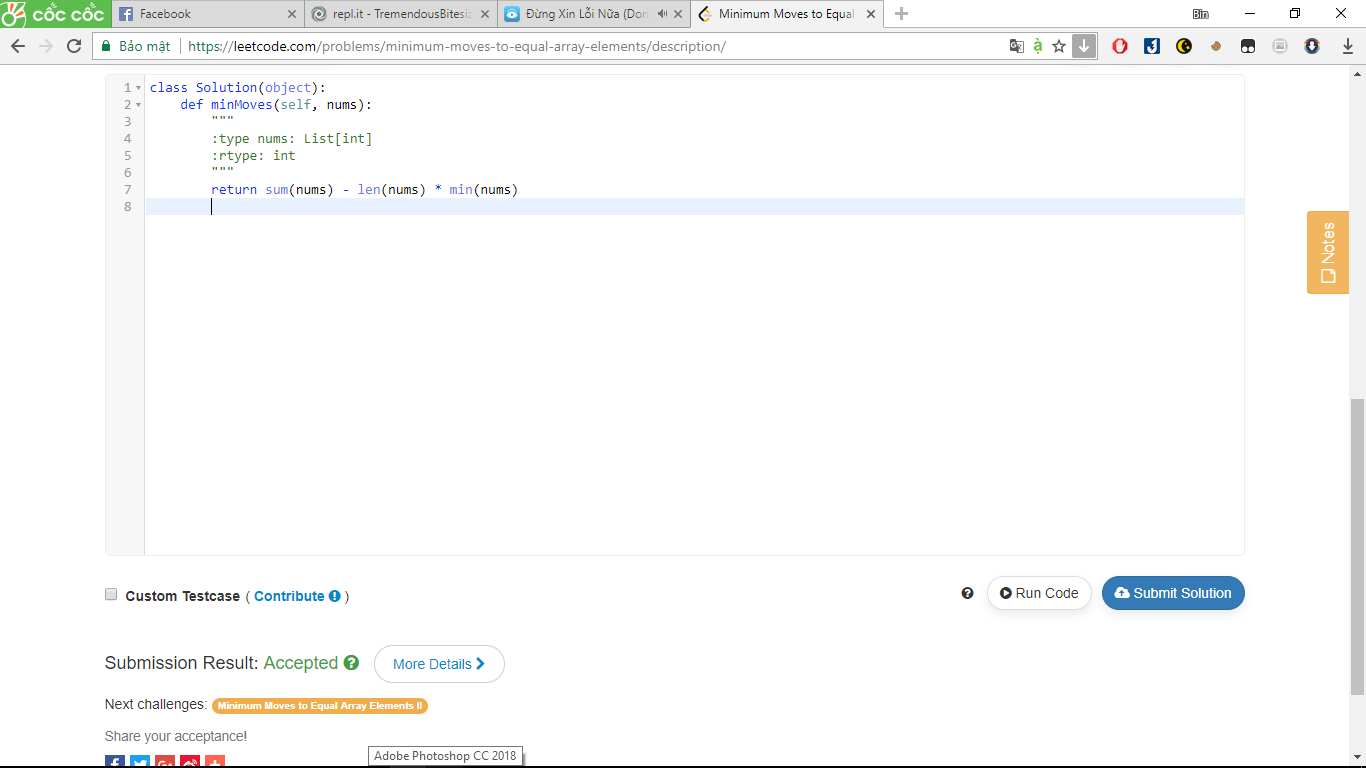
"""

:type nums: List[int]

:rtype: int

"""

return sum(nums) - len(nums) \* min(nums)



268. Missing Number

iven an array containing *n* distinct numbers taken from 0, 1, 2, ..., n, find the one that is missing from the array.

**Example 1**

**Input:** [3,0,1]

**Output:** 2

**Example 2**

**Input:** [9,6,4,2,3,5,7,0,1]

**Output:** 8

Answer:

class Solution:

def missingNumber(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

return int(len(nums) \* (len(nums) + 1) / 2 - sum(nums))



415. Add Strings

Given two non-negative integers num1 and num2 represented as string, return the sum of num1 and num2.

**Note:**

1. The length of both num1 and num2 is < 5100.
2. Both num1 and num2 contains only digits 0-9.
3. Both num1 and num2 does not contain any leading zero.
4. You **must not use any built-in BigInteger library** or **convert the inputs to integer** directly.

Answer:

class Solution:

def addStrings(self, num1, num2):

"""

:type num1: str

:type num2: str

:rtype: str

"""

return str(eval(num1) + eval(num2))



\*\* hàm eval() chuyển số ở dạng string sang số

599. Minimum Index Sum of Two Lists

uppose Andy and Doris want to choose a restaurant for dinner, and they both have a list of favorite restaurants represented by strings.

You need to help them find out their **common interest** with the **least list index sum**. If there is a choice tie between answers, output all of them with no order requirement. You could assume there always exists an answer.

**Example 1:**

**Input:**

["Shogun", "Tapioca Express", "Burger King", "KFC"]

["Piatti", "The Grill at Torrey Pines", "Hungry Hunter Steakhouse", "Shogun"]

**Output:** ["Shogun"]

**Explanation:** The only restaurant they both like is "Shogun".

**Example 2:**

**Input:**

["Shogun", "Tapioca Express", "Burger King", "KFC"]

["KFC", "Shogun", "Burger King"]

**Output:** ["Shogun"]

**Explanation:** The restaurant they both like and have the least index sum is "Shogun" with index sum 1 (0+1).

**Note:**

1. The length of both lists will be in the range of [1, 1000].
2. The length of strings in both lists will be in the range of [1, 30].
3. The index is starting from 0 to the list length minus 1.
4. No duplicates in both lists.

Answer:

class Solution:

def findRestaurant(self, list1, list2):

"""

:type list1: List[str]

:type list2: List[str]

:rtype: List[str]

"""

d = {}

result = []

minimum = (len(list1) + len(list2) - 2)

for i in range(len(list1)):

d[list1[i]] = i

for j in range(len(list2)):

if list2[j] in d:

index\_sum = j+d[list2[j]]

if index\_sum == minimum:

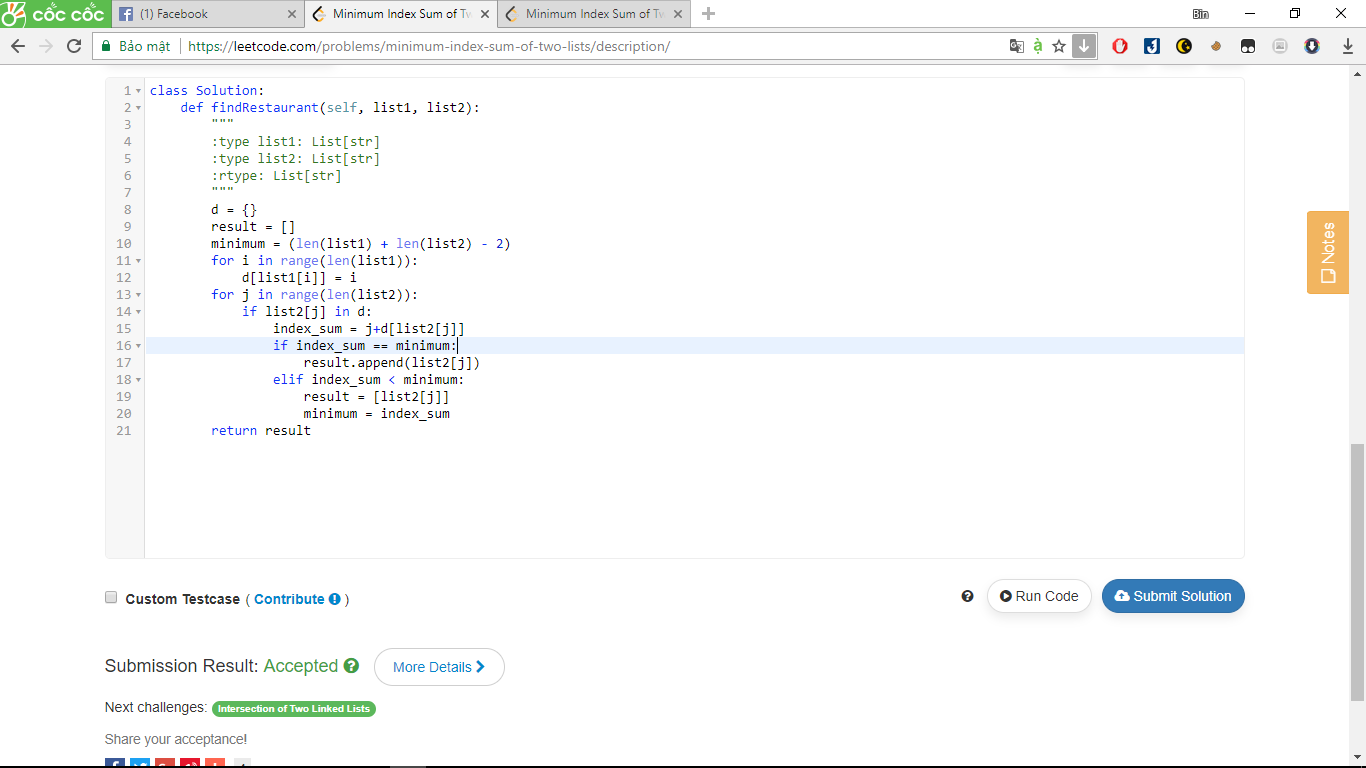
result.append(list2[j])

elif index\_sum < minimum:

result = [list2[j]]

minimum = index\_sum

return result



206. Reverse Linked List

Answer:

# Definition for singly-linked list.

# class ListNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.next = None

class Solution:

def reverseList(self, head):

"""

:type head: ListNode

:rtype: ListNode

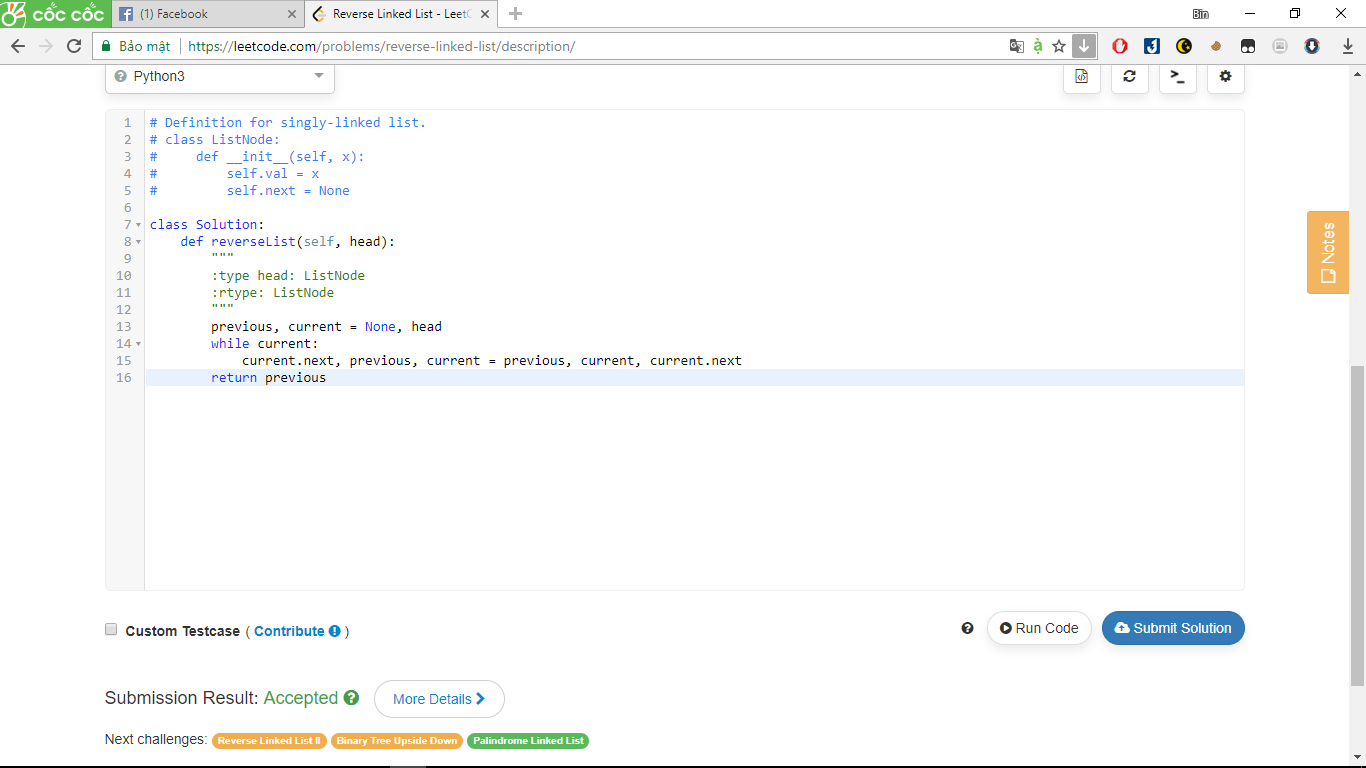
"""

previous, current = None, head

while current:

current.next, previous, current = previous, current, current.next

return previous



217. Contains Duplicate

Given an array of integers, find if the array contains any duplicates. Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.

Answer:

class Solution:

def containsDuplicate(self, nums):

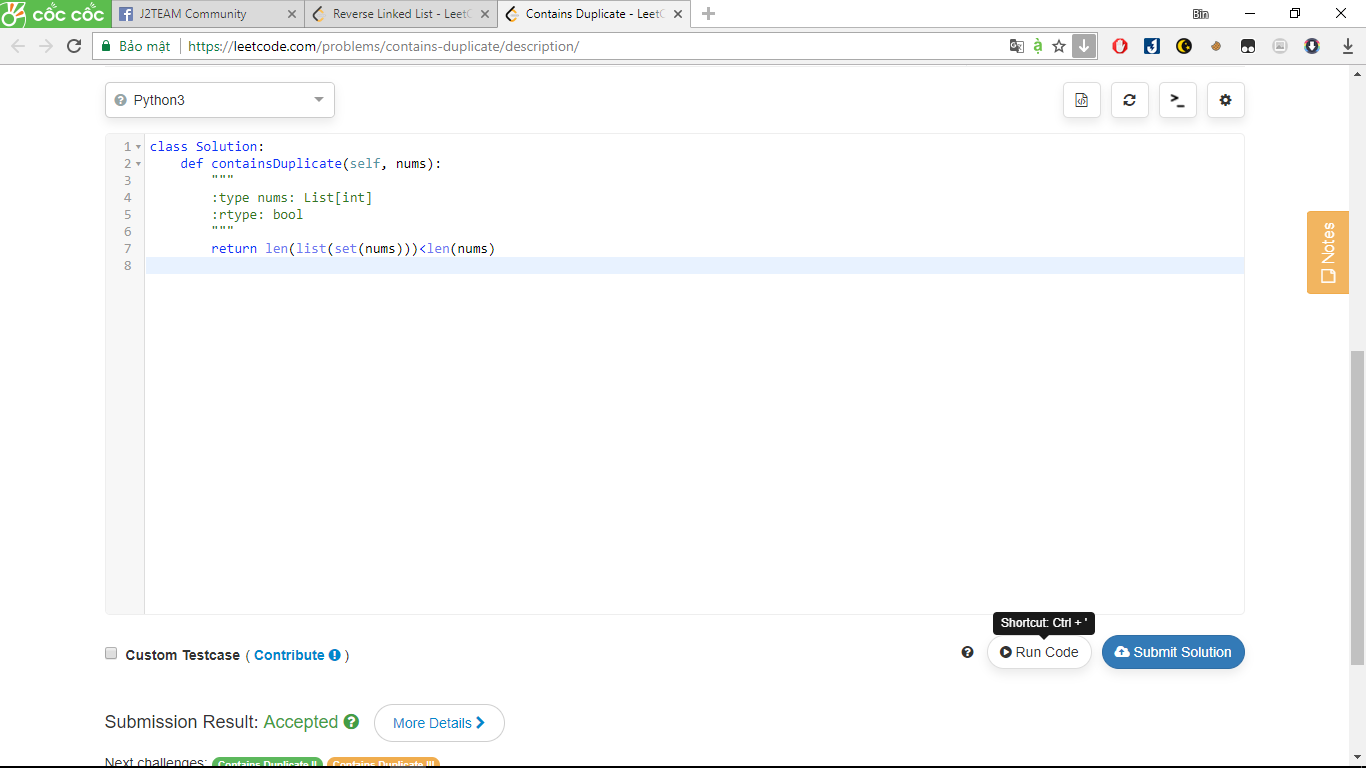
"""

:type nums: List[int]

:rtype: bool

"""

return len(list(set(nums)))<len(nums)



237. Delete Node in a Linked List

Write a function to delete a node (except the tail) in a singly linked list, given only access to that node.

Supposed the linked list is 1 -> 2 -> 3 -> 4 and you are given the third node with value 3, the linked list should become 1 -> 2 -> 4 after calling your function.

Answer:

# Definition for singly-linked list.

# class ListNode(object):

# def \_\_init\_\_(self, x):

# self.val = x

# self.next = None

class Solution(object):

def deleteNode(self, node):

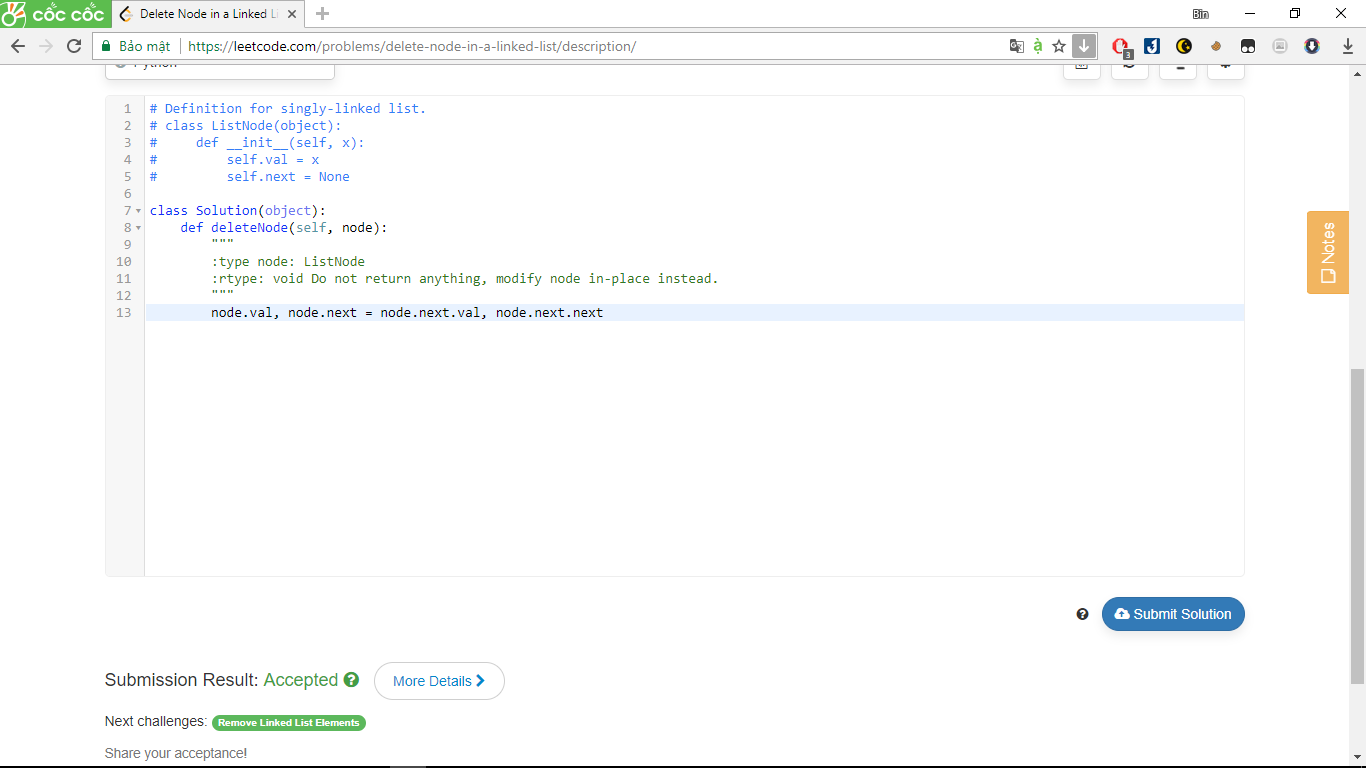
"""

:type node: ListNode

:rtype: void Do not return anything, modify node in-place instead.

"""

node.val, node.next = node.next.val, node.next.next



242. Valid Anagram

iven two strings *s* and *t*, write a function to determine if *t* is an anagram of *s*.

For example,  
*s* = "anagram", *t* = "nagaram", return true.  
*s* = "rat", *t* = "car", return false.

**Note:**  
You may assume the string contains only lowercase alphabets.

**Follow up:**  
What if the inputs contain unicode characters? How would you adapt your solution to such case?

Answer:

class Solution(object):

def isAnagram(self, s, t):

"""

:type s: str

:type t: str

:rtype: bool

"""

maps = {}

mapt = {}

for c in s:

maps[c] = maps.get(c,0)+1

for c in t:

mapt[c] = mapt.get(c,0)+1

return maps == mapt



342. Power of Four

Given an integer (signed 32 bits), write a function to check whether it is a power of 4.

**Example:**  
Given num = 16, return true. Given num = 5, return false.

**Follow up**: Could you solve it without loops/recursion?

**Credits:**

Answer:

class Solution:

def isPowerOfFour(self, num):

"""

:type num: int

:rtype: bool

"""

return num != 0 and num &(num-1) == 0 and num & 1431655765== num

231. Power of Two

Given an integer, write a function to determine if it is a power of two.

Answer:

class Solution:

def isPowerOfTwo(self, n):

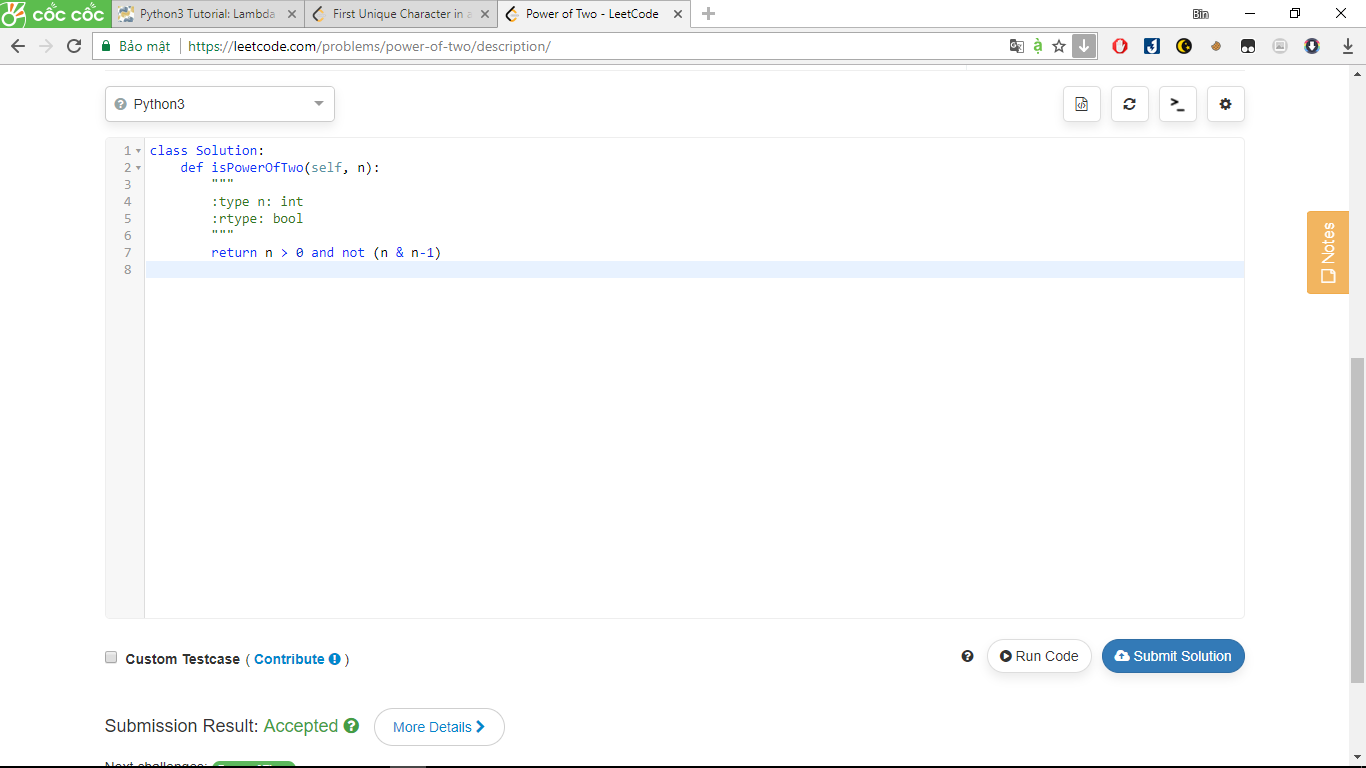
"""

:type n: int

:rtype: bool

"""

return n > 0 and not (n & n-1)



\*\* n&(n-1) xoa bit đầu tiên trái của bit n . nếu n là power of 2 thì trong dãy nhị phân chỉ có 1 bit một ở đầu tiền. vd: 16=0b10000 ,16-1=0b01111 , và 16&(16-1)=0b10000&0b01111 =0 và 16!=0 ,nên 16 là power of 2 .

\*\* not (một số khác không) -> false

\*\*not 0=True

383. Ransom Note

Given an arbitrary ransom note string and another string containing letters from all the magazines, write a function that will return true if the ransom note can be constructed from the magazines ; otherwise, it will return false.

Each letter in the magazine string can only be used once in your ransom note.

**Note:**  
You may assume that both strings contain only lowercase letters.

canConstruct("a", "b") -> false

canConstruct("aa", "ab") -> false

canConstruct("aa", "aab") -> true

answer:

class Solution:

def canConstruct(self, ransomNote, magazine):

"""

:type ransomNote: str

:type magazine: str

:rtype: bool

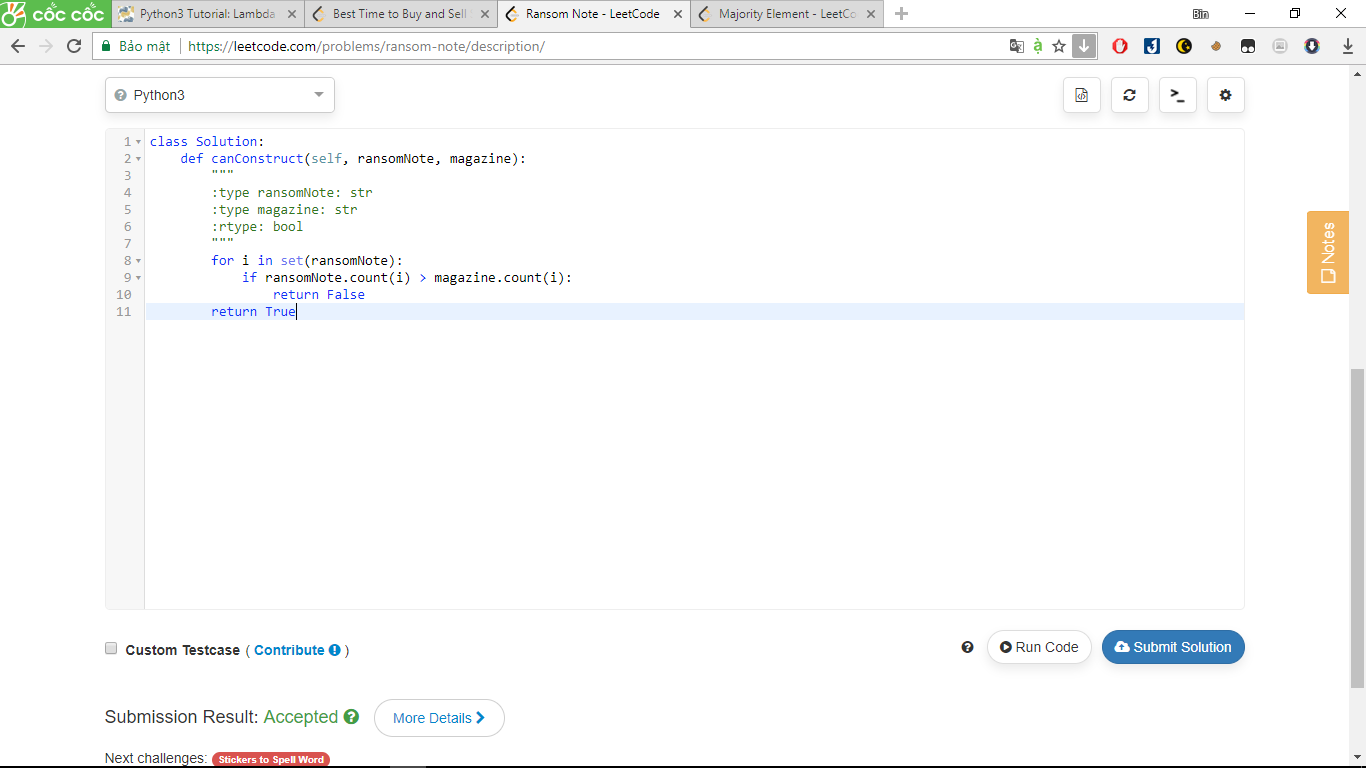
"""

for i in set(ransomNote):

if ransomNote.count(i) > magazine.count(i):

return False

return True



434. Number of Segments in a String

Count the number of segments in a string, where a segment is defined to be a contiguous sequence of non-space characters.

Please note that the string does not contain any **non-printable** characters.

**Example:**

**Input:** "Hello, my name is John"

**Output:** 5

Answer:

class Solution:

def countSegments(self, s):

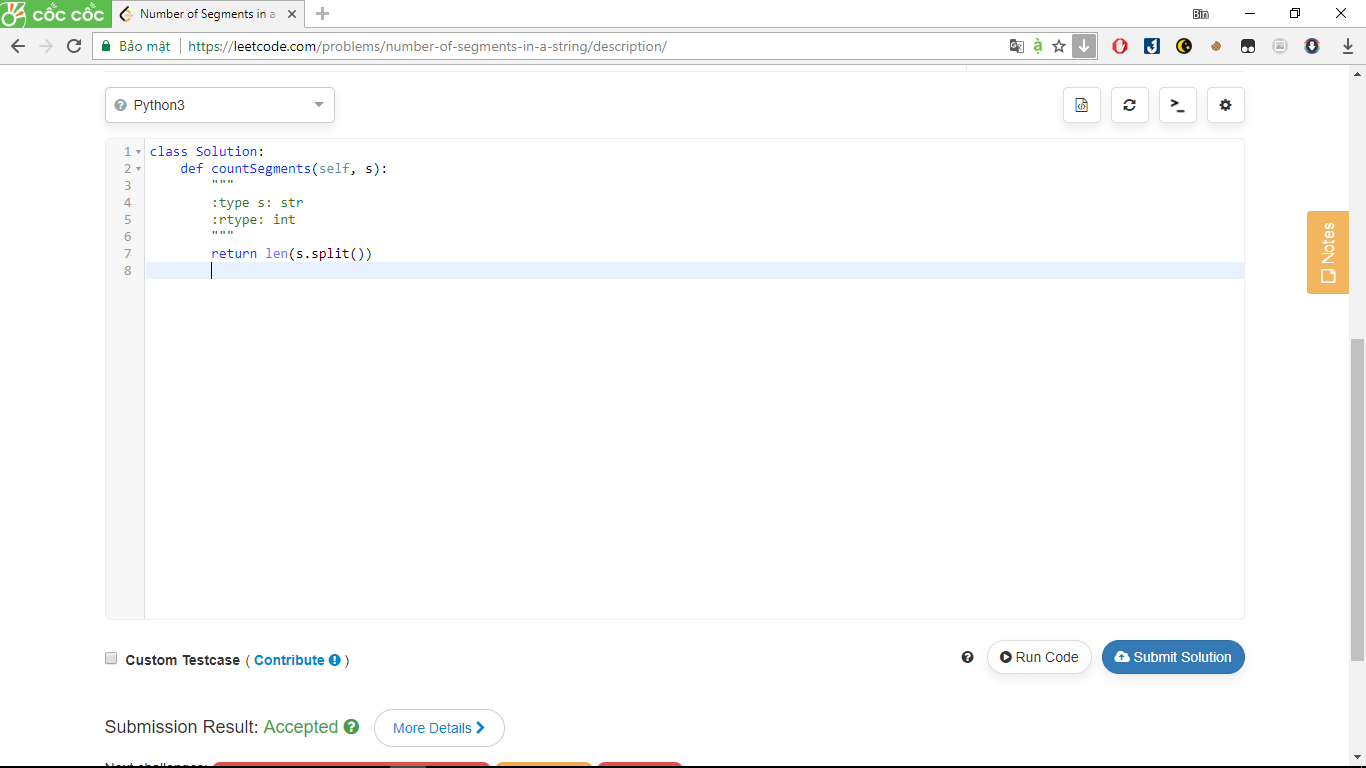
"""

:type s: str

:rtype: int

"""

return len(s.split())



141. Linked List Cycle

Given a linked list, determine if it has a cycle in it.

Follow up:  
Can you solve it without using extra space?

Answer:

# Definition for singly-linked list.

# class ListNode(object):

# def \_\_init\_\_(self, x):

# self.val = x

# self.next = None

class Solution(object):

def hasCycle(self, head):

"""

:type head: ListNode

:rtype: bool

"""

fast=head

slow=head

while fast is not None and fast.next is not None:

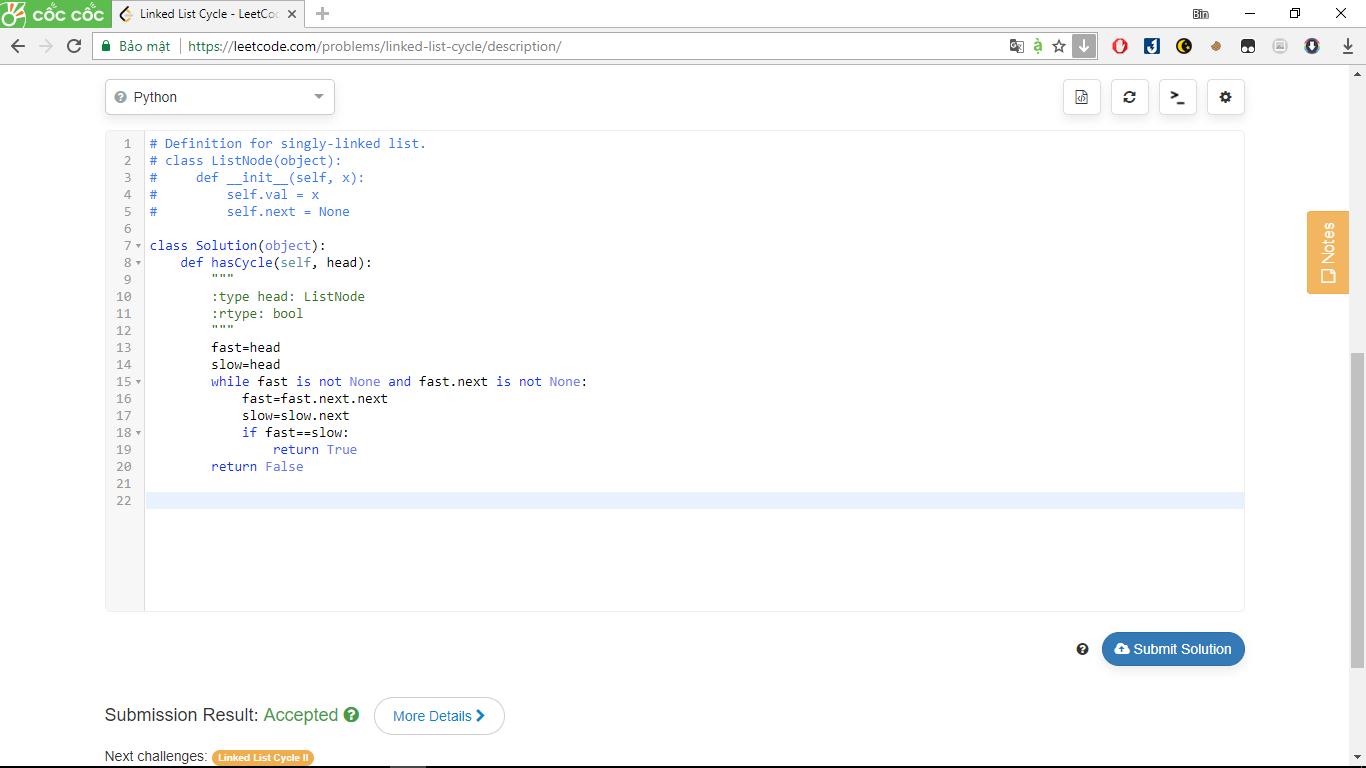
fast=fast.next.next

slow=slow.next

if fast==slow:

return True

return False



326. Power of Three

import math

class Solution(object):

def isPowerOfThree(self, n):

"""

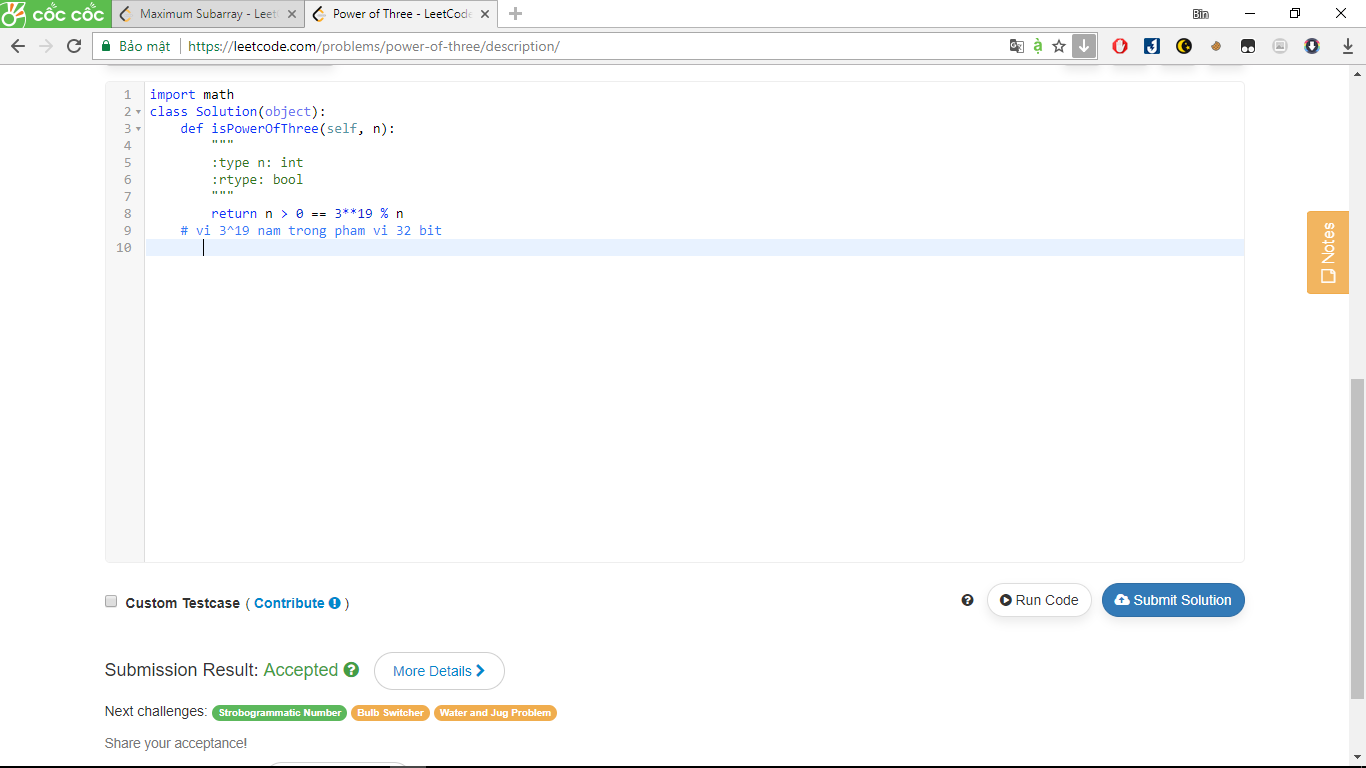
:type n: int

:rtype: bool

"""

return n > 0 == 3\*\*19 % n

# vi 3^19 nam trong pham vi 32 bit



136. Single Number

iven an array of integers, every element appears *twice* except for one. Find that single one.

**Note:**  
Your algorithm should have a linear runtime complexity. Could you implement it without using extra memory?

Answer:

class Solution(object):

def singleNumber(self, nums):

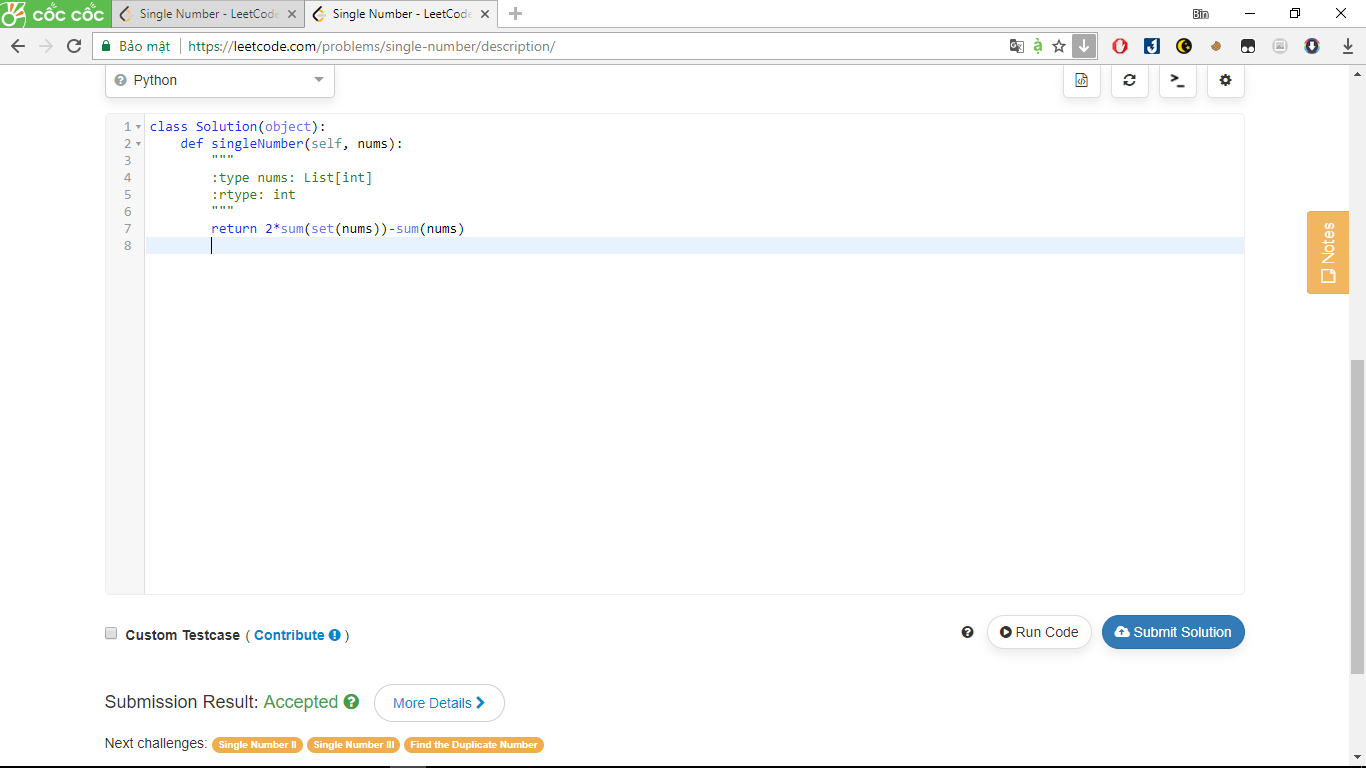
"""

:type nums: List[int]

:rtype: int

"""

return 2\*sum(set(nums))-sum(nums)



766. Toeplitz Matrix

A matrix is *Toeplitz* if every diagonal from top-left to bottom-right has the same element.

Now given an M x N matrix, return True if and only if the matrix is *Toeplitz*.

**Example 1:**

**Input:** matrix = [[1,2,3,4],[5,1,2,3],[9,5,1,2]]

**Output:** True

**Explanation:**

1234

5123

9512

In the above grid, the diagonals are "[9]", "[5, 5]", "[1, 1, 1]", "[2, 2, 2]", "[3, 3]", "[4]", and in each diagonal all elements are the same, so the answer is True.

**Example 2:**

**Input:** matrix = [[1,2],[2,2]]

**Output:** False

**Explanation:**

The diagonal "[1, 2]" has different elements.

**Note:**

1. matrix will be a 2D array of integers.
2. matrix will have a number of rows and columns in range [1, 20].
3. matrix[i][j] will be integers in range [0, 99].

**Answer:**

class Solution(object):

def isToeplitzMatrix(self, matrix):

"""

:type matrix: List[List[int]]

:rtype: bool

"""

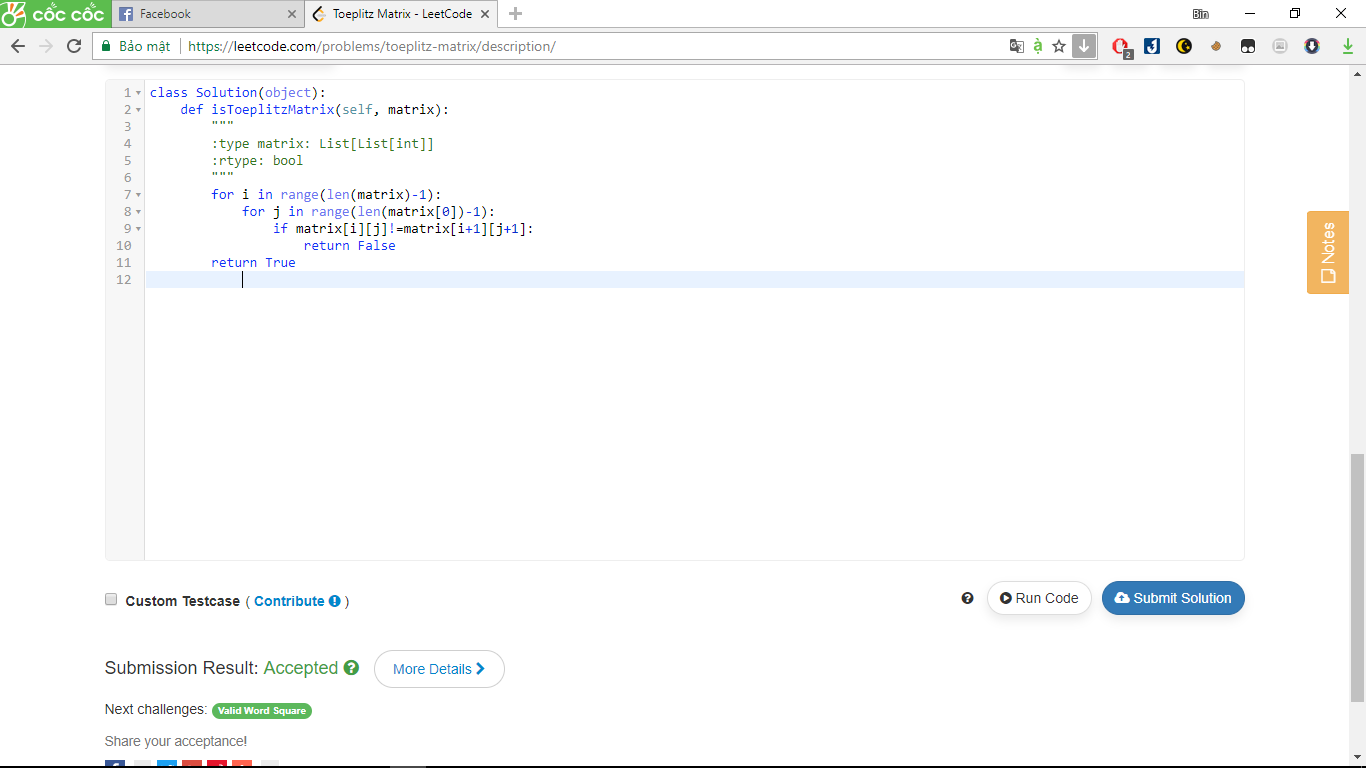
for i in range(len(matrix)-1):

for j in range(len(matrix[0])-1):

if matrix[i][j]!=matrix[i+1][j+1]:

return False

return True



387. First Unique Character in a String

Given a string, find the first non-repeating character in it and return it's index. If it doesn't exist, return -1.

**Examples:**

s = "leetcode"

return 0.

s = "loveleetcode",

return 2.

Answer:

class Solution:

def firstUniqChar(self, s):

"""

:type s: str

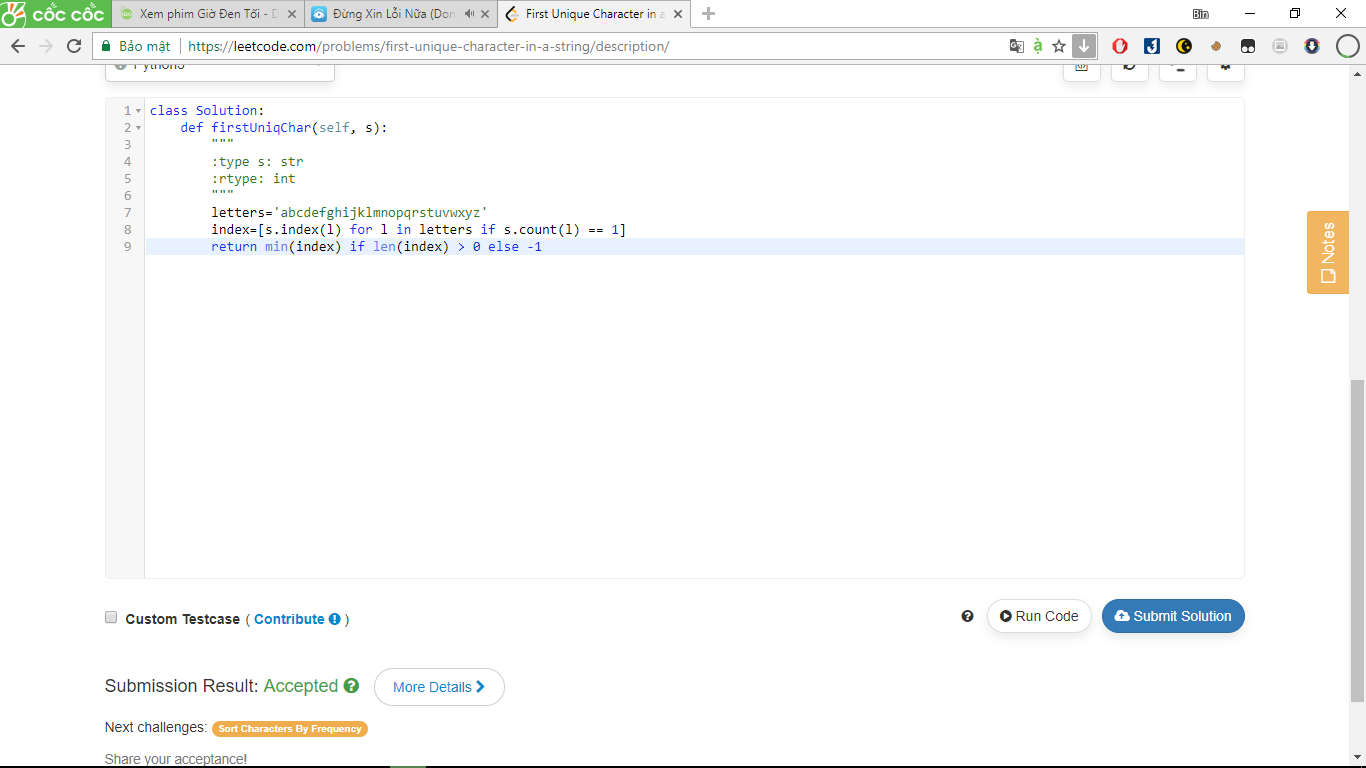
:rtype: int

"""

letters='abcdefghijklmnopqrstuvwxyz'

index=[s.index(l) for l in letters if s.count(l) == 1]

return min(index) if len(index) > 0 else -1



551. Student Attendance Record I

You are given a string representing an attendance record for a student. The record only contains the following three characters:

1. **'A'** : Absent.
2. **'L'** : Late.
3. **'P'** : Present.

A student could be rewarded if his attendance record doesn't contain **more than one 'A' (absent)** or **more than two continuous 'L' (late)**.

You need to return whether the student could be rewarded according to his attendance record.

**Example 1:**

**Input:** "PPALLP"

**Output:** True

**Example 2:**

**Input:** "PPALLL"

**Output:** False

Answer:

class Solution:

def checkRecord(self, s):

"""

:type s: str

:rtype: bool

"""

return s.count('A') <=1 and s.count('LLL')==0



66. Plus One

Given a non-negative integer represented as a **non-empty** array of digits, plus one to the integer.

You may assume the integer do not contain any leading zero, except the number 0 itself.

The digits are stored such that the most significant digit is at the head of the list.

Answer:

class Solution:

def plusOne(self, digits):

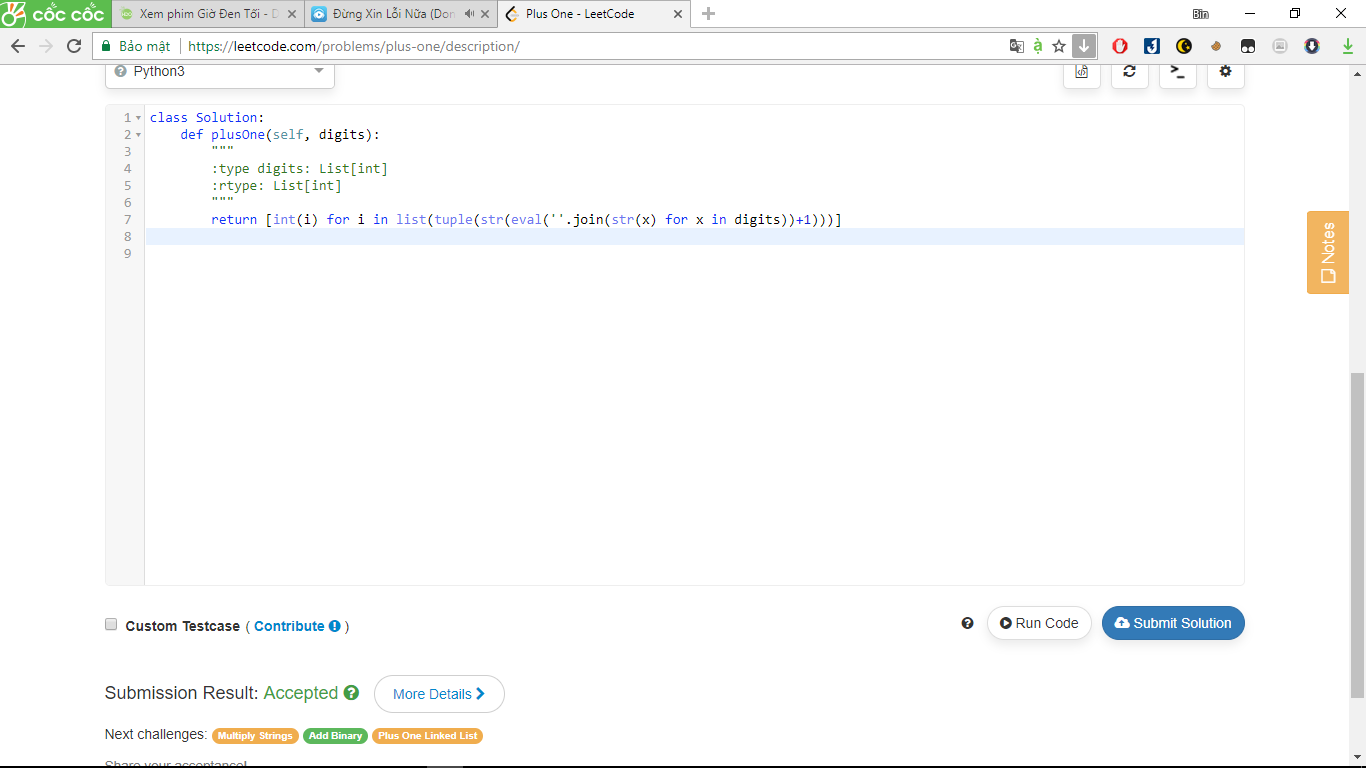
"""

:type digits: List[int]

:rtype: List[int]

"""

return [int(i) for i in list(tuple(str(eval(''.join(str(x) for x in digits))+1)))]



645. Set Mismatch

The set S originally contains numbers from 1 to n. But unfortunately, due to the data error, one of the numbers in the set got duplicated to **another** number in the set, which results in repetition of one number and loss of another number.

Given an array nums representing the data status of this set after the error. Your task is to firstly find the number occurs twice and then find the number that is missing. Return them in the form of an array.

**Example 1:**

**Input:** nums = [1,2,2,4]

**Output:** [2,3]

**Note:**

1. The given array size will in the range [2, 10000].
2. The given array's numbers won't have any order.

Answer:

class Solution:

def findErrorNums(self, nums):

"""

:type nums: List[int]

:rtype: List[int]

"""

return [sum(nums) - sum(set(nums)), sum(range(1, len(nums)+1)) - sum(set(nums))]

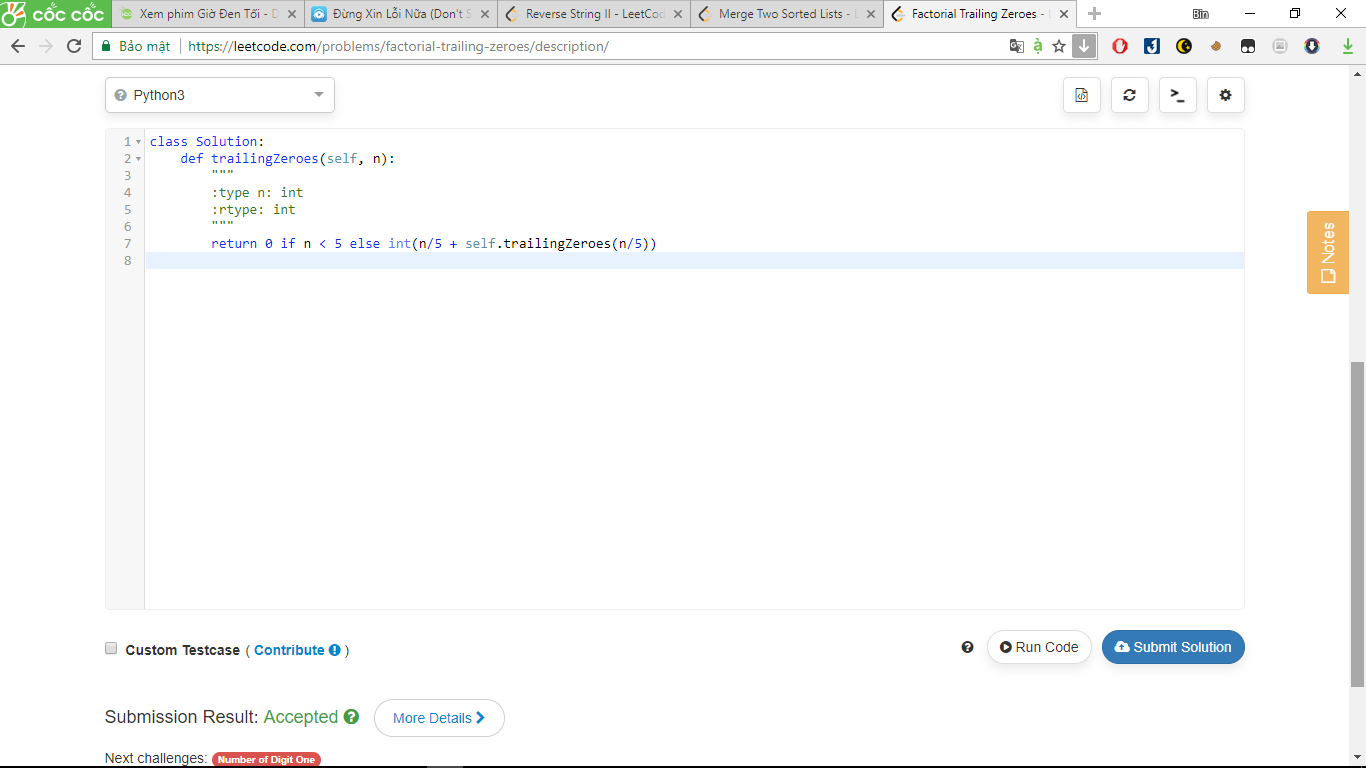


172. Factorial Trailing Zeroes

Given an integer *n*, return the number of trailing zeroes in *n*!.

**Note:**Your solution should be in logarithmic time complexity.

Answer:



541. Reverse String II

Given a string and an integer k, you need to reverse the first k characters for every 2k characters counting from the start of the string. If there are less than k characters left, reverse all of them. If there are less than 2k but greater than or equal to k characters, then reverse the first k characters and left the other as original.

**Example:**

**Input:** s = "abcdefg", k = 2

**Output:** "bacdfeg"

**Restrictions:**

1. The string consists of lower English letters only.
2. Length of the given string and k will in the range [1, 10000]

Answer:

class Solution(object):

def reverseStr(self, s, k):

"""

:type s: str

:type k: int

:rtype: str

"""

if k >= len(s):

return s[::-1]

for i in range(0, len(s), 2 \* k):

s = s[:i]+s[i:i+k][::-1] + s[i+k:]

return s



202. Happy Number

Write an algorithm to determine if a number is "happy".

A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1. Those numbers for which this process ends in 1 are happy numbers.

**Example:**19 is a happy number

* 12 + 92 = 82
* 82 + 22 = 68
* 62 + 82 = 100
* 12 + 02 + 02 = 1

Answer:

class Solution:

def isHappy(self, n):

mem = set()

while n != 1:

n = sum([int(i) \*\* 2 for i in str(n)])

if n in mem:

return False

else:

mem.add(n)

else:

return True



728. Self Dividing Numbers

A *self-dividing number* is a number that is divisible by every digit it contains.

For example, 128 is a self-dividing number because 128 % 1 == 0, 128 % 2 == 0, and 128 % 8 == 0.

Also, a self-dividing number is not allowed to contain the digit zero.

Given a lower and upper number bound, output a list of every possible self dividing number, including the bounds if possible.

**Example 1:**

**Input:**

left = 1, right = 22

**Output:** [1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 15, 22]

Answer:

class Solution:

def selfDividingNumbers(self, left, right):

"""

:type left: int

:type right: int

:rtype: List[int]

"""

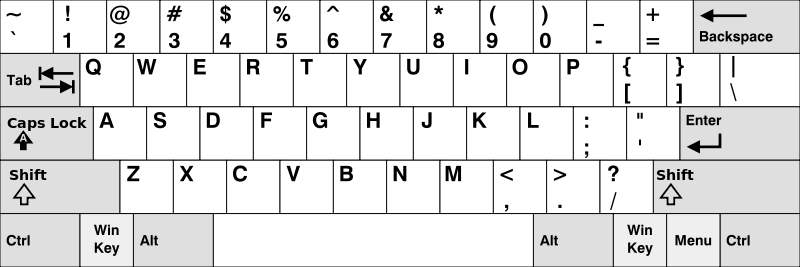
#doan code nay co tham khao vi thay hay

return [i for i in range(left,right+1) if all(int(x) != 0 and (i % int(x)) == 0 for x in str(i))]



500. Keyboard Row

Given a List of words, return the words that can be typed using letters of **alphabet** on only one row's of American keyboard like the image below.



**Example 1:**

**Input:** ["Hello", "Alaska", "Dad", "Peace"]

**Output:** ["Alaska", "Dad"]

**Note:**

1. You may use one character in the keyboard more than once.
2. You may assume the input string will only contain letters of alphabet.

Answer:

class Solution:

def findWords(self, words):

"""

:type words: List[str]

:rtype: List[str]

"""

alpha1 = ['q','w','e','r','t','y','u','i','o','p']

alpha2 = ['a','s','d','f','g','h','j','k','l']

alpha3 = ['z','x','c','v','b','n','m']

rword = []

for w in words:

word = set(w.lower())

if not word.difference(alpha1):

rword.append(w)

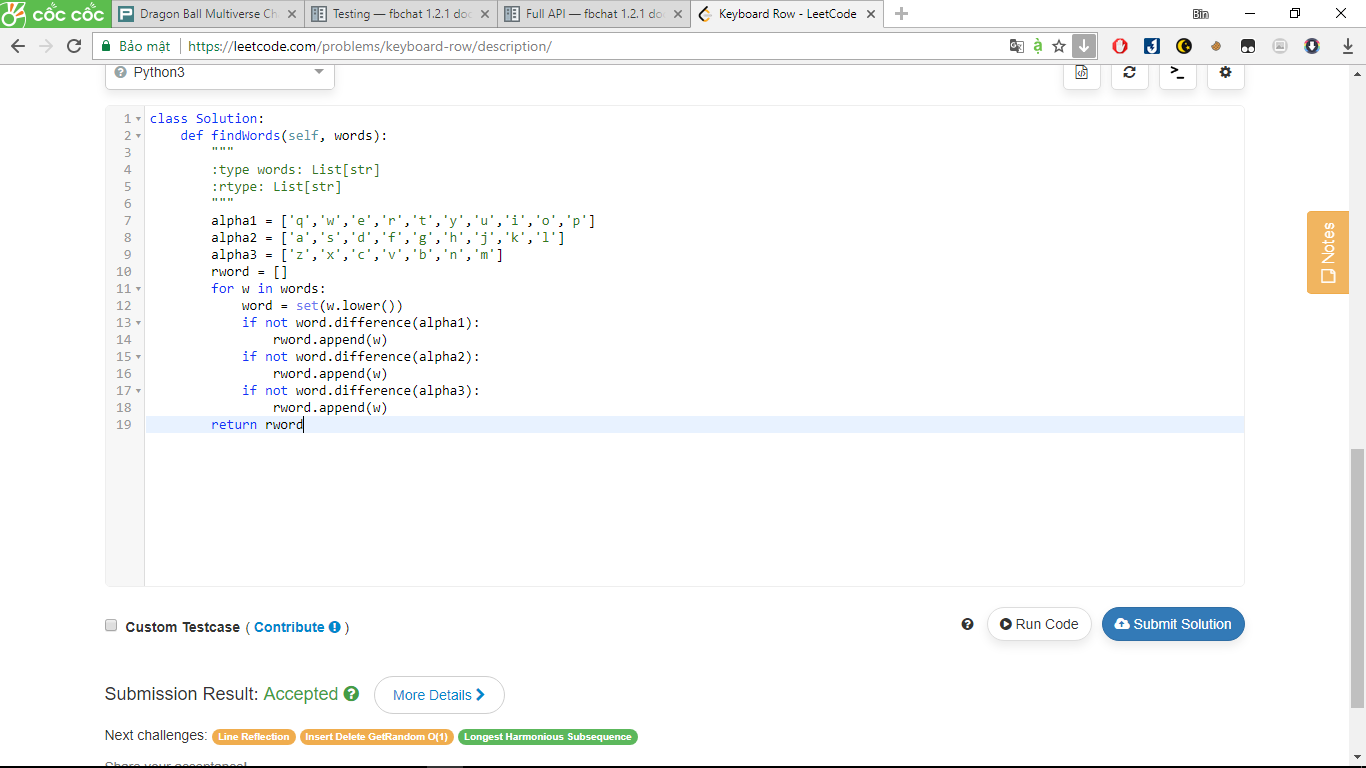
if not word.difference(alpha2):

rword.append(w)

if not word.difference(alpha3):

rword.append(w)

return rword



682. Baseball Game

You're now a baseball game point recorder.

Given a list of strings, each string can be one of the 4 following types:

1. Integer (one round's score): Directly represents the number of points you get in this round.
2. "+" (one round's score): Represents that the points you get in this round are the sum of the last two valid round's points.
3. "D" (one round's score): Represents that the points you get in this round are the doubled data of the last valid round's points.
4. "C" (an operation, which isn't a round's score): Represents the last valid round's points you get were invalid and should be removed.

Each round's operation is permanent and could have an impact on the round before and the round after.

You need to return the sum of the points you could get in all the rounds.

**Example 1:**

**Input:** ["5","2","C","D","+"]

**Output:** 30

**Explanation:**

Round 1: You could get 5 points. The sum is: 5.

Round 2: You could get 2 points. The sum is: 7.

Operation 1: The round 2's data was invalid. The sum is: 5.

Round 3: You could get 10 points (the round 2's data has been removed). The sum is: 15.

Round 4: You could get 5 + 10 = 15 points. The sum is: 30.

**Example 2:**

**Input:** ["5","-2","4","C","D","9","+","+"]

**Output:** 27

**Explanation:**

Round 1: You could get 5 points. The sum is: 5.

Round 2: You could get -2 points. The sum is: 3.

Round 3: You could get 4 points. The sum is: 7.

Operation 1: The round 3's data is invalid. The sum is: 3.

Round 4: You could get -4 points (the round 3's data has been removed). The sum is: -1.

Round 5: You could get 9 points. The sum is: 8.

Round 6: You could get -4 + 9 = 5 points. The sum is 13.

Round 7: You could get 9 + 5 = 14 points. The sum is 27.

Answer:

class Solution(object):

def calPoints(self, ops):

"""

:type ops: List[str]

:rtype: int

"""

s = 0

valid = []

for op in ops:

if (op == 'C'):

s -= valid[-1]

valid.pop()

elif(op == 'D'):

s += 2\*valid[-1]

valid.append(2\*valid[-1])

elif(op == '+'):

s += sum(valid[-2:])

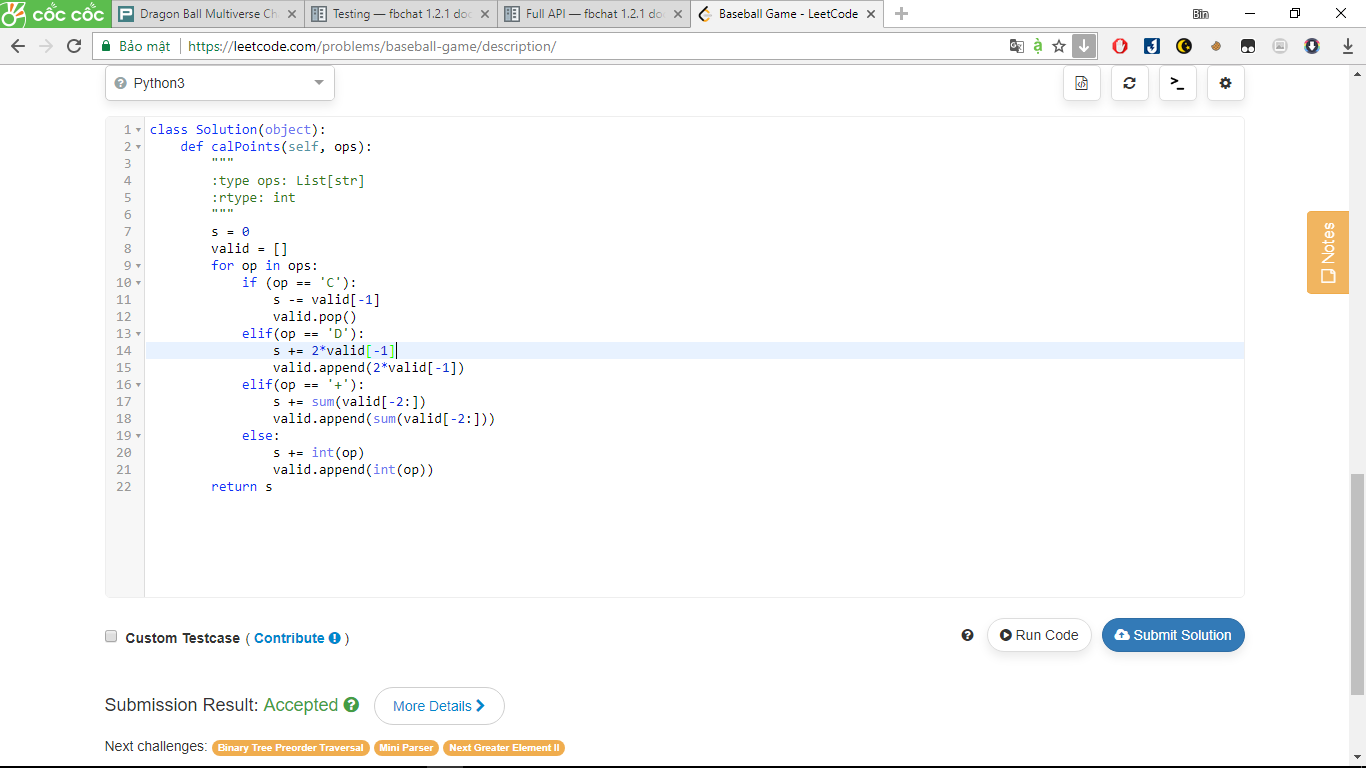
valid.append(sum(valid[-2:]))

else:

s += int(op)

valid.append(int(op))

return s



566. Reshape the Matrix

In MATLAB, there is a very useful function called 'reshape', which can reshape a matrix into a new one with different size but keep its original data.

You're given a matrix represented by a two-dimensional array, and two **positive** integers **r** and **c** representing the **row** number and **column** number of the wanted reshaped matrix, respectively.

The reshaped matrix need to be filled with all the elements of the original matrix in the same **row-traversing** order as they were.

If the 'reshape' operation with given parameters is possible and legal, output the new reshaped matrix; Otherwise, output the original matrix.

**Example 1:**

**Input:**

nums =

[[1,2],

[3,4]]

r = 1, c = 4

**Output:**

[[1,2,3,4]]

**Explanation:**  
The **row-traversing** of nums is [1,2,3,4]. The new reshaped matrix is a 1 \* 4 matrix, fill it row by row by using the previous list.

**Example 2:**

**Input:**

nums =

[[1,2],

[3,4]]

r = 2, c = 4

**Output:**

[[1,2],

[3,4]]

**Explanation:**  
There is no way to reshape a 2 \* 2 matrix to a 2 \* 4 matrix. So output the original matrix.

Answer:

class Solution:

def matrixReshape(self, nums, r, c):

"""

:type nums: List[List[int]]

:type r: int

:type c: int

:rtype: List[List[int]]

"""

tmp = []

list\_t = []

for i in nums:

tmp.extend(i)

s = len(tmp)

if s < r \* c:

return nums

else:

for i in range(r):

t = []

s = i \* c

t.extend(tmp[s:(i+1)\*c])

list\_t.append(t)

return list\_t



349. Intersection of Two Arrays

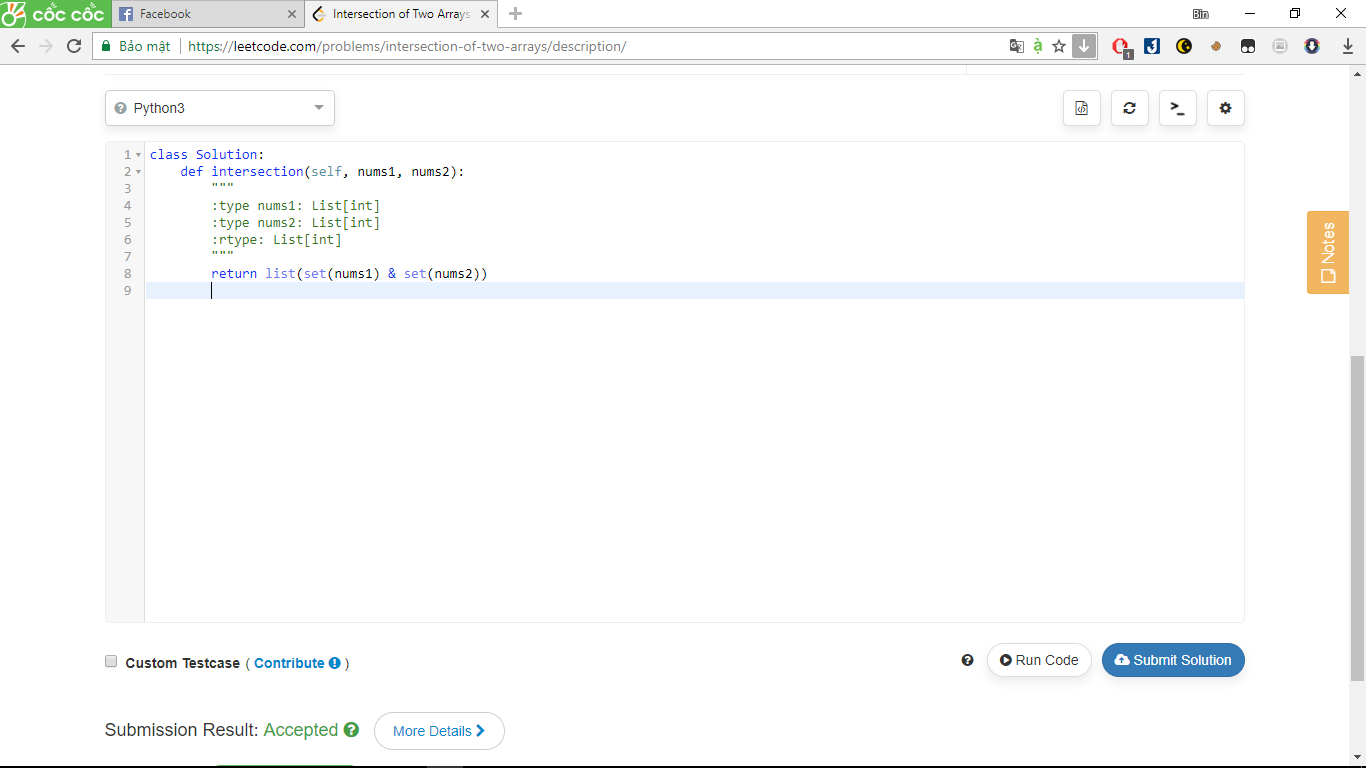
Given two arrays, write a function to compute their intersection.

**Example:**  
Given *nums1* = [1, 2, 2, 1], *nums2* = [2, 2], return [2].

**Note:**

* Each element in the result must be unique.
* The result can be in any order.

Answer:



122. Best Time to Buy and Sell Stock II

Say you have an array for which the *i*th element is the price of a given stock on day *i*.

Design an algorithm to find the maximum profit. You may complete as many transactions as you like (ie, buy one and sell one share of the stock multiple times). However, you may not engage in multiple transactions at the same time (ie, you must sell the stock before you buy again).

Answer:

class Solution:

def maxProfit(self, prices):

"""

:type prices: List[int]

:rtype: int

"""

profit = 0

for i in range(len(prices)-1):

profit += max(prices[i+1] - prices[i], 0)

return profit



169. Majority Element

Given an array of size *n*, find the majority element. The majority element is the element that appears **more than** ⌊ n/2 ⌋ times.

You may assume that the array is non-empty and the majority element always exist in the array.

Answer:

class Solution:

def majorityElement(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

return [x for x in set(nums) if nums.count(x) > len(nums) // 2][0]



13. Roman to Integer

Given a roman numeral, convert it to an integer.

Input is guaranteed to be within the range from 1 to 3999.

Answer:

class Solution:

def romanToInt(self, s):

val = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M': 1000}

ans = val[s[-1]]

for i in range(len(s) - 1):

if val[s[i]] < val[s[i + 1]]:

ans -= val[s[i]]

else:

ans += val[s[i]]

return ans



167. Two Sum II - Input array is sorted

Given an array of integers that is already ***sorted in ascending order***, find two numbers such that they add up to a specific target number.

The function twoSum should return indices of the two numbers such that they add up to the target, where index1 must be less than index2. Please note that your returned answers (both index1 and index2) are not zero-based.

You may assume that each input would have *exactly* one solution and you may not use the *same* element twice.

**Input:** numbers={2, 7, 11, 15}, target=9  
**Output:** index1=1, index2=2

Answer:

class Solution:

def twoSum(self, numbers, target):

"""

:type numbers: List[int]

:type target: int

:rtype: List[int]

"""

#bai nay tham khao

head = 0

tail = len(numbers) - 1

while numbers[head] + numbers[tail] != target:

if numbers[head] + numbers[tail] > target:

tail -= 1

else:

head += 1

return [head + 1, tail + 1]



506. Relative Ranks

Given scores of **N** athletes, find their relative ranks and the people with the top three highest scores, who will be awarded medals: "Gold Medal", "Silver Medal" and "Bronze Medal".

**Example 1:**

**Input:** [5, 4, 3, 2, 1]

**Output:** ["Gold Medal", "Silver Medal", "Bronze Medal", "4", "5"]

**Explanation:** The first three athletes got the top three highest scores, so they got "Gold Medal", "Silver Medal" and "Bronze Medal".   
For the left two athletes, you just need to output their relative ranks according to their scores.

**Note:**

1. N is a positive integer and won't exceed 10,000.
2. All the scores of athletes are guaranteed to be unique.

Answer:

class Solution:

def findRelativeRanks(self, nums):

"""

:type nums: List[int]

:rtype: List[str]

"""

#bai nay tham khao

result=[]

rank=sorted(nums)

for i in range(len(nums)):

if rank.index(nums[i])==len(nums)-1:

result.append('Gold Medal')

elif rank.index(nums[i])==len(nums)-2:

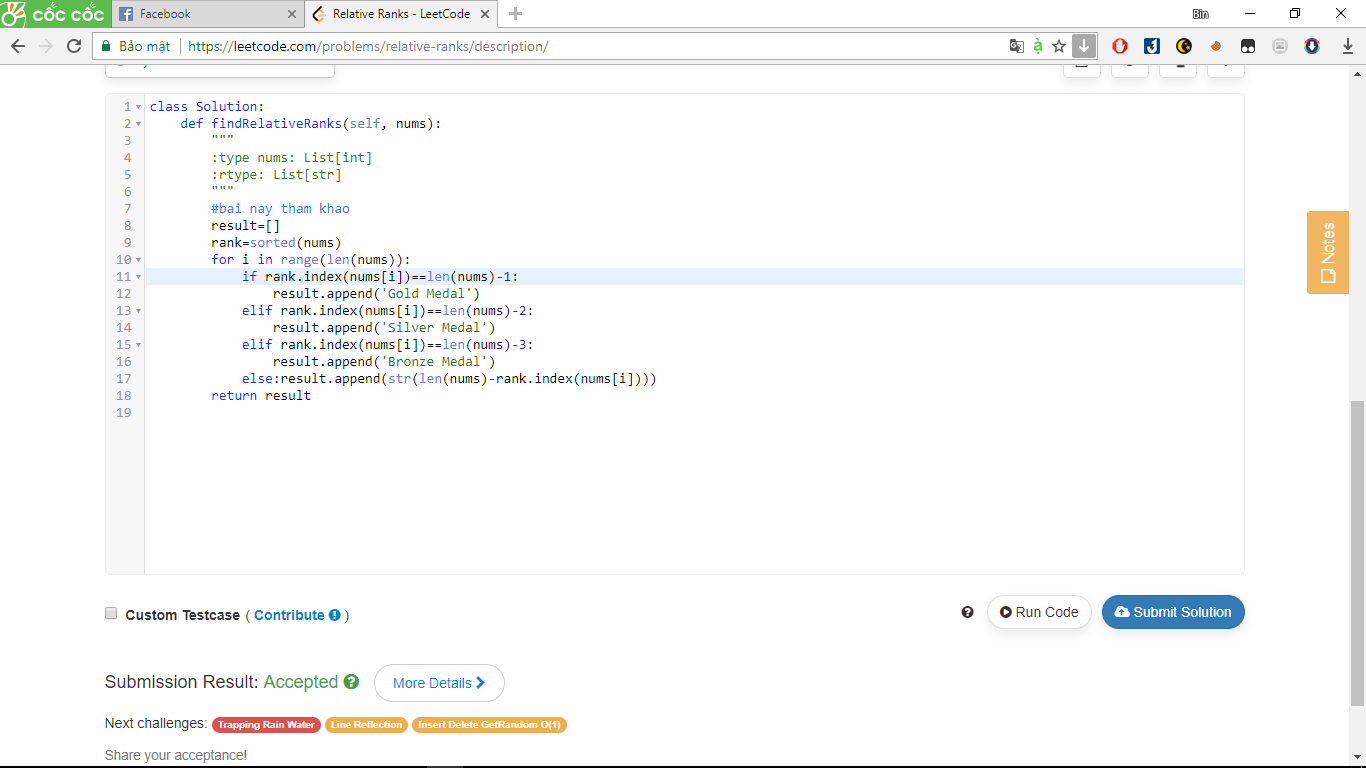
result.append('Silver Medal')

elif rank.index(nums[i])==len(nums)-3:

result.append('Bronze Medal')

else:result.append(str(len(nums)-rank.index(nums[i])))

return result



409. Longest Palindrome

Given a string which consists of lowercase or uppercase letters, find the length of the longest palindromes that can be built with those letters.

This is case sensitive, for example "Aa" is not considered a palindrome here.

**Note:**  
Assume the length of given string will not exceed 1,010.

**Example:**

Input:

"abccccdd"

Output:

7

Explanation:

One longest palindrome that can be built is "dccaccd", whose length is 7.

Answer:

class Solution:

def longestPalindrome(self, s):

"""

:type s: str

:rtype: int

"""

cnt = collections.Counter(s).values()

return sum(i&~1 for i in cnt) + any(i%2 for i in cnt)



350. Intersection of Two Arrays II

Given two arrays, write a function to compute their intersection.

**Example:**  
Given *nums1* = [1, 2, 2, 1], *nums2* = [2, 2], return [2, 2].

**Note:**

* Each element in the result should appear as many times as it shows in both arrays.
* The result can be in any order.

**Follow up:**

* What if the given array is already sorted? How would you optimize your algorithm?
* What if *nums1*'s size is small compared to *nums2*'s size? Which algorithm is better?
* What if elements of *nums2* are stored on disk, and the memory is limited such that you cannot load all elements into the memory at once?

Answer:

class Solution:

def intersect(self, nums1, nums2):

"""

:type nums1: List[int]

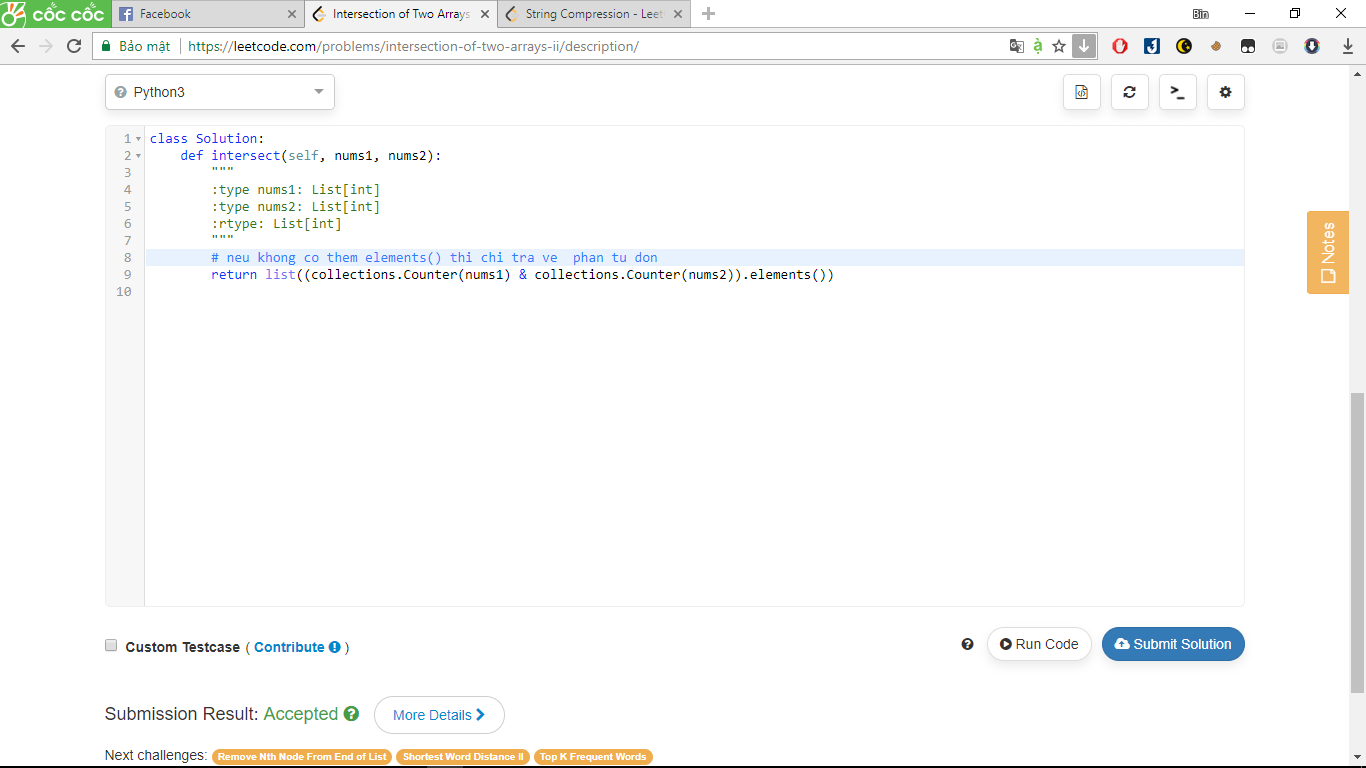
:type nums2: List[int]

:rtype: List[int]

"""

# neu khong co them elements() thi chi tra ve phan tu don

return list((collections.Counter(nums1) & collections.Counter(nums2)).elements())



345. Reverse Vowels of a String

Write a function that takes a string as input and reverse only the vowels of a string.

**Example 1:**  
Given s = "hello", return "holle".

**Example 2:**  
Given s = "leetcode", return "leotcede".

**Note:**  
The vowels does not include the letter "y".

Answer:

class Solution:

def reverseVowels(self, s):

"""

:type s: str

:rtype: str

"""

vowels = set(['a','e','i','o','u','A','E','I','O','U'])

pool = [ i for i in s if i in vowels ]

return "".join( [ i if i not in vowels else pool.pop() for i in s ] )



496. Next Greater Element I

You are given two arrays **(without duplicates)** nums1 and nums2 where nums1’s elements are subset of nums2. Find all the next greater numbers for nums1's elements in the corresponding places of nums2.

The Next Greater Number of a number **x** in nums1 is the first greater number to its right in nums2. If it does not exist, output -1 for this number.

**Example 1:**

**Input:** **nums1** = [4,1,2], **nums2** = [1,3,4,2].

**Output:** [-1,3,-1]

**Explanation:**

For number 4 in the first array, you cannot find the next greater number for it in the second array, so output -1.

For number 1 in the first array, the next greater number for it in the second array is 3.

For number 2 in the first array, there is no next greater number for it in the second array, so output -1.

**Example 2:**

**Input:** **nums1** = [2,4], **nums2** = [1,2,3,4].

**Output:** [3,-1]

**Explanation:**

For number 2 in the first array, the next greater number for it in the second array is 3.

For number 4 in the first array, there is no next greater number for it in the second array, so output -1.

**Note:**

1. All elements in nums1 and nums2 are unique.
2. The length of both nums1 and nums2 would not exceed 1000.

Answer:

class Solution:

def nextGreaterElement(self, findNums, nums):

"""

:type nums1: List[int]

:type nums2: List[int]

:rtype: List[int]

"""

#bai nay tham khao

h, stack = {}, []

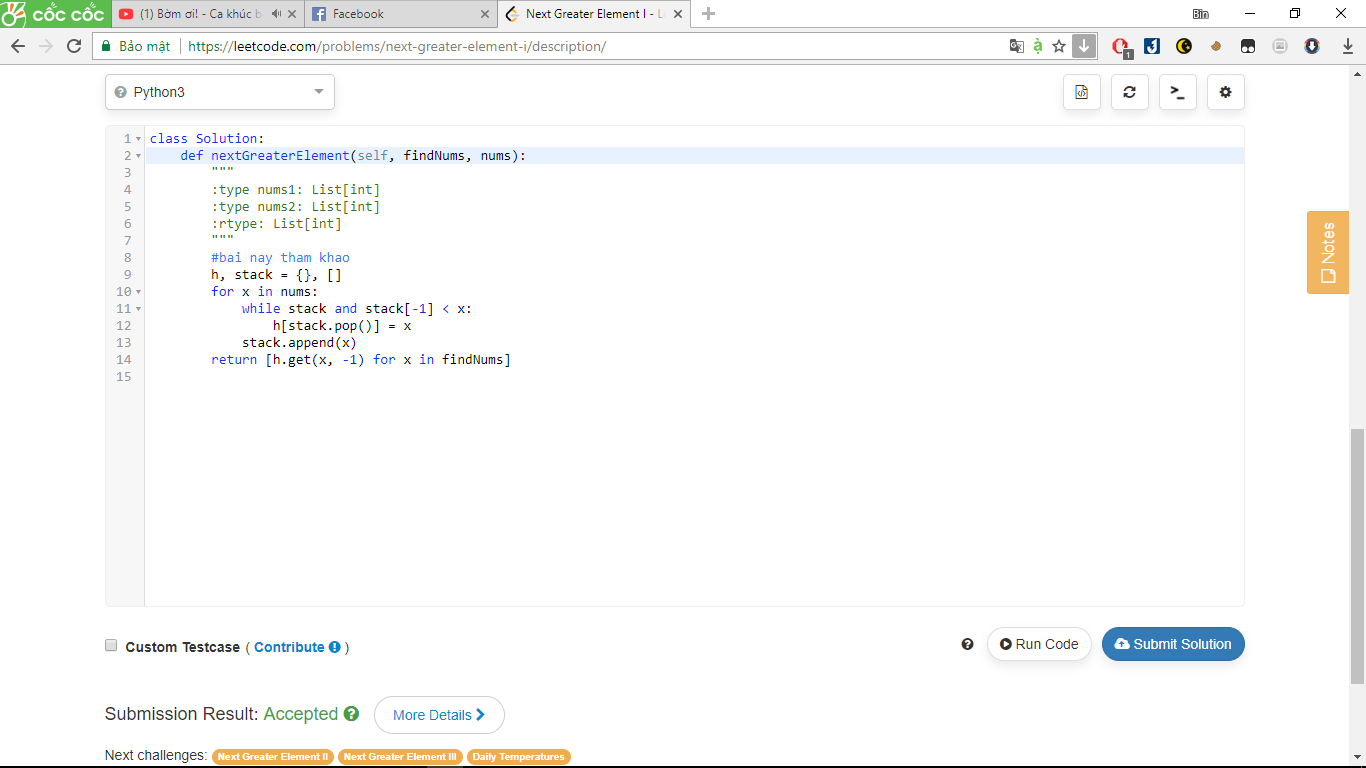
for x in nums:

while stack and stack[-1] < x:

h[stack.pop()] = x

stack.append(x)

return [h.get(x, -1) for x in findNums]



717. 1-bit and 2-bit Characters

We have two special characters. The first character can be represented by one bit 0. The second character can be represented by two bits (10 or 11).

Now given a string represented by several bits. Return whether the last character must be a one-bit character or not. The given string will always end with a zero.

**Example 1:**

**Input:**

bits = [1, 0, 0]

**Output:** True

**Explanation:**

The only way to decode it is two-bit character and one-bit character. So the last character is one-bit character.

**Example 2:**

**Input:**

bits = [1, 1, 1, 0]

**Output:** False

**Explanation:**

The only way to decode it is two-bit character and two-bit character. So the last character is NOT one-bit character.

**Note:**

 1 <= len(bits) <= 1000.

 bits[i] is always 0 or 1.

Answer:

class Solution:

def isOneBitCharacter(self, bits):

"""

:type bits: List[int]

:rtype: bool

"""

#bai nay tham khoe

index = 0

while index < len(bits) - 1:

index += 2 if bits[index] == 1 else 1

return index == len(bits) - 1



504. Base 7

Given an integer, return its base 7 string representation.

**Example 1:**

**Input:** 100

**Output:** "202"

**Example 2:**

**Input:** -7

**Output:** "-10"

**Note:** The input will be in range of [-1e7, 1e7].

Answer:

class Solution:

def convertToBase7(self, num):

"""

:type num: int

:rtype: str

"""

if num == 0: return '0'

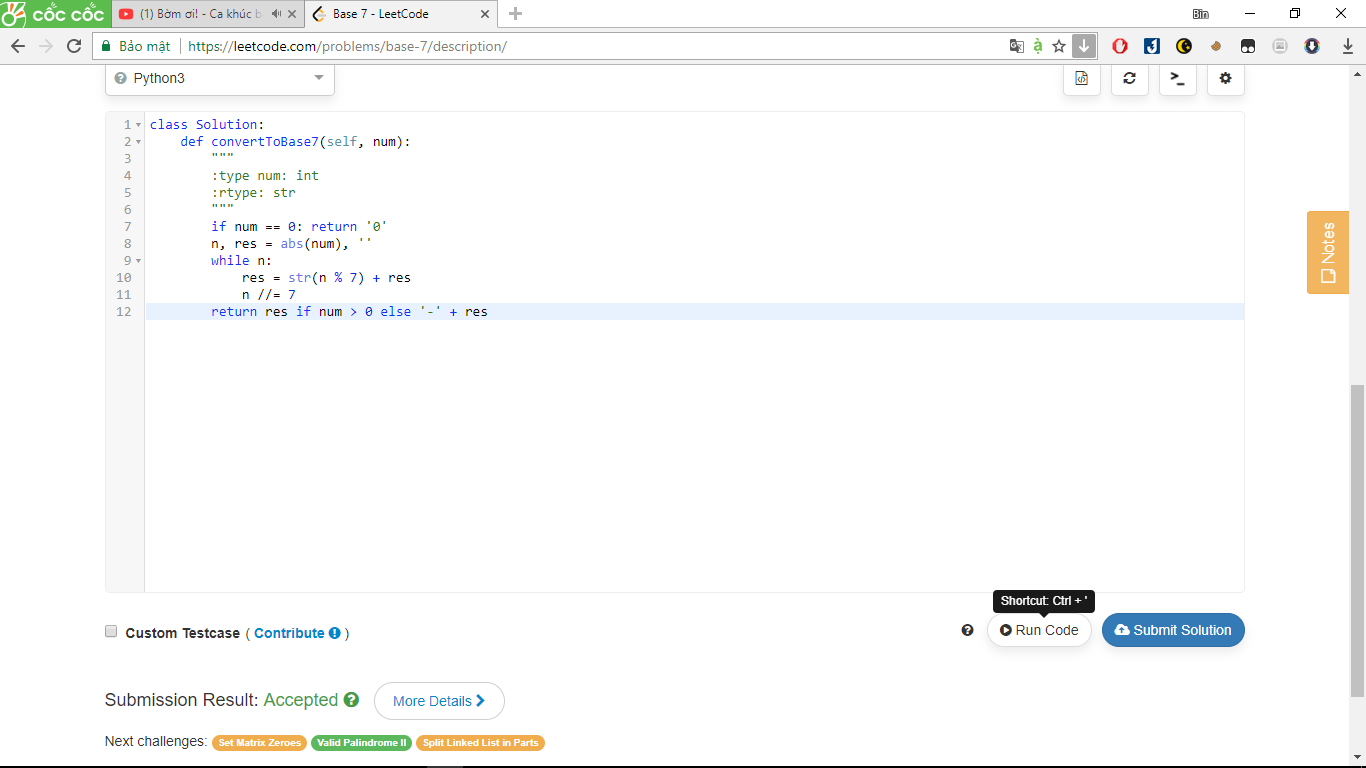
n, res = abs(num), ''

while n:

res = str(n % 7) + res

n //= 7

return res if num > 0 else '-' + res



118. Pascal's Triangle

Given *numRows*, generate the first *numRows* of Pascal's triangle.

For example, given *numRows* = 5,  
Return

[

[1],

[1,1],

[1,2,1],

[1,3,3,1],

[1,4,6,4,1]

]

Answer:

class Solution:

def generate(self, numRows):

"""

:type numRows: int

:rtype: List[List[int]]

"""

#bai nay tham khao

pascal = [[1]\*(i+1) for i in range(numRows)]

for i in range(numRows):

for j in range(1,i):

pascal[i][j] = pascal[i-1][j-1] + pascal[i-1][j]

return pascal



119. Pascal's Triangle II

Given an index *k*, return the *k*th row of the Pascal's triangle.

For example, given *k* = 3,  
Return [1,3,3,1].

Answer:

class Solution:

def getRow(self, rowIndex):

"""

:type rowIndex: int

:rtype: List[int]

"""

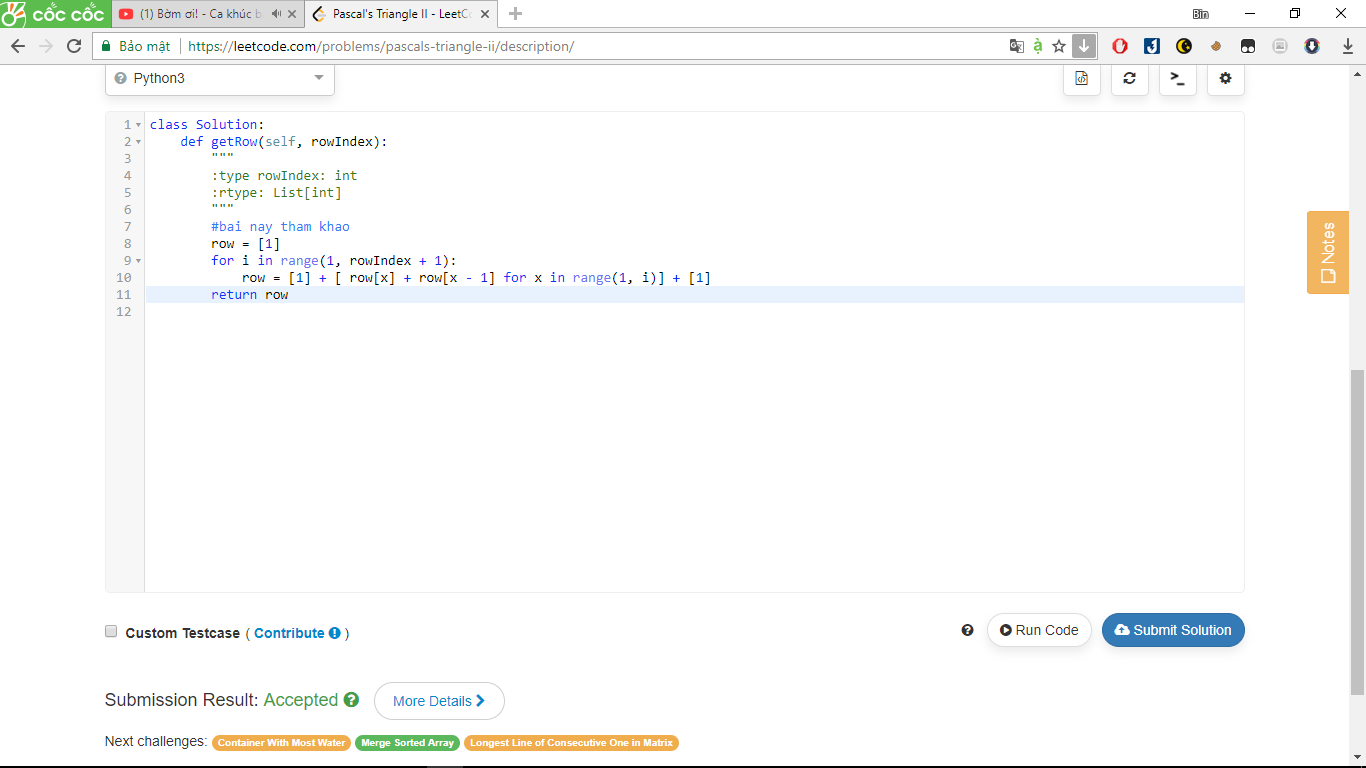
#bai nay tham khao

row = [1]

for i in range(1, rowIndex + 1):

row = [1] + [ row[x] + row[x - 1] for x in range(1, i)] + [1]

return row



205. Isomorphic Strings(bài nay hay)

Given two strings ***s*** and ***t***, determine if they are isomorphic.

Two strings are isomorphic if the characters in ***s*** can be replaced to get ***t***.

All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character but a character may map to itself.

For example,  
Given "egg", "add", return true.

Given "foo", "bar", return false.

Given "paper", "title", return true.

Answer:

class Solution:

def isIsomorphic(self, s, t):

"""

:type s: str

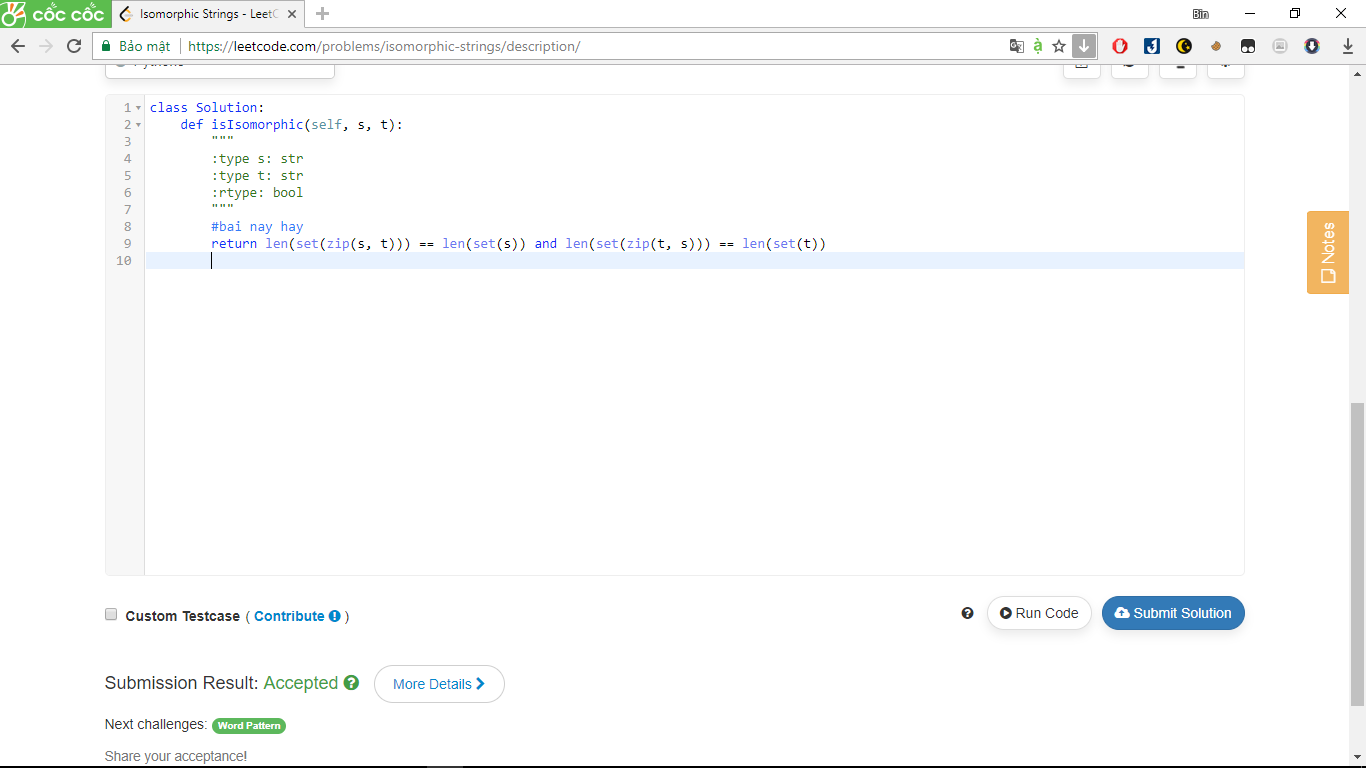
:type t: str

:rtype: bool

"""

#bai nay hay

return len(set(zip(s, t))) == len(set(s)) and len(set(zip(t, s))) == len(set(t))



20. Valid Parentheses

Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

The brackets must close in the correct order, "()" and "()[]{}" are all valid but "(]" and "([)]" are not.

Answer:

class Solution:

def isValid(self, s):

"""

:type s: str

:rtype: bool

"""

n = len(s)

if n == 0:

return True

if n % 2 != 0:

return False

while "()" in s or "{}" in s or "[]" in s:

s = s.replace("()","").replace("{}","").replace("[]","")

return len(s) == 0



441. Arranging Coins

You have a total of *n* coins that you want to form in a staircase shape, where every *k*-th row must have exactly *k* coins.

Given *n*, find the total number of **full** staircase rows that can be formed.

*n* is a non-negative integer and fits within the range of a 32-bit signed integer.

**Example 1:**

n = 5

The coins can form the following rows:

¤

¤ ¤

¤ ¤

Because the 3rd row is incomplete, we return 2.

**Example 2:**

n = 8

The coins can form the following rows:

¤

¤ ¤

¤ ¤ ¤

¤ ¤

Because the 4th row is incomplete, we return 3.

Ta thấy :

1+2+3+…+x = n  
-> (1+x)x/2 = n  
-> x^2+x = 2n  
-> x^2+x+1/4 = 2n +1/4  
-> (x+1/2)^2 = 2n +1/4  
-> (x+0.5) = sqrt(2n+0.25)  
-> x = -0.5 + sqrt(2n+0.25)

Answer:

class Solution:

def arrangeCoins(self, n):

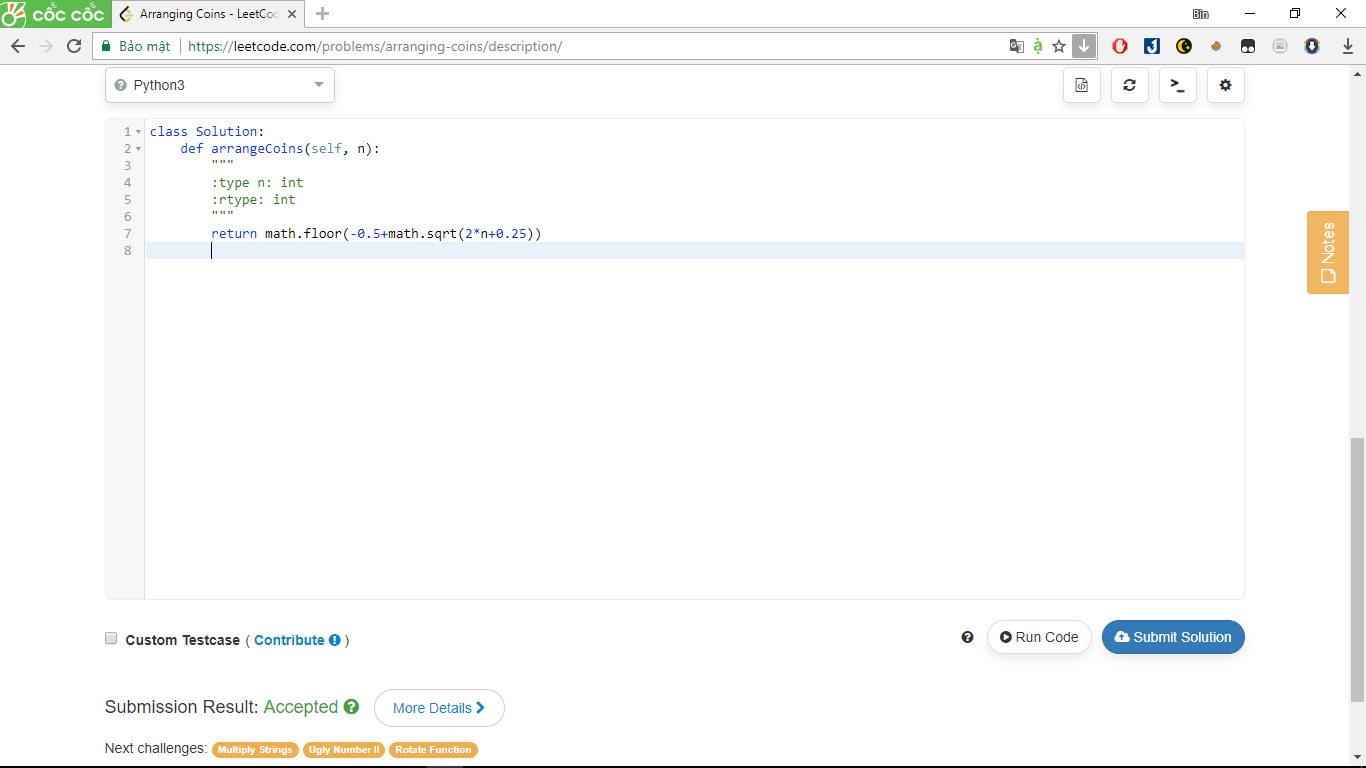
"""

:type n: int

:rtype: int

"""

return math.floor(-0.5+math.sqrt(2\*n+0.25))



374. Guess Number Higher or Lower

We are playing the Guess Game. The game is as follows:

I pick a number from **1** to ***n***. You have to guess which number I picked.

Every time you guess wrong, I'll tell you whether the number is higher or lower.

You call a pre-defined API guess(int num) which returns 3 possible results (-1, 1, or 0):

-1 : My number is lower

1 : My number is higher

0 : Congrats! You got it!

**Example:**

n = 10, I pick 6.

Return 6.

Answer:

# The guess API is already defined for you.

# @param num, your guess

# @return -1 if my number is lower, 1 if my number is higher, otherwise return 0

# def guess(num):

class Solution(object):

def guessNumber(self, n):

"""

:type n: int

:rtype: int

"""

#bai nay tham khao

low = 1

high = n

while low <= high:

mid = low + (high-low)//2

ans = guess(mid)

if ans == 0:

return mid

elif ans == -1:

high = mid - 1

else:

low = mid + 1



26. Remove Duplicates from Sorted Array

Given a sorted array, remove the duplicates [**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) such that each element appear only *once* and return the new length.

Do not allocate extra space for another array, you must do this by **modifying the input array**[**in-place**](https://en.wikipedia.org/wiki/In-place_algorithm) with O(1) extra memory.

**Example:**

Given *nums* = [1,1,2],

Your function should return length = 2, with the first two elements of *nums* being 1 and 2 respectively.

It doesn't matter what you leave beyond the new length.

Answer:

class Solution(object):

def removeDuplicates(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

#bai nay tham khao

if not nums:

return 0

newTail = 0

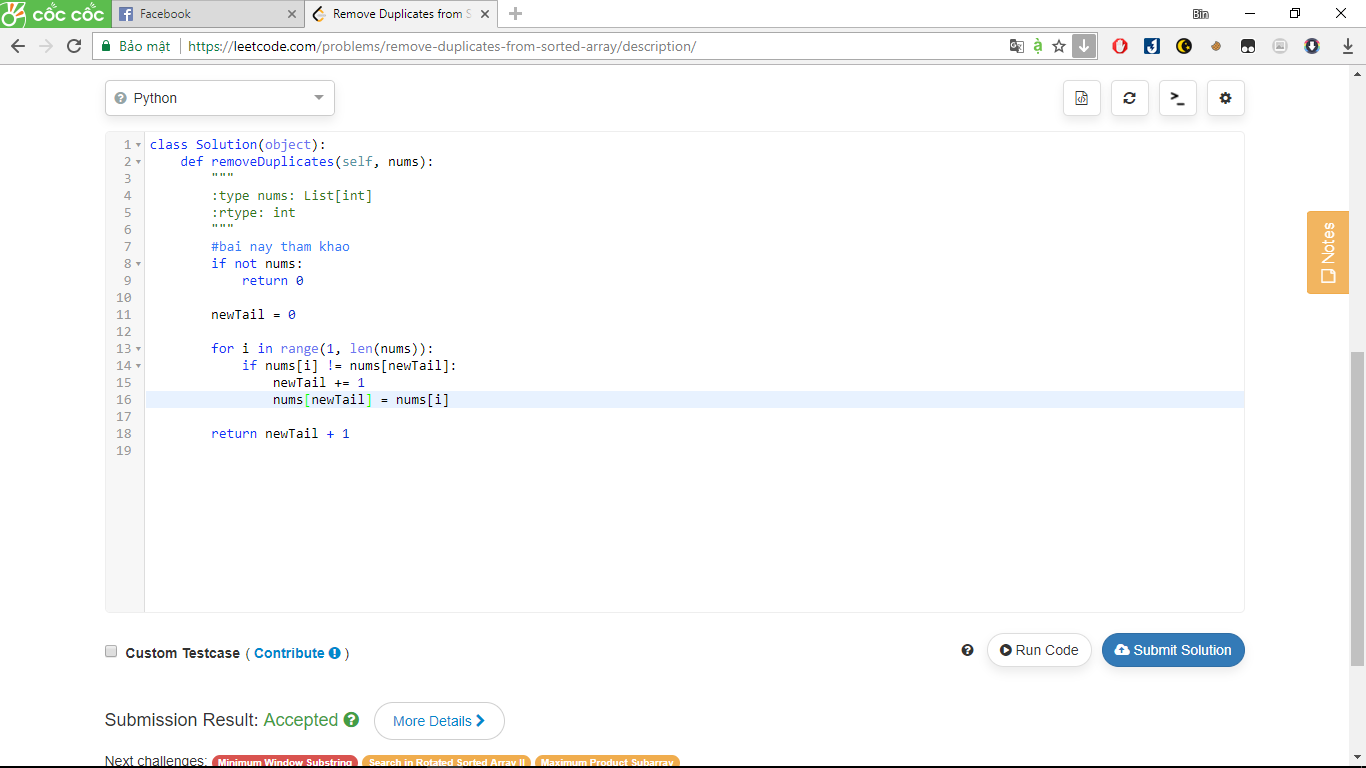
for i in range(1, len(nums)):

if nums[i] != nums[newTail]:

newTail += 1

nums[newTail] = nums[i]

return newTail + 1



643. Maximum Average Subarray I

Given an array consisting of n integers, find the contiguous subarray of given length k that has the maximum average value. And you need to output the maximum average value.

**Example 1:**

**Input:** [1,12,-5,-6,50,3], k = 4

**Output:** 12.75

**Explanation:** Maximum average is (12-5-6+50)/4 = 51/4 = 12.75

**Note:**

1. 1 <= k <= n <= 30,000.
2. Elements of the given array will be in the range [-10,000, 10,000].

Answer:

class Solution:

def findMaxAverage(self, nums, k):

"""

:type nums: List[int]

:type k: int

:rtype: float

"""

#bai nay tham khao

if len(nums) <= k:

return sum(nums) / k

i = 0

cs = sum(nums[i:i+k])

ma = cs / k

while True:

i += 1

if (i > len(nums) - k):

break

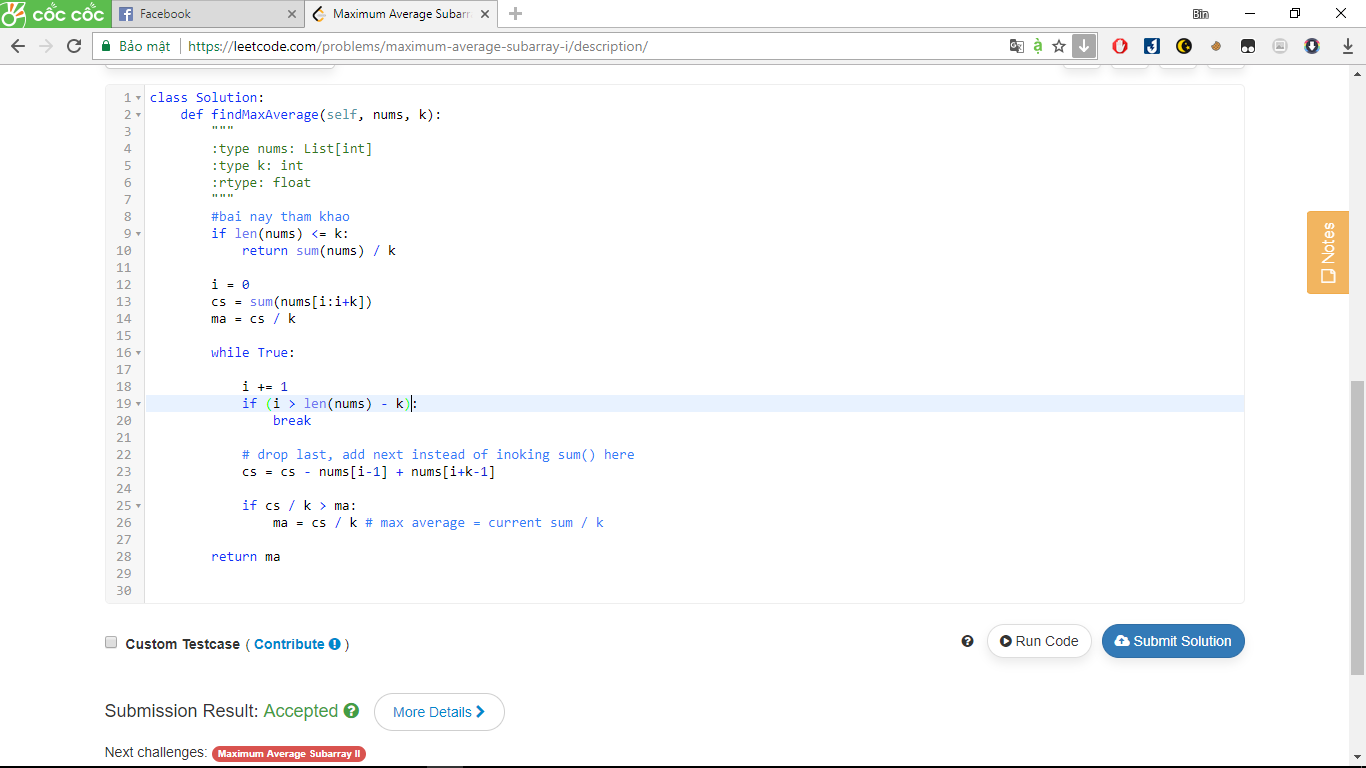
# drop last, add next instead of inoking sum() here

cs = cs - nums[i-1] + nums[i+k-1]

if cs / k > ma:

ma = cs / k # max average = current sum / k

return ma



83. Remove Duplicates from Sorted List

Given a sorted linked list, delete all duplicates such that each element appear only *once*.

For example,  
Given 1->1->2, return 1->2.  
Given 1->1->2->3->3, return 1->2->3.

Answer:

# Definition for singly-linked list.

# class ListNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.next = None

class Solution:

def deleteDuplicates(self, head):

"""

:type head: ListNode

:rtype: ListNode

"""

first, second = head, head.next if head else None

while second:

if first.val == second.val:

second = second.next

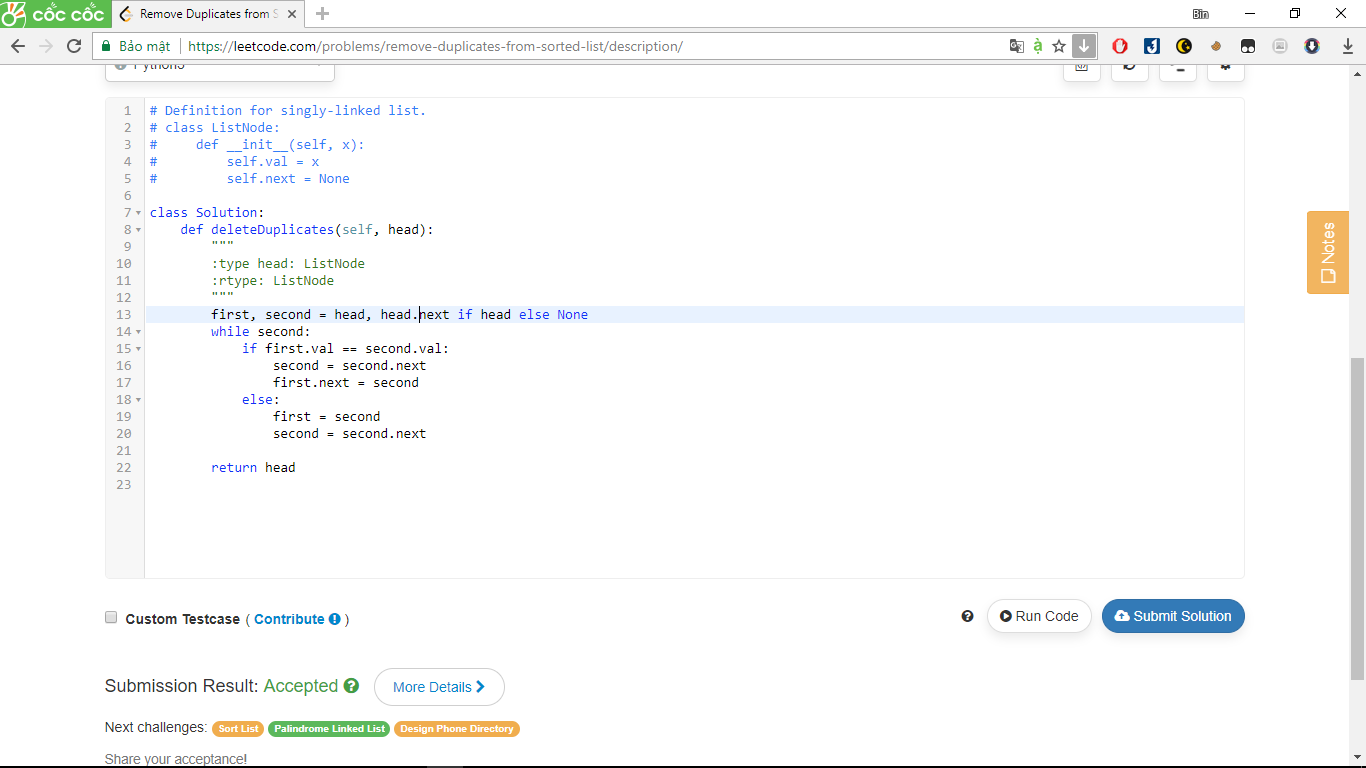
first.next = second

else:

first = second

second = second.next

return head



53. Maximum Subarray

Find the contiguous subarray within an array (containing at least one number) which has the largest sum.

For example, given the array [-2,1,-3,4,-1,2,1,-5,4],  
the contiguous subarray [4,-1,2,1] has the largest sum = 6.

[click to show more practice.](https://leetcode.com/problems/maximum-subarray/description/)

**More practice:**

If you have figured out the O(*n*) solution, try coding another solution using the divide and conquer approach, which is more subtle.

Answer:

class Solution:

def maxSubArray(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

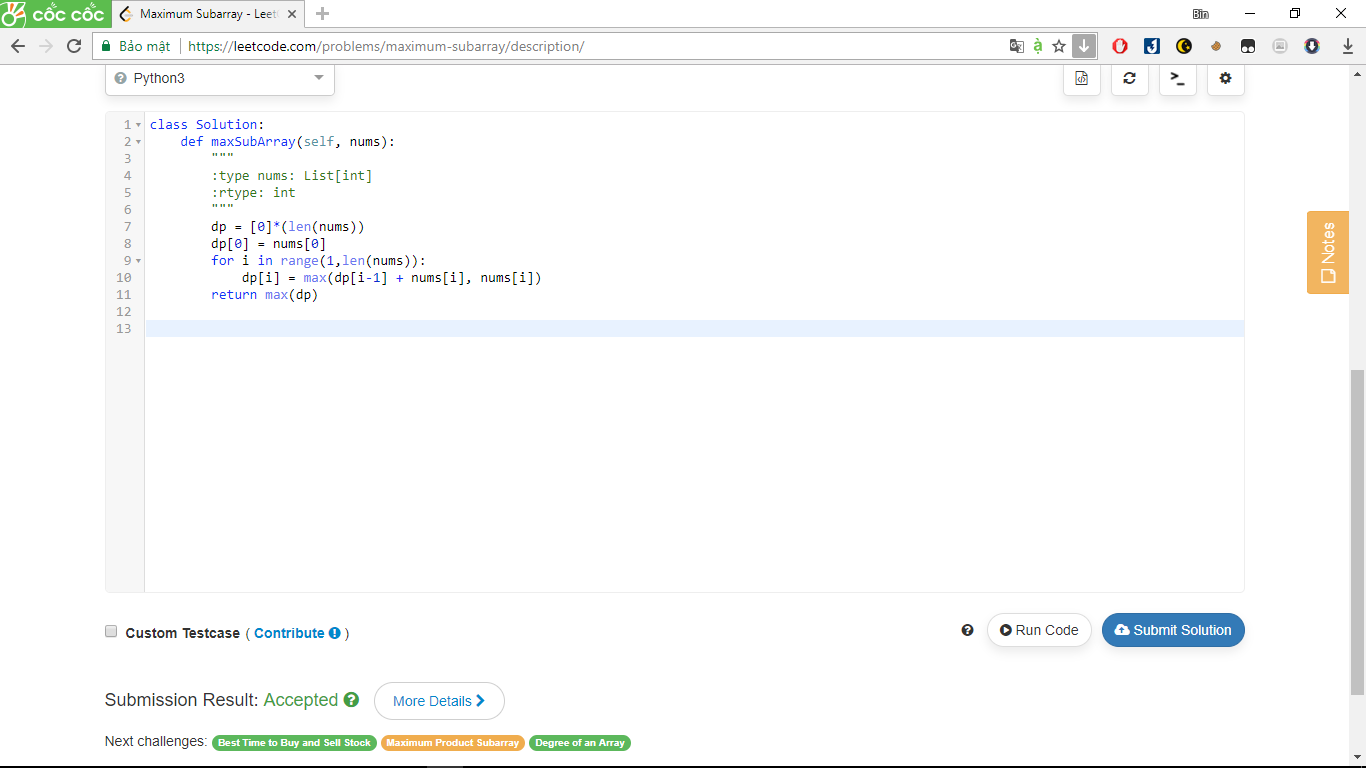
dp = [0]\*(len(nums))

dp[0] = nums[0]

for i in range(1,len(nums)):

dp[i] = max(dp[i-1] + nums[i], nums[i])

return max(dp)



70. Climbing Stairs

You are climbing a stair case. It takes *n* steps to reach to the top.

Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

**Note:** Given *n* will be a positive integer.

**Example 1:**

**Input:** 2

**Output:** 2

**Explanation:** There are two ways to climb to the top.

1. 1 step + 1 step

2. 2 steps

**Example 2:**

**Input:** 3

**Output:** 3

**Explanation:** There are three ways to climb to the top.

1. 1 step + 1 step + 1 step

2. 1 step + 2 steps

3. 2 steps + 1 step

Answer:

class Solution:

def climbStairs(self, n):

"""

:type n: int

:rtype: int

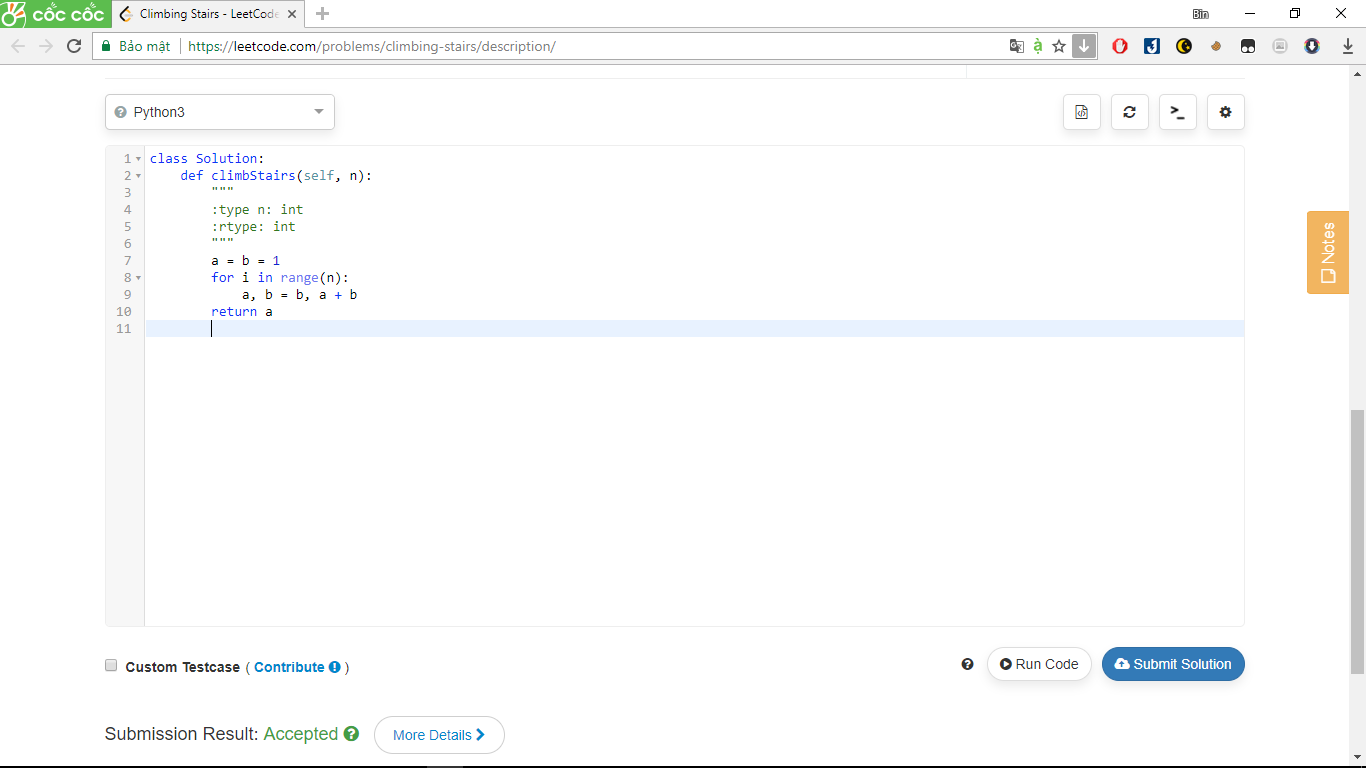
"""

a = b = 1

for i in range(n):

a, b = b, a + b

return a



405. Convert a Number to Hexadecimal

Given an integer, write an algorithm to convert it to hexadecimal. For negative integer, [two’s complement](https://en.wikipedia.org/wiki/Two%27s_complement) method is used.

**Note:**

1. All letters in hexadecimal (a-f) must be in lowercase.
2. The hexadecimal string must not contain extra leading 0s. If the number is zero, it is represented by a single zero character '0'; otherwise, the first character in the hexadecimal string will not be the zero character.
3. The given number is guaranteed to fit within the range of a 32-bit signed integer.
4. You **must not use *any* method provided by the library** which converts/formats the number to hex directly.

**Example 1:**

Input:

26

Output:

"1a"

**Example 2:**

Input:

-1

Output:

"ffffffff"

Answer:

class Solution:

def toHex(self, num):

"""

:type num: int

:rtype: str

"""

#bai nay tham khao

ret = ''

letter = ('0', '1','2','3','4','5','6','7','8','9','a','b','c','d','e','f')

if num == 0: return '0'

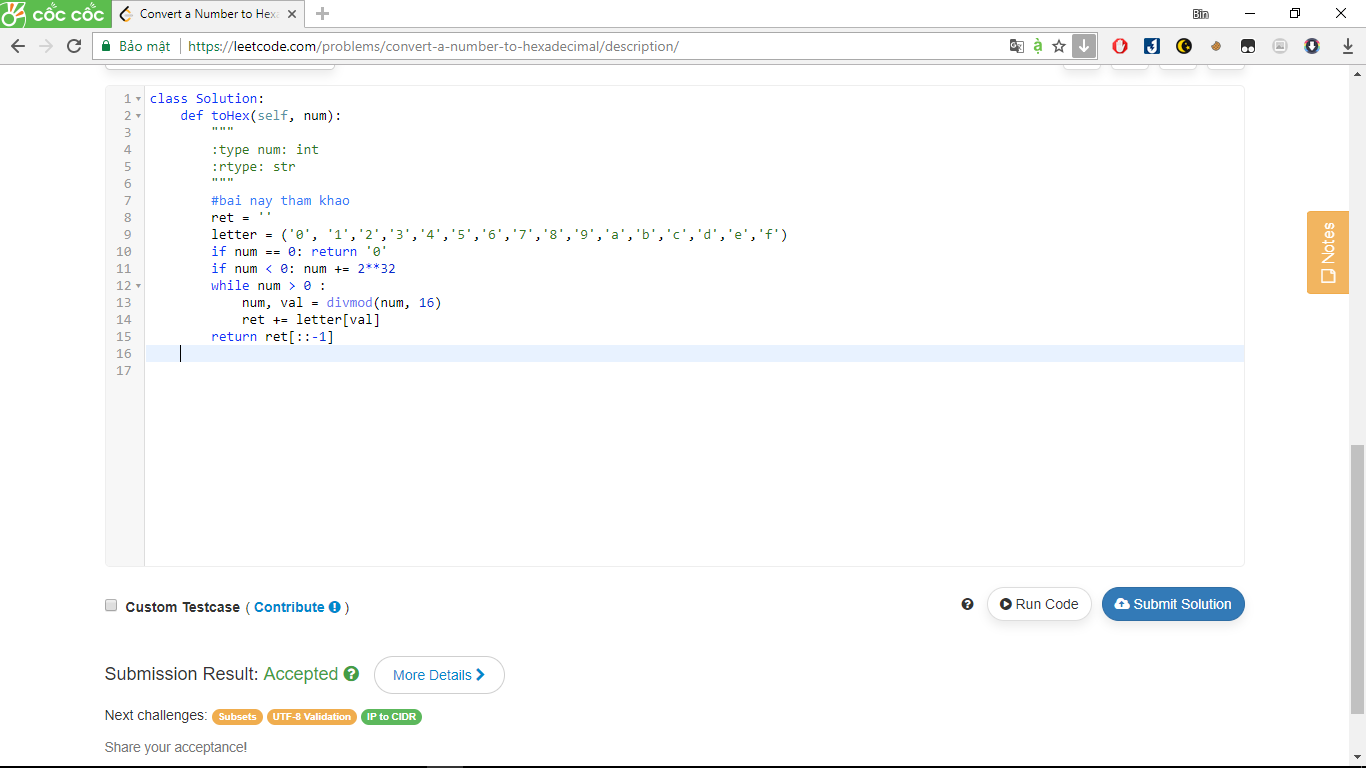
if num < 0: num += 2\*\*32

while num > 0 :

num, val = divmod(num, 16)

ret += letter[val]

return ret[::-1]



594. Longest Harmonious Subsequence

We define a harmonious array is an array where the difference between its maximum value and its minimum value is **exactly** 1.

Now, given an integer array, you need to find the length of its longest harmonious subsequence among all its possible [subsequences](https://en.wikipedia.org/wiki/Subsequence).

**Example 1:**

**Input:** [1,3,2,2,5,2,3,7]

**Output:** 5

**Explanation:** The longest harmonious subsequence is [3,2,2,2,3].

**Note:** The length of the input array will not exceed 20,000.

Answer:

class Solution:

def findLHS(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

#bai nay tham khao

count = collections.Counter(nums)

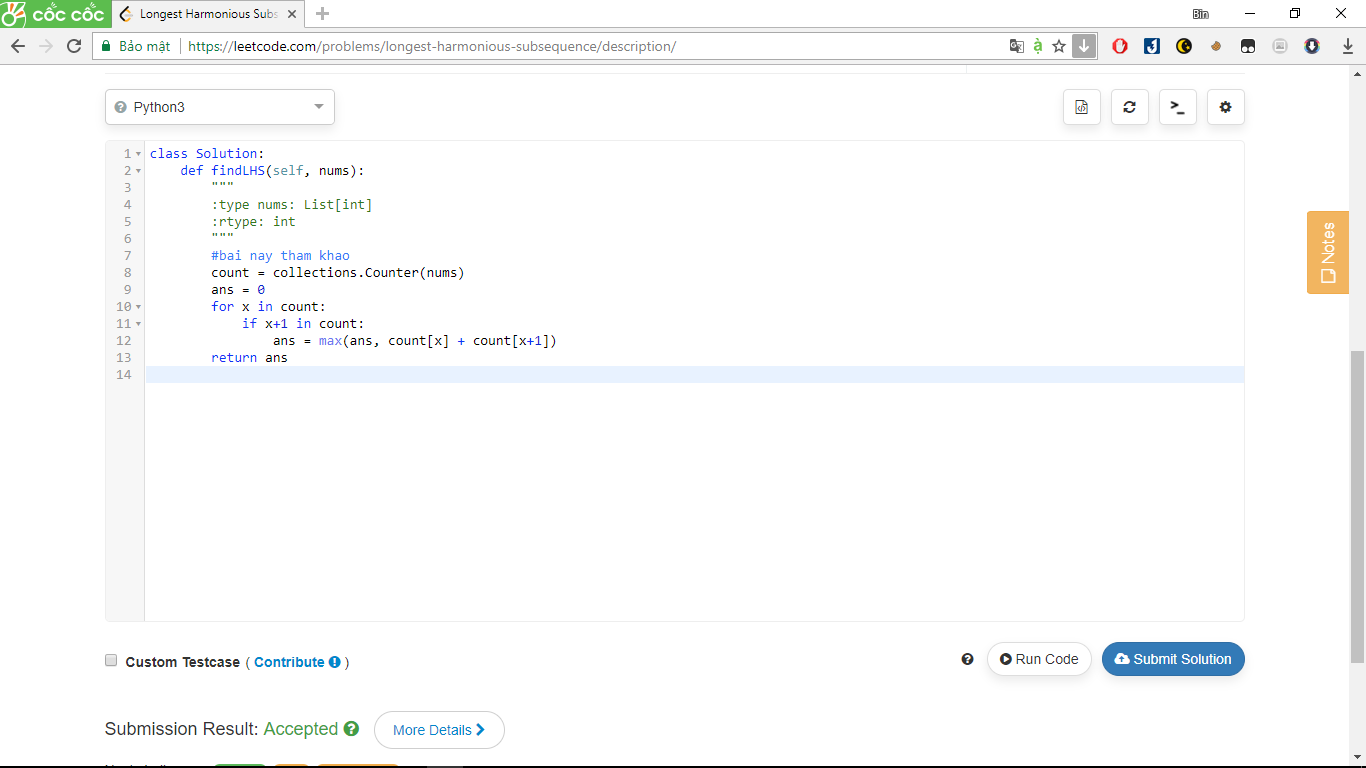
ans = 0

for x in count:

if x+1 in count:

ans = max(ans, count[x] + count[x+1])

return ans



720. Longest Word in Dictionary

Given a list of strings words representing an English Dictionary, find the longest word in words that can be built one character at a time by other words in words. If there is more than one possible answer, return the longest word with the smallest lexicographical order.

If there is no answer, return the empty string.

**Example 1:**

**Input:**

words = ["w","wo","wor","worl", "world"]

**Output:** "world"

**Explanation:**

The word "world" can be built one character at a time by "w", "wo", "wor", and "worl".

**Example 2:**

**Input:**

words = ["a", "banana", "app", "appl", "ap", "apply", "apple"]

**Output:** "apple"

**Explanation:**

Both "apply" and "apple" can be built from other words in the dictionary. However, "apple" is lexicographically smaller than "apply".

**Note:**

 All the strings in the input will only contain lowercase letters.

 The length of words will be in the range [1, 1000].

 The length of words[i] will be in the range [1, 30].

Answer:

class Solution:

def longestWord(self, words):

"""

:type words: List[str]

:rtype: str

"""

#bai nay tham khao

words.sort()

words\_set, longest\_word = set(['']), ''

for word in words:

if word[:-1] in words\_set:

words\_set.add(word)

if len(word) > len(longest\_word):

longest\_word = word

return longest\_word



674. Longest Continuous Increasing Subsequence

Given an unsorted array of integers, find the length of longest continuous increasing subsequence (subarray).

**Example 1:**

**Input:** [1,3,5,4,7]

**Output:** 3

**Explanation:** The longest continuous increasing subsequence is [1,3,5], its length is 3.

Even though [1,3,5,7] is also an increasing subsequence, it's not a continuous one where 5 and 7 are separated by 4.

**Example 2:**

**Input:** [2,2,2,2,2]

**Output:** 1

**Explanation:** The longest continuous increasing subsequence is [2], its length is 1.

**Note:** Length of the array will not exceed 10,000.

Answer:

class Solution:

def findLengthOfLCIS(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

if nums == []:return 0

if len(nums) <= 1:return 1

count = 1

max\_count = 1

for i in range(1,len(nums)):

if i and nums[i-1] < nums[i]:

count += 1

if count > max\_count:

max\_count = count

else: count = 1

return max\_count



747. Largest Number At Least Twice of Others

In a given integer array nums, there is always exactly one largest element.

Find whether the largest element in the array is at least twice as much as every other number in the array.

If it is, return the **index** of the largest element, otherwise return -1.

**Example 1:**

**Input:** nums = [3, 6, 1, 0]

**Output:** 1

**Explanation:** 6 is the largest integer, and for every other number in the array x,

6 is more than twice as big as x. The index of value 6 is 1, so we return 1.

**Example 2:**

**Input:** nums = [1, 2, 3, 4]

**Output:** -1

**Explanation:** 4 isn't at least as big as twice the value of 3, so we return -1.

**Note:**

1. nums will have a length in the range [1, 50].
2. Every nums[i] will be an integer in the range [0, 99].

Answer:

class Solution:

def dominantIndex(self, nums):

"""

:type nums: List[int]

:rtype: int

"""

nums2 = nums.copy()

nums2.sort()

if nums2 == None:

return -1

elif len(nums2) == 1:

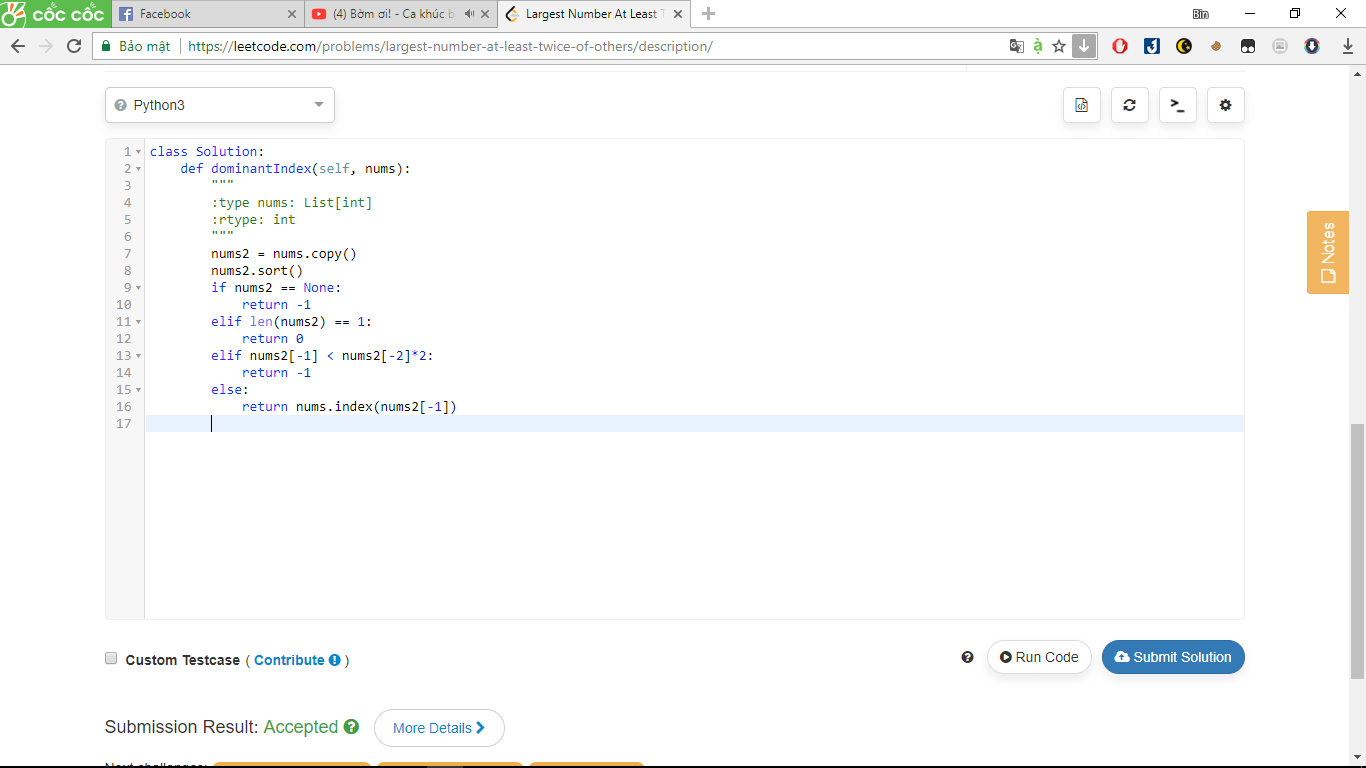
return 0

elif nums2[-1] < nums2[-2]\*2:

return -1

else:

return nums.index(nums2[-1])



455. Assign Cookies

Assume you are an awesome parent and want to give your children some cookies. But, you should give each child at most one cookie. Each child i has a greed factor gi, which is the minimum size of a cookie that the child will be content with; and each cookie j has a size sj. If sj >= gi, we can assign the cookie j to the child i, and the child i will be content. Your goal is to maximize the number of your content children and output the maximum number.

**Note:**  
You may assume the greed factor is always positive.   
You cannot assign more than one cookie to one child.

**Example 1:**

**Input:** [1,2,3], [1,1]

**Output:** 1

**Explanation:** You have 3 children and 2 cookies. The greed factors of 3 children are 1, 2, 3.

And even though you have 2 cookies, since their size is both 1, you could only make the child whose greed factor is 1 content.

You need to output 1.

**Example 2:**

**Input:** [1,2], [1,2,3]

**Output:** 2

**Explanation:** You have 2 children and 3 cookies. The greed factors of 2 children are 1, 2.

You have 3 cookies and their sizes are big enough to gratify all of the children,

You need to output 2.

Answer:

class Solution:

def findContentChildren(self, g, s):

"""

:type g: List[int]

:type s: List[int]

:rtype: int

"""

#bai nay tham khao

g.sort()

s.sort()

childi = 0

cookiei = 0

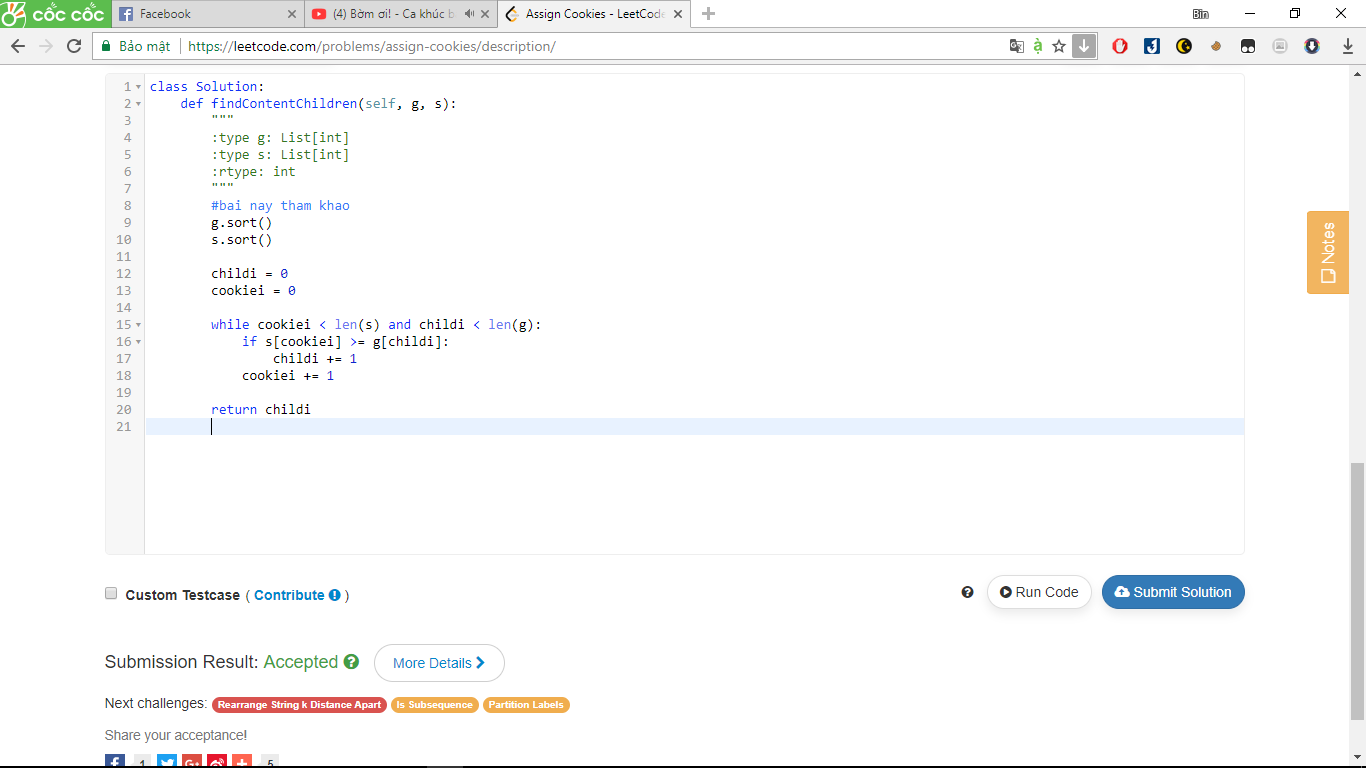
while cookiei < len(s) and childi < len(g):

if s[cookiei] >= g[childi]:

childi += 1

cookiei += 1

return childi



101. Symmetric Tree

Given a binary tree, check whether it is a mirror of itself (ie, symmetric around its center).

For example, this binary tree [1,2,2,3,4,4,3] is symmetric:

1

/ \

2 2

/ \ / \

3 4 4 3

But the following [1,2,2,null,3,null,3] is not:

1

/ \

2 2

\ \

3 3

**Note:**  
Bonus points if you could solve it both recursively and iteratively.

Answer:

# Definition for a binary tree node.

# class TreeNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.left = None

# self.right = None

class Solution:

def isSymmetric(self, root):

if not root:

return True

return self.dfs(root.left, root.right)

def dfs(self, l, r):

if l and r:

return l.val == r.val and self.dfs(l.left, r.right) and self.dfs(l.right, r.left)

return l == r



521. Longest Uncommon Subsequence I

Given a group of two strings, you need to find the longest uncommon subsequence of this group of two strings. The longest uncommon subsequence is defined as the longest subsequence of one of these strings and this subsequence should not be **any** subsequence of the other strings.

A **subsequence** is a sequence that can be derived from one sequence by deleting some characters without changing the order of the remaining elements. Trivially, any string is a subsequence of itself and an empty string is a subsequence of any string.

The input will be two strings, and the output needs to be the length of the longest uncommon subsequence. If the longest uncommon subsequence doesn't exist, return -1.

**Example 1:**

**Input:** "aba", "cdc"

**Output:** 3

**Explanation:** The longest uncommon subsequence is "aba" (or "cdc"),   
because "aba" is a subsequence of "aba",   
but not a subsequence of any other strings in the group of two strings.

**Note:**

1. Both strings' lengths will not exceed 100.
2. Only letters from a ~ z will appear in input strings.

Answer:

class Solution:

def findLUSlength(self, a, b):

"""

:type a: str

:type b: str

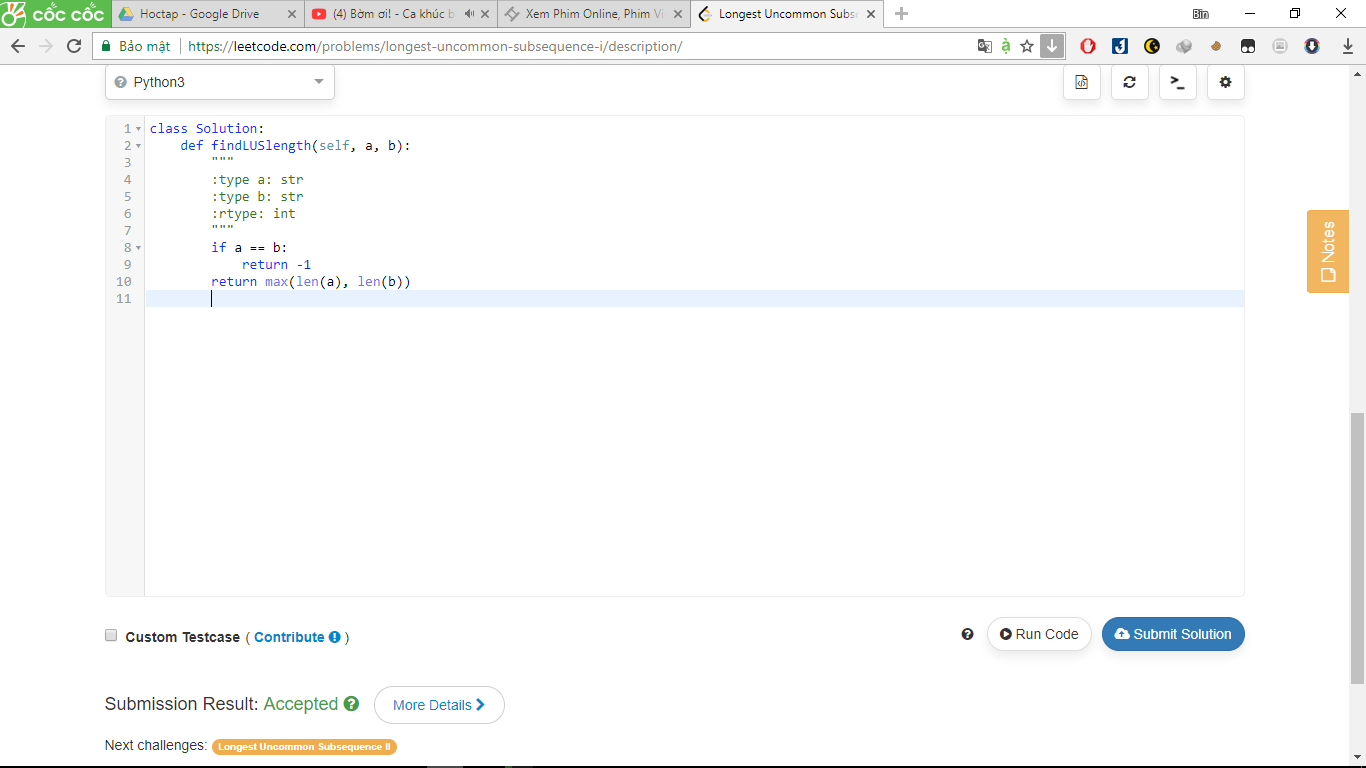
:rtype: int

"""

if a == b:

return -1

return max(len(a), len(b))



292. Nim Game

You are playing the following Nim Game with your friend: There is a heap of stones on the table, each time one of you take turns to remove 1 to 3 stones. The one who removes the last stone will be the winner. You will take the first turn to remove the stones.

Both of you are very clever and have optimal strategies for the game. Write a function to determine whether you can win the game given the number of stones in the heap.

For example, if there are 4 stones in the heap, then you will never win the game: no matter 1, 2, or 3 stones you remove, the last stone will always be removed by your friend.

Answer:

class Solution:

def canWinNim(self, n):

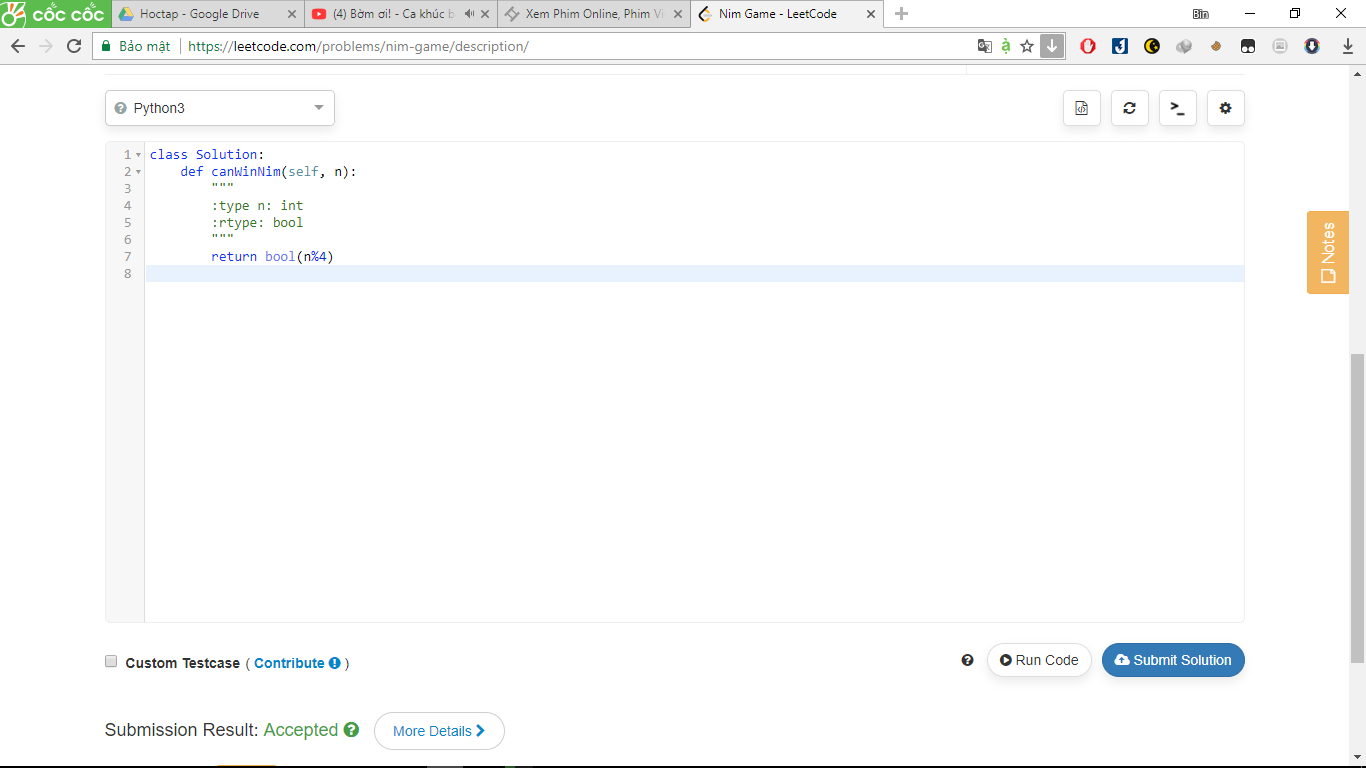
"""

:type n: int

:rtype: bool

"""

return bool(n%4)



226. Invert Binary Tree

Invert a binary tree.

4

/ \

2 7

/ \ / \

1 3 6 9

to

4

/ \

7 2

/ \ / \

9 6 3 1

**Answer:**

# Definition for a binary tree node.

# class TreeNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.left = None

# self.right = None

class Solution:

def invertTree(self, root):

"""

:type root: TreeNode

:rtype: TreeNode

"""

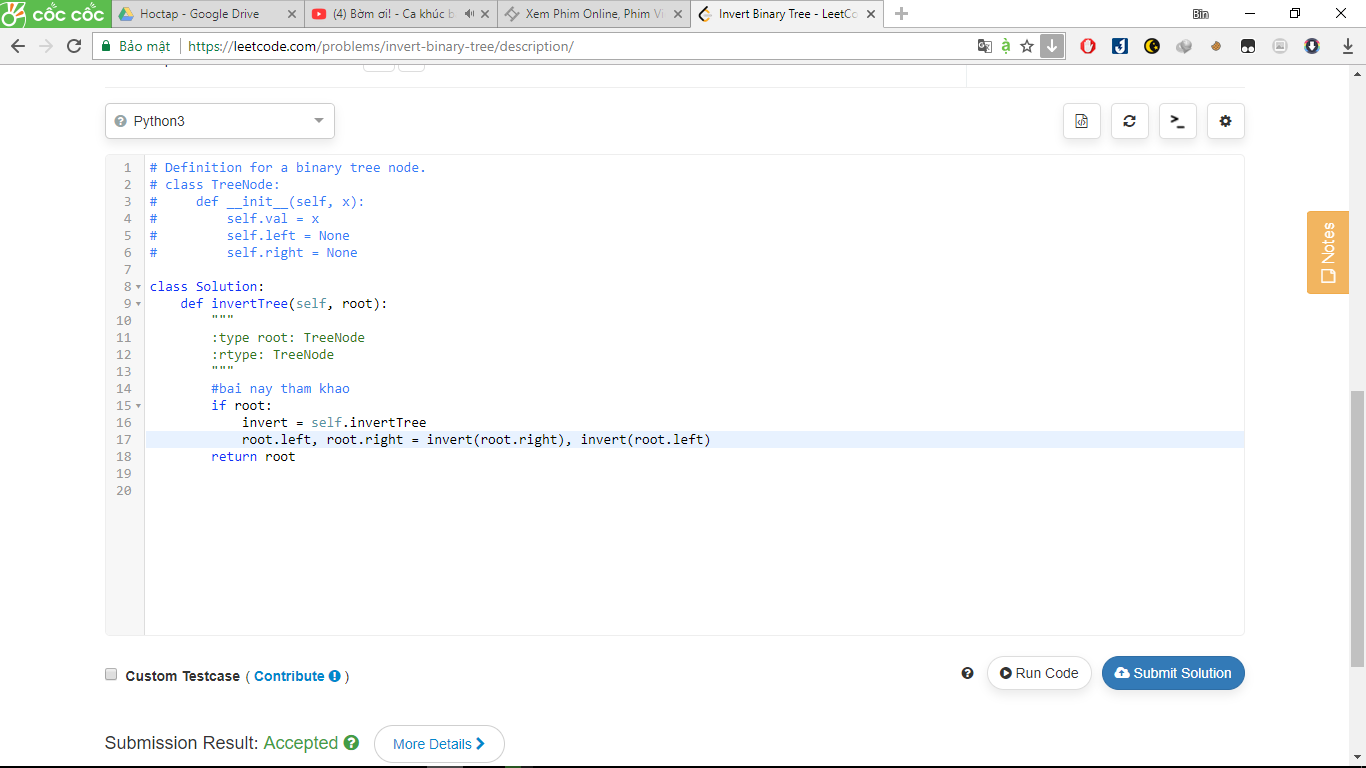
#bai nay tham khao

if root:

invert = self.invertTree

root.left, root.right = invert(root.right), invert(root.left)

return root



482. License Key Formatting

You are given a license key represented as a string S which consists only alphanumeric character and dashes. The string is separated into N+1 groups by N dashes.

Given a number K, we would want to reformat the strings such that each group contains *exactly* K characters, except for the first group which could be shorter than K, but still must contain at least one character. Furthermore, there must be a dash inserted between two groups and all lowercase letters should be converted to uppercase.

Given a non-empty string S and a number K, format the string according to the rules described above.

**Example 1:**

**Input:** S = "5F3Z-2e-9-w", K = 4

**Output:** "5F3Z-2E9W"

**Explanation:** The string S has been split into two parts, each part has 4 characters.

Note that the two extra dashes are not needed and can be removed.

**Example 2:**

**Input:** S = "2-5g-3-J", K = 2

**Output:** "2-5G-3J"

**Explanation:** The string S has been split into three parts, each part has 2 characters except the first part as it could be shorter as mentioned above.

**Note:**

1. The length of string S will not exceed 12,000, and K is a positive integer.
2. String S consists only of alphanumerical characters (a-z and/or A-Z and/or 0-9) and dashes(-).
3. String S is non-empty.

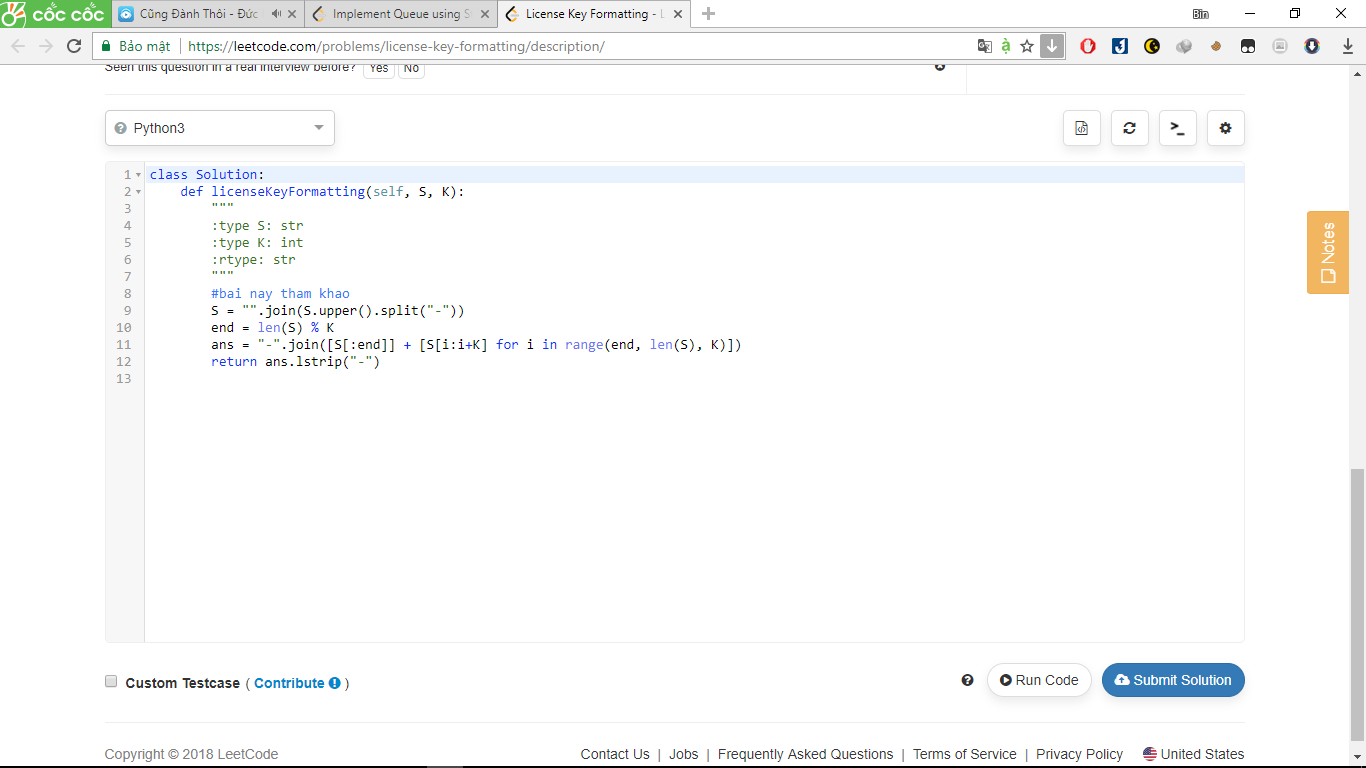
Answer:

S = "".join(S.upper().split("-"))

end = len(S) % K

ans = "-".join([S[:end]] + [S[i:i+K] **for** i **in** range(end, len(S), K)])

**return** ans.lstrip("-")



232. Implement Queue using Stacks

Implement the following operations of a queue using stacks.

* push(x) -- Push element x to the back of queue.
* pop() -- Removes the element from in front of queue.
* peek() -- Get the front element.
* empty() -- Return whether the queue is empty.

**Notes:**

* You must use *only* standard operations of a stack -- which means only push to top, peek/pop from top, size, and is emptyoperations are valid.
* Depending on your language, stack may not be supported natively. You may simulate a stack by using a list or deque (double-ended queue), as long as you use only standard operations of a stack.
* You may assume that all operations are valid (for example, no pop or peek operations will be called on an empty queue).

Answer:

class MyQueue:

def \_\_init\_\_(self):

"""

Initialize your data structure here.

"""

self.queue=[]

def push(self, x):

"""

:type x: int

:rtype: nothing

"""

self.queue.insert(0,x)

def pop(self):

"""

:rtype: nothing

"""

return self.queue.pop()

def peek(self):

"""

:rtype: int

"""

return self.queue[-1]

def empty(self):

"""

:rtype: bool

"""

return len(self.queue) == 0

# Your MyQueue object will be instantiated and called as such:

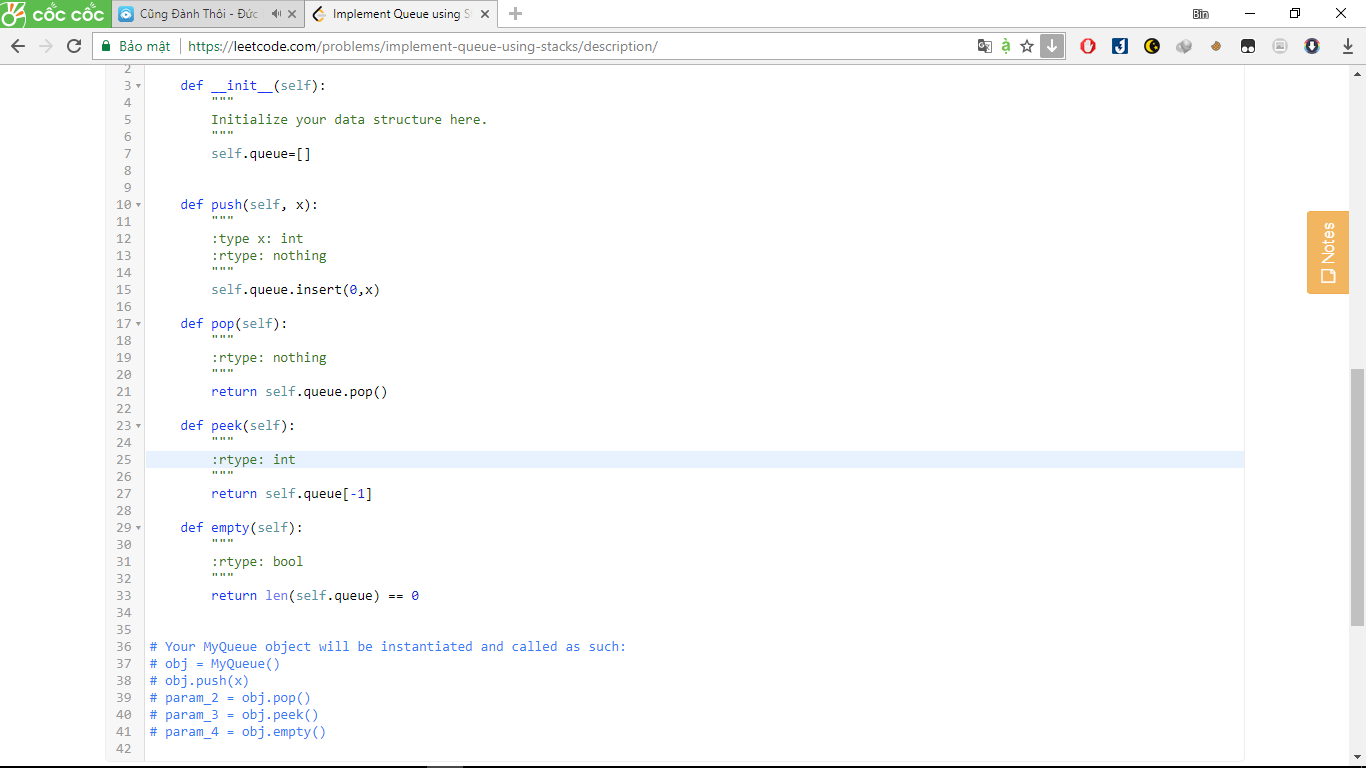
# obj = MyQueue()

# obj.push(x)

# param\_2 = obj.pop()

# param\_3 = obj.peek()

# param\_4 = obj.empty()



38. Count and Say

he count-and-say sequence is the sequence of integers with the first five terms as following:

1. 1

2. 11

3. 21

4. 1211

5. 111221

1 is read off as "one 1" or 11.  
11 is read off as "two 1s" or 21.  
21 is read off as "one 2, then one 1" or 1211.

Given an integer *n*, generate the *n*th term of the count-and-say sequence.

Note: Each term of the sequence of integers will be represented as a string.

**Example 1:**

**Input:** 1

**Output:** "1"

**Example 2:**

**Input:** 4

**Output:** "1211"

Answer:

class Solution:

def countAndSay(self, n):

"""

:type n: int

:rtype: str

"""

#bai nay tham khao

if n ==1:

return "1"

a=self.countAndSay(n-1)

s=""

num = a[0]

c=1

for i in range(1,len(a)):

if num != a[i]:

s = s+(str(c) + num)

num=a[i]

c=1

else:

c+=1

s = s + (str(c) + num)

return s



463. Island Perimeter

You are given a map in form of a two-dimensional integer grid where 1 represents land and 0 represents water. Grid cells are connected horizontally/vertically (not diagonally). The grid is completely surrounded by water, and there is exactly one island (i.e., one or more connected land cells). The island doesn't have "lakes" (water inside that isn't connected to the water around the island). One cell is a square with side length 1. The grid is rectangular, width and height don't exceed 100. Determine the perimeter of the island.

**Example:**

[[0,1,0,0],

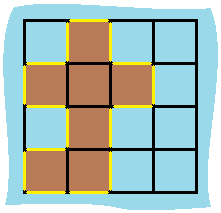
[1,1,1,0],

[0,1,0,0],

[1,1,0,0]]

Answer: 16

Explanation: The perimeter is the 16 yellow stripes in the image below:



Answer:

class Solution:

def islandPerimeter(self, grid):

"""

:type grid: List[List[int]]

:rtype: int

"""

#bai nay tham khao

total = 0

for tot in grid:

total += sum(tot)

row = len(grid)

col = len(grid[0])

dup = 0

for i in range(row):

for j in range(col):

if i+1 <row:

if grid[i][j]+grid[i+1][j]==2:

dup += 1

if j+1 < col:

if grid[i][j]+grid[i][j+1]==2:

dup += 1

return total\*4-dup\*2



746. Min Cost Climbing Stairs

On a staircase, the i-th step has some non-negative cost cost[i] assigned (0 indexed).

Once you pay the cost, you can either climb one or two steps. You need to find minimum cost to reach the top of the floor, and you can either start from the step with index 0, or the step with index 1.

**Example 1:**

**Input:** cost = [10, 15, 20]

**Output:** 15

**Explanation:** Cheapest is start on cost[1], pay that cost and go to the top.

**Example 2:**

**Input:** cost = [1, 100, 1, 1, 1, 100, 1, 1, 100, 1]

**Output:** 6

**Explanation:** Cheapest is start on cost[0], and only step on 1s, skipping cost[3].

**Note:**

Answer:

class Solution:

def minCostClimbingStairs(self, cost):

"""

:type cost: List[int]

:rtype: int

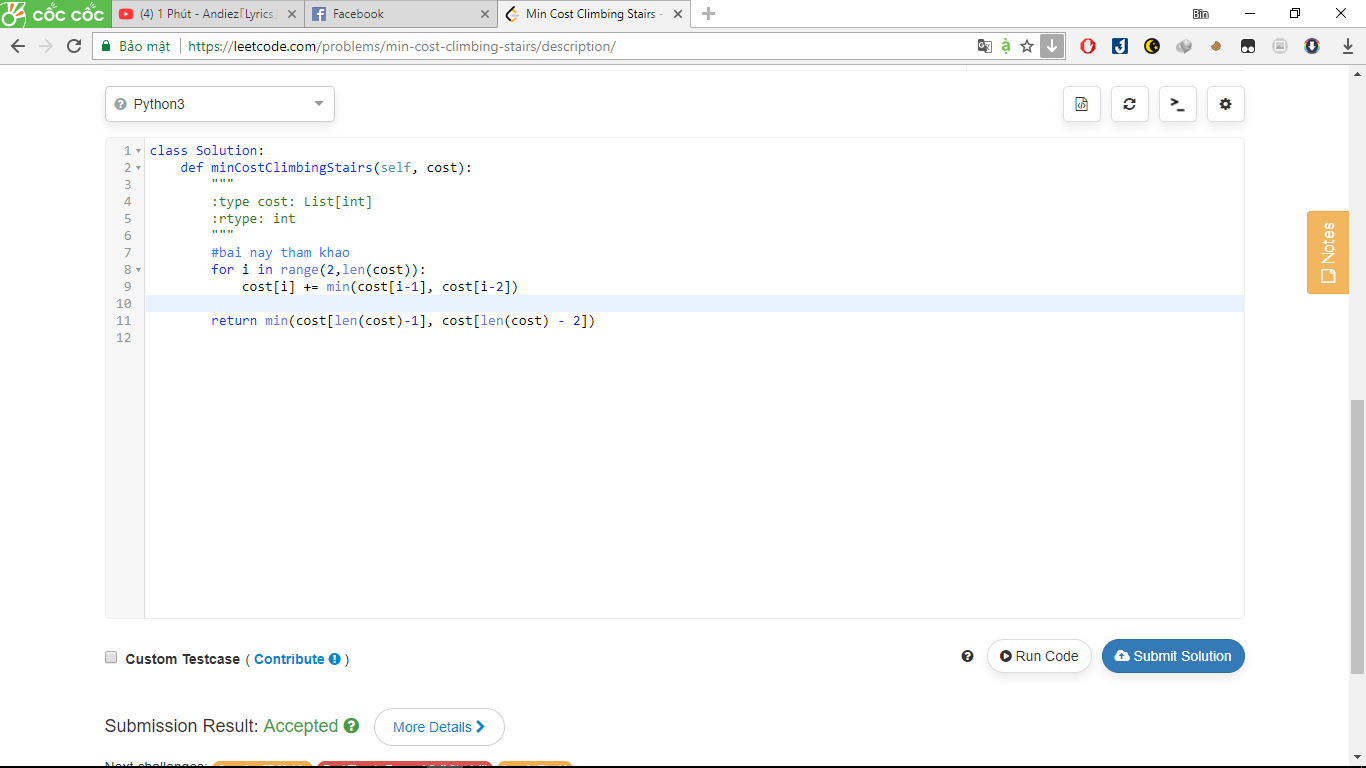
"""

#bai nay tham khao

for i in range(2,len(cost)):

cost[i] += min(cost[i-1], cost[i-2])

return min(cost[len(cost)-1], cost[len(cost) - 2])



112. Path Sum

Given a binary tree and a sum, determine if the tree has a root-to-leaf path such that adding up all the values along the path equals the given sum.

For example:  
Given the below binary tree and sum = 22,

5

/ \

4 8

/ / \

11 13 4

/ \ \

7 2 1

return true, as there exist a root-to-leaf path 5->4->11->2 which sum is 22.

Answer:

# Definition for a binary tree node.

# class TreeNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.left = None

# self.right = None

class Solution:

def hasPathSum(self, root, sum):

"""

:type root: TreeNode

:type sum: int

:rtype: bool

"""

#bai nay tham khao

if root==None:

return False

if root.left==None and root.right==None and root.val==sum:

return True

if root.left!=None:

if self.hasPathSum(root.left,sum-root.val)==True:

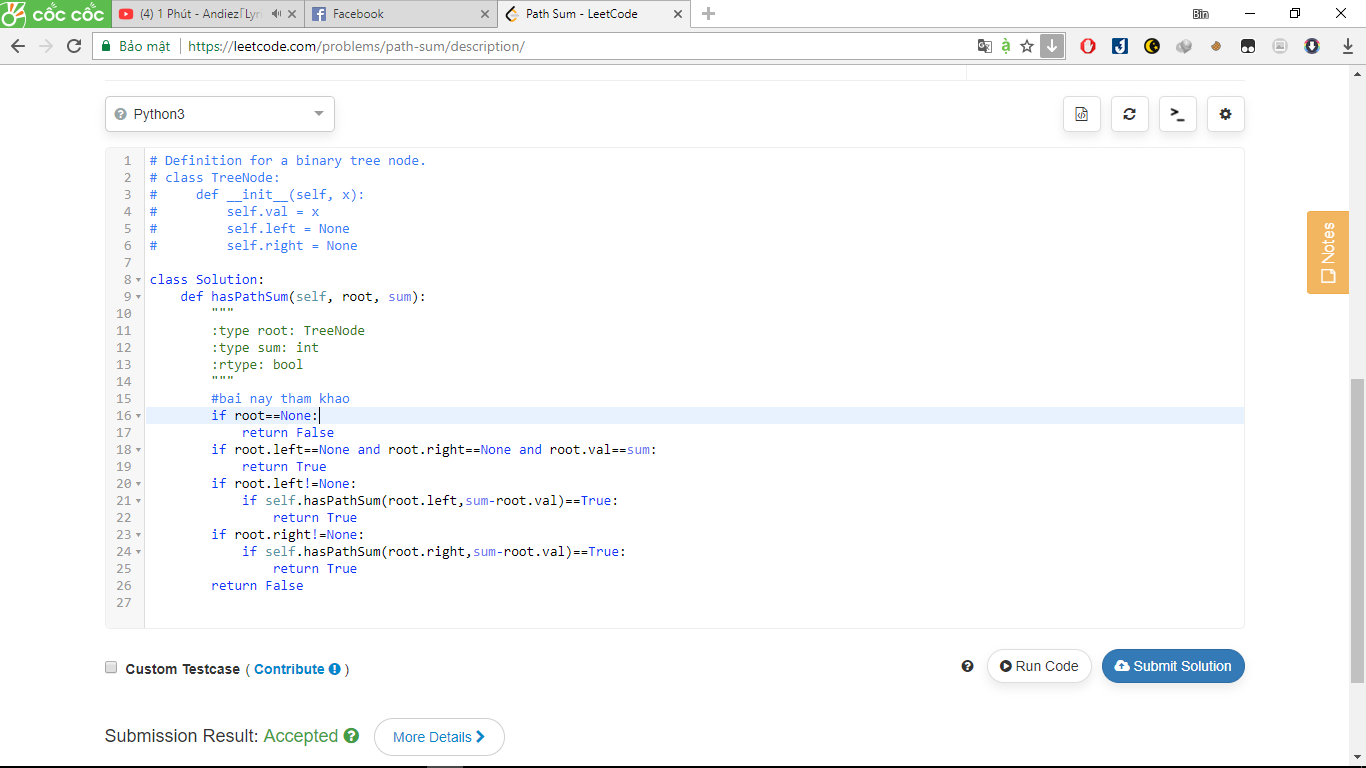
return True

if root.right!=None:

if self.hasPathSum(root.right,sum-root.val)==True:

return True

return False



121. Best Time to Buy and Sell Stock

Say you have an array for which the *i*th element is the price of a given stock on day *i*.

If you were only permitted to complete at most one transaction (ie, buy one and sell one share of the stock), design an algorithm to find the maximum profit.

**Example 1:**

Input: [7, 1, 5, 3, 6, 4]

Output: 5

max. difference = 6-1 = 5 (not 7-1 = 6, as selling price needs to be larger than buying price)

**Example 2:**

Input: [7, 6, 4, 3, 1]

Output: 0

In this case, no transaction is done, i.e. max profit = 0.

Answer:

class Solution:

def maxProfit(self, prices):

"""

:type prices: List[int]

:rtype: int

"""

if len(prices) < 2: return 0

buy = prices[0]

sale = 0

profit = 0

for i in range(len(prices)):

if prices[i] < buy: #if item lower than buy, set as buy

buy = prices[i]

elif prices[i] > buy:

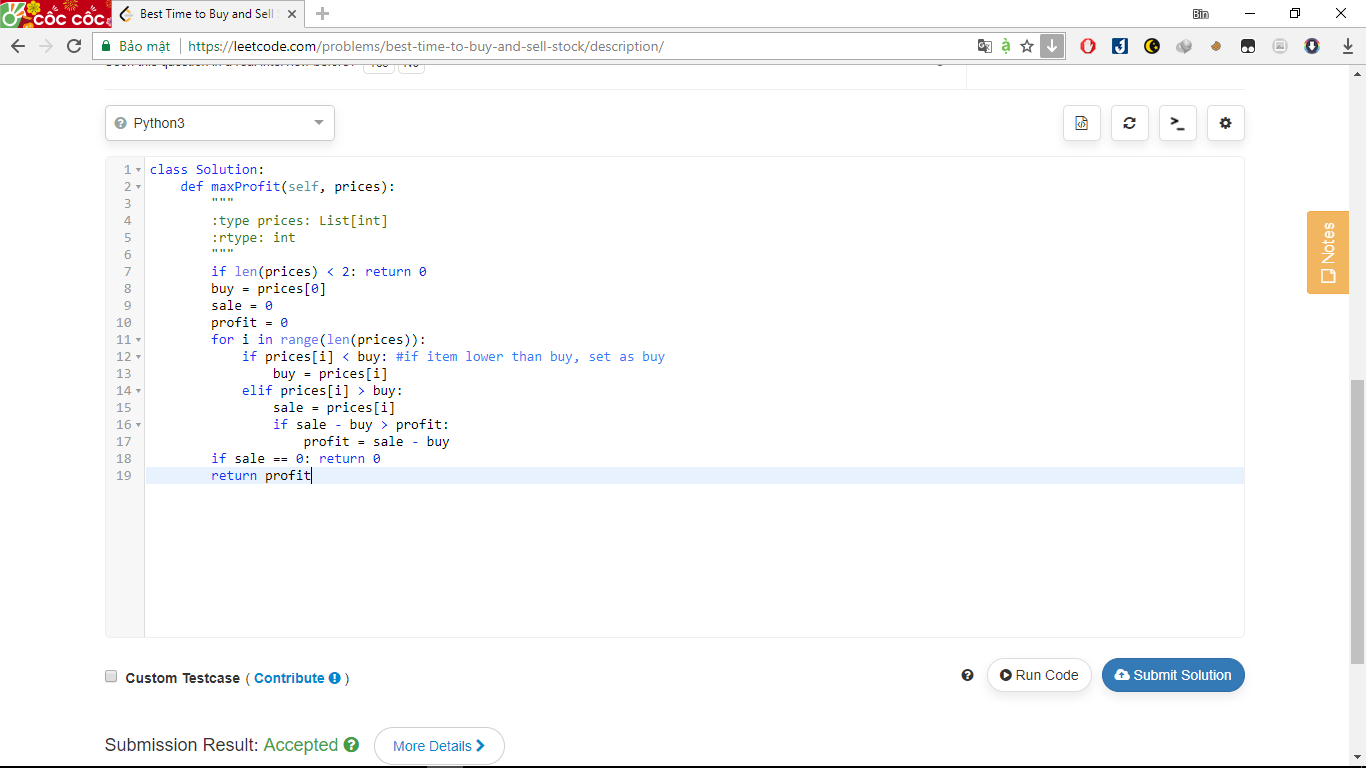
sale = prices[i]

if sale - buy > profit:

profit = sale - buy

if sale == 0: return 0

return profit



690. Employee Importance

You are given a data structure of employee information, which includes the employee's **unique id**, his **importance value** and his **direct**subordinates' id.

For example, employee 1 is the leader of employee 2, and employee 2 is the leader of employee 3. They have importance value 15, 10 and 5, respectively. Then employee 1 has a data structure like [1, 15, [2]], and employee 2 has [2, 10, [3]], and employee 3 has [3, 5, []]. Note that although employee 3 is also a subordinate of employee 1, the relationship is **not direct**.

Now given the employee information of a company, and an employee id, you need to return the total importance value of this employee and all his subordinates.

**Example 1:**

**Input:** [[1, 5, [2, 3]], [2, 3, []], [3, 3, []]], 1

**Output:** 11

**Explanation:**

Employee 1 has importance value 5, and he has two direct subordinates: employee 2 and employee 3. They both have importance valu

Answer:

"""

# Employee info

class Employee:

def \_\_init\_\_(self, id, importance, subordinates):

# It's the unique id of each node.

# unique id of this employee

self.id = id

# the importance value of this employee

self.importance = importance

# the id of direct subordinates

self.subordinates = subordinates

"""

class Solution:

def getImportance(self, employees, id):

"""

:type employees: Employee

:type id: int

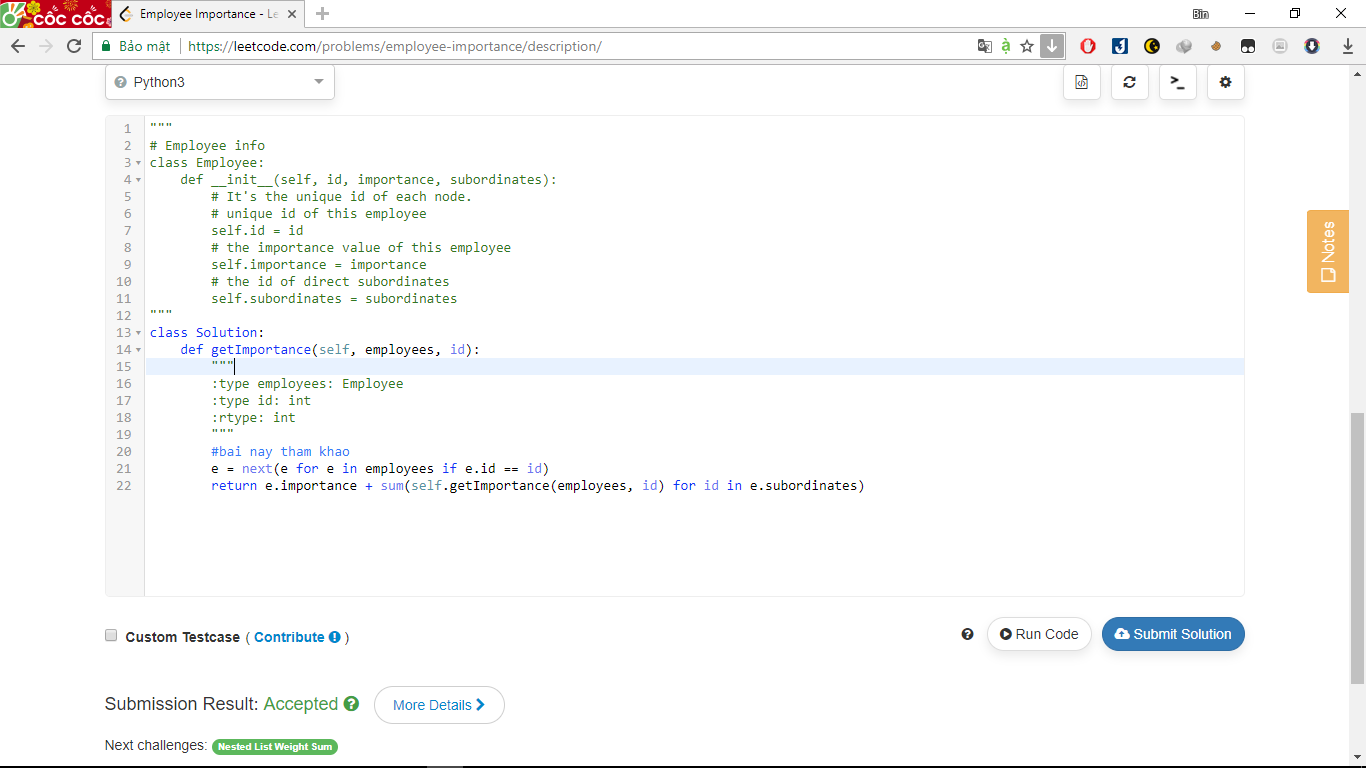
:rtype: int

"""

#bai nay tham khao

e = next(e for e in employees if e.id == id)

return e.importance + sum(self.getImportance(employees, id) for id in e.subordinates)



447. Number of Boomerangs

Given *n* points in the plane that are all pairwise distinct, a "boomerang" is a tuple of points (i, j, k) such that the distance between iand j equals the distance between i and k (**the order of the tuple matters**).

Find the number of boomerangs. You may assume that *n* will be at most **500** and coordinates of points are all in the range **[-10000, 10000]** (inclusive).

**Example:**

**Input:**

[[0,0],[1,0],[2,0]]

**Output:**

2

**Explanation:**

The two boomerangs are **[[1,0],[0,0],[2,0]]** and **[[1,0],[2,0],[0,0]]**

Answee:

class Solution:

def numberOfBoomerangs(self, points):

"""

:type points: List[List[int]]

:rtype: int

"""

#bai nay tham khao

nums = 0

for x1, y1 in points:

distance = collections.defaultdict(int)

for x2, y2 in points:

dx = abs(x2 - x1)

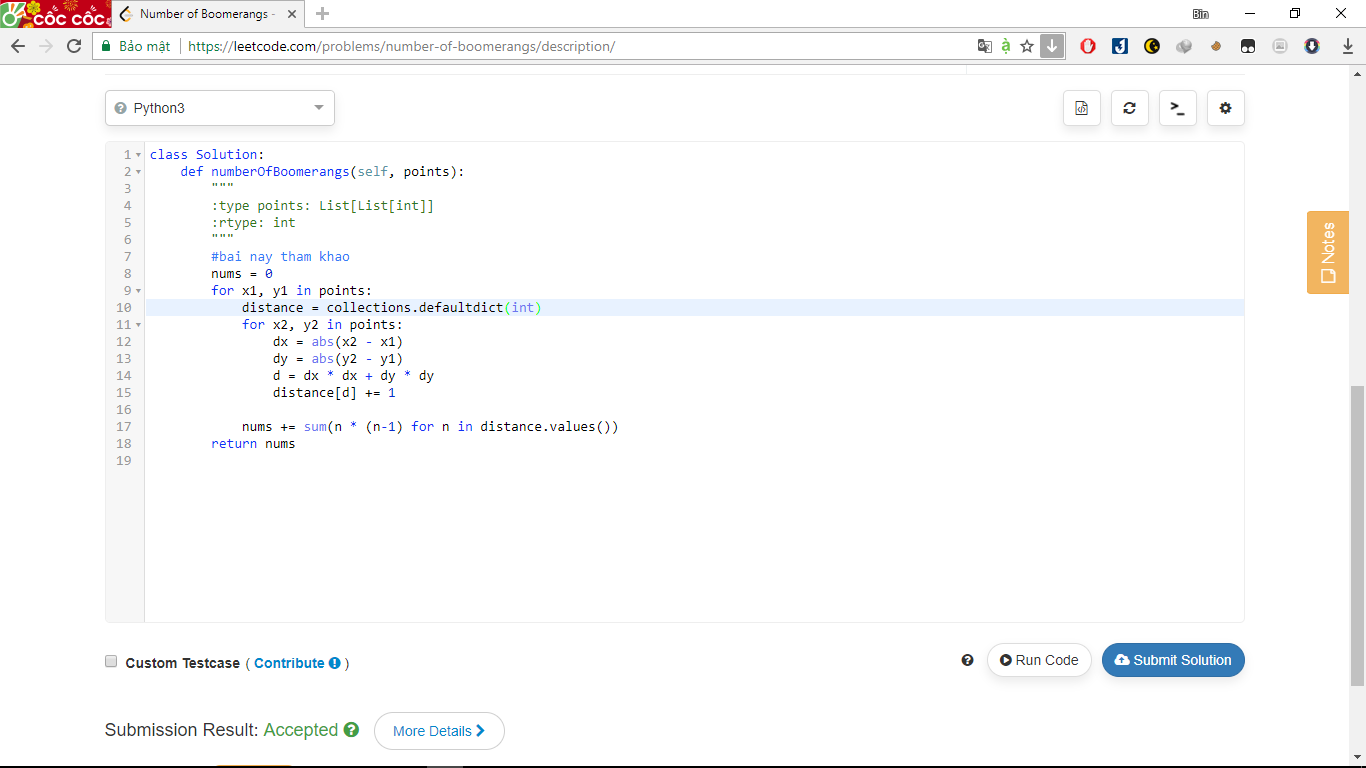
dy = abs(y2 - y1)

d = dx \* dx + dy \* dy

distance[d] += 1

nums += sum(n \* (n-1) for n in distance.values())

return nums



404. Sum of Left Leaves

Find the sum of all left leaves in a given binary tree.

**Example:**

3

/ \

9 20

/ \

15 7

There are two left leaves in the binary tree, with values **9** and **15** respectively. Return **24**.

Answer:

# Definition for a binary tree node.

# class TreeNode:

# def \_\_init\_\_(self, x):

# self.val = x

# self.left = None

# self.right = None

class Solution:

def sumOfLeftLeaves(self, root):

"""

:type root: TreeNode

:rtype: int

"""

#bai nay tham khao

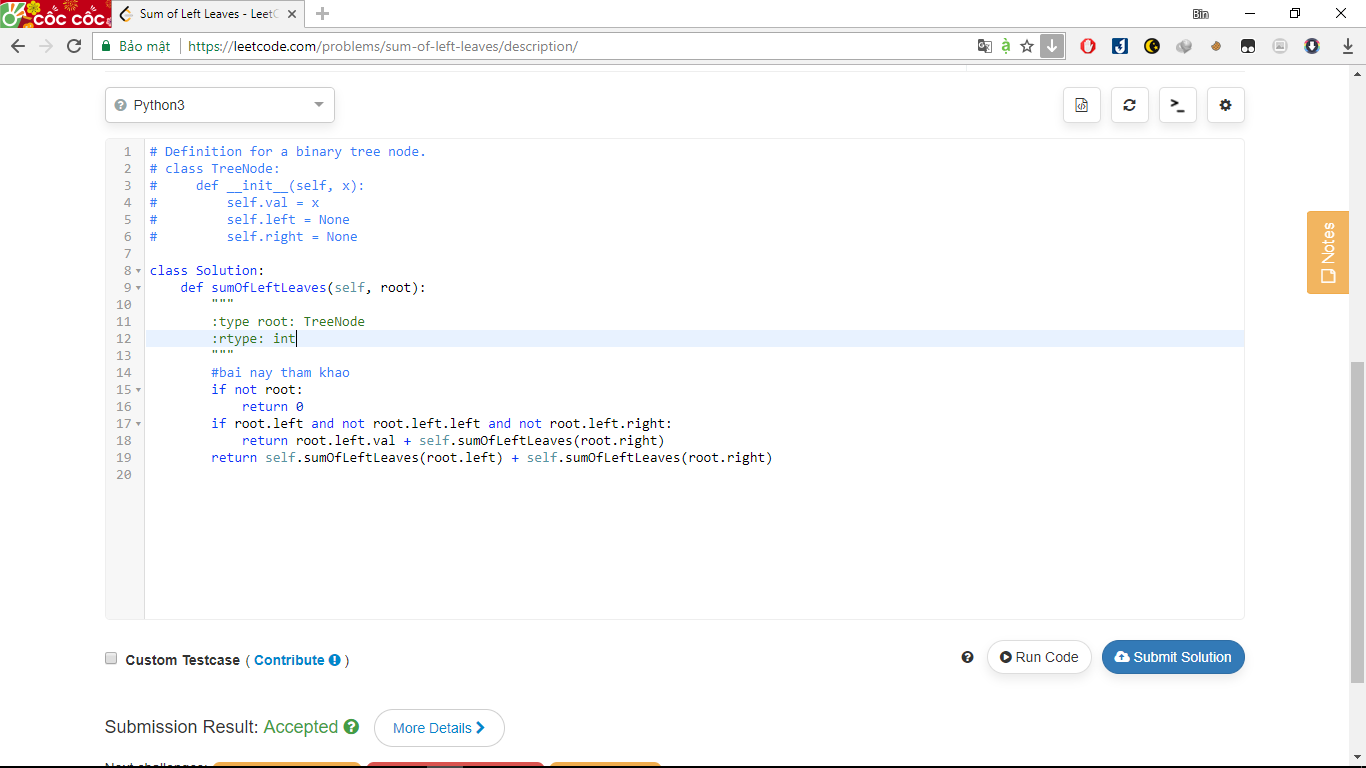
if not root:

return 0

if root.left and not root.left.left and not root.left.right:

return root.left.val + self.sumOfLeftLeaves(root.right)

return self.sumOfLeftLeaves(root.left) + self.sumOfLeftLeaves(root.right)



606. Construct String from Binary Tree

You need to construct a string consists of parenthesis and integers from a binary tree with the preorder traversing way.

The null node needs to be represented by empty parenthesis pair "()". And you need to omit all the empty parenthesis pairs that don't affect the one-to-one mapping relationship between the string and the original binary tree.

**Example 1:**

**Input:** Binary tree: [1,2,3,4]

1

/ \

2 3

/

4

**Output:** "1(2(4))(3)"

**Explanation:** Originallay it needs to be "1(2(4)())(3()())",   
but you need to omit all the unnecessary empty parenthesis pairs.   
And it will be "1(2(4))(3)".

**Example 2:**

**Input:** Binary tree: [1,2,3,null,4]

1

/ \

2 3

\

4

**Output:** "1(2()(4))(3)"

**Explanation:** Almost the same as the first example,   
except we can't omit the first parenthesis pair to break the one-to-one mapping relationship between the input and the output.

Answer:

class Solution:

def tree2str(self, t):

"""

:type t: TreeNode

:rtype: str

"""

if not t:

return ""

res = ""

left = self.tree2str(t.left)

right = self.tree2str(t.right)

if left or right:

res += "(%s)" % left

if right:

res += "(%s)" % right

return str(t.val) + res

