Assignment-7

06 June 2023 00:17

Question 1

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

Two strings s and t 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.

Example 1:

Input: s = "egg", t = "add"

Output: true

```
class Solution(object):
    def isIsomorphic(self, s, t):
        if len(s)!=len(t):
            return False
        mp={}
        i=0
        for i in range(len(s)):
            if s[i] in mp and mp[s[i]]!=t[i]:
                return False
            elif t[i] in mp.values():
                for k,v in mp.items():
                    if v==t[i] and k!=s[i]:
                        return False
            else:
                mp[s[i]]=t[i]

        return True
```

Question 3

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

You must solve the problem without using any built-in library for handling large integers (such as BigInteger. You must also not convert the inputs to integers directly.

Example 1:

Input: num1 = "11", num2 = "123"

Output:

"134"

```
class Solution(object):
    def addStrings(self, num1, num2):
        n1,n2=len(num1),len(num2)
        s=t=0
        for i in range(n1):
            asci=ord(num1[i])-48
            a=s*10
            s=a+asci
        for i in range(n2):
            asci=ord(num2[i])-48
            a=t*10
            t=a+asci
        return str(s+t)
```

Question 4

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

Example 1:

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

Output: "s'teL ekat edoCteeL tsetnoc"

```
class Solution(object):
    def reverseWords(self, s):
        lt=s.split(" ")
        return " ".join([item[::-1] for item in lt])
```

Question 6

Given two strings s and goal, return true *if and only if* s *can become* goal *after some number of* ***shifts*** *on* s.

A **shift** on s consists of moving the leftmost character of s to the rightmost position.

- For example, if s = "abcde", then it will be "bcdea" after one shift.

Example 1:

Input: s = "abcde", goal = "cdeab"

Output:

true

```
class Solution(object):
    def rotateString(self, s, goal):
        if len(s)!=len(goal):
            return False
        str_con=goal+goal
        if s in str_con:
            return True
        return False
```

Question 7

Given two strings s and t, return true *if they are equal when both are typed into empty text editors*. '#' means a backspace character.

Note that after backspacing an empty text, the text will continue empty.

Example 1:

Input: s = "ab#c", t = "ad#c"

Output: true

Explanation:

Both s and t become "ac".

```
class Solution(object):
    def backspaceCompare(self, s, t):
        st1=[]
        st2=[]
        res1=""
        res2=""
        for char in s:
            if len(st1)>0 and char =='#':
                st1.pop()
            elif char>='a' and char<='z':
                st1.append(char)

        for char in t:
            if len(st2)>0 and char =='#':
                st2.pop()
            elif char>='a' and char<='z':
                st2.append(char);

        while len(st1)>0:
            res1=res1+st1.pop()

        while len(st2)>0:
            res2=res2+st2.pop()

        if res1==res2:
            return True
        else:
            return False
```

Question 8

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

Example 1:

Input: coordinates = [[1,2],[2,3],[3,4],[4,5],[5,6],[6,7]]

Output: true

```
class Solution(object):
    def checkStraightLine(self, coordinates):
        y_dif=coordinates[1][1]-coordinates[0][1]
        x_dif=coordinates[1][0]-coordinates[0][0]

        for i in range(2,len(coordinates)):
            curr_y=coordinates[i][1]-coordinates[i-1][1]
            curr_x=coordinates[i][0]-coordinates[i-1][0]
            if y_dif*curr_x != x_dif*curr_y:
                return False
        return True
```

Question 5

Given a string s and an integer k, reverse the first k characters for every 2k characters counting from the start of the string.

If there are fewer 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 leave the other as original.

Example 1:

Input: s = "abcdefg", k = 2

Output:

"bacdfeg"

```
class Solution(object):
    def reverseStr(self, s, k):
        for i in range(0,len(s),2*k):
            if(i+k<len(s)):
                s=s[0:i]+s[i:i+k][::-1]+s[i+k:]
            else:
                s=s[0:i]+s[i:i+k][::-1]
        return s
```

Question 2

Given a string num which represents an integer, return true *if* num *is a **strobogrammatic number***.

A **strobogrammatic number** is a number that looks the same when rotated 180 degrees (looked at upside down).

Example 1:

Input: num = "69"

Output:

true

```
class Solution(object):
    def isStrobogrammatic(self, arr):
        strobo={'0':'0',
                '6':'9',
                '9':'6',
                '8':'8',
                '1':'1'

         if len(arr)==1:
             return arr in ['0','8','1']
         else:
             i=0
             j=len(arr)-1

             while(i<=j):
                 if arr[i] in strobo and strobo[arr[i]]=arr[j]:
                     i+=1
                     j-=1
                 else:
                     return False
             return True
           }
```