```
希 剑指offer > 二维数组中的查找 ▼
                                                             □相似的企业真题
                                                                               下—颗>
                                                                                  Python(2.7.3) > ● 核心代码模式
 时间限制: C/C++ 1秒, 其他语言2秒 空间限制: C/C++ 64M, 其他语言128M
 热度指数: 2044457
 本颢知识点: 数组
 ■算法知识视频讲解
                                                                                     def Find(self, target, array):
题目描述
在一个二维数组中(每个一维数组的长度相同),每一行都按照从左到右递增的顺序排序,每一列都按照
                                                                                        if array == []:
从上到下递增的顺序排序。请完成一个函数,输入这样的一个二维数组和一个整数,判断数组中是否含有
                                                                                         j = len(array[0])-1
[1,2,8,9],
                                                                                         while i < len(array) and j >=0:
    if array[i][j] == target:
[2,4,9,12],
[4,7,10,13],
                                                                                            'elif 'array[i][j] '< target:</pre>
[6,8,11,15]
s
给定 target = 7,返回 true。
给定 target = 3,返回 false。
```

骨 剑指offer > 替换空格 ▼

时间限制:C/C++1秒,其他语言2秒 空间限制:C/C++256M,其他语言512M 热度指数:33943 本题知识点: 字符串

■ 算法知识规频并解

题目描述

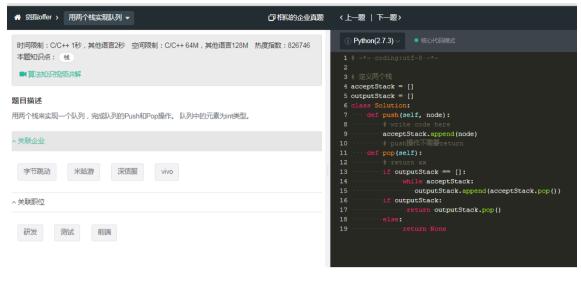
请实现一个函数,将一个字符串中的每个空格替换成"%20"。例如,当字符串为We Are Happy则经过替换之后的字符串为We%20Are%20Happy。

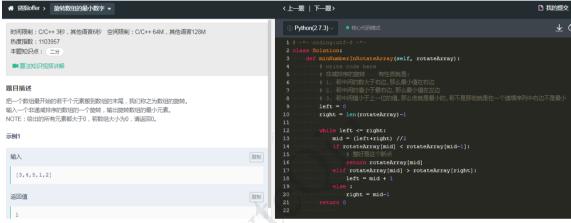
示例1



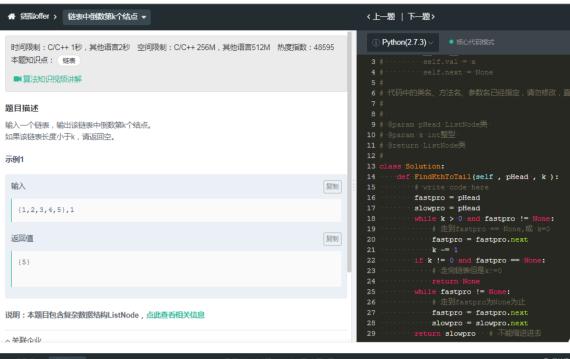
〈上—题 | 下—题>

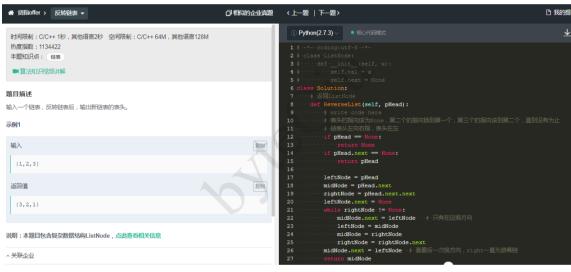
说明:本题目包含复杂数据结构ListNode,点此查看相关信息

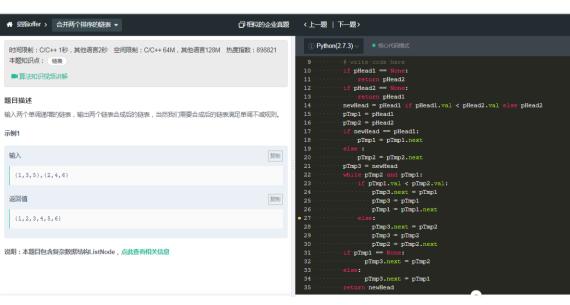








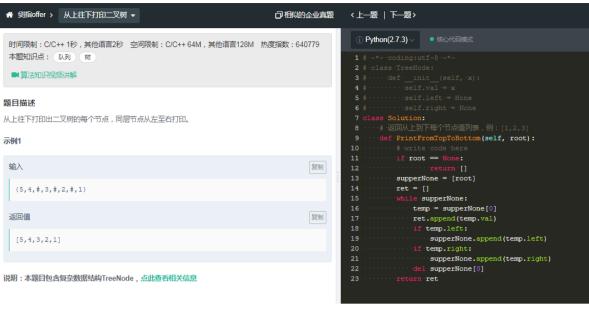






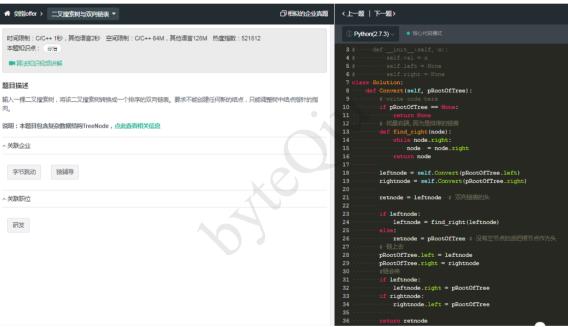


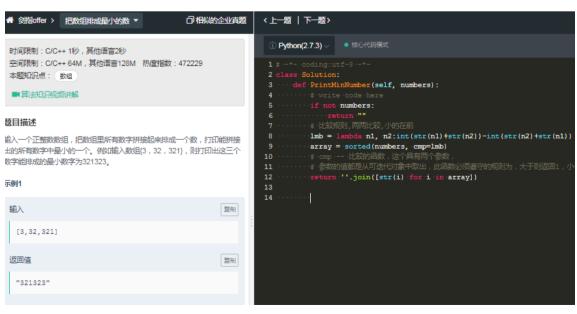


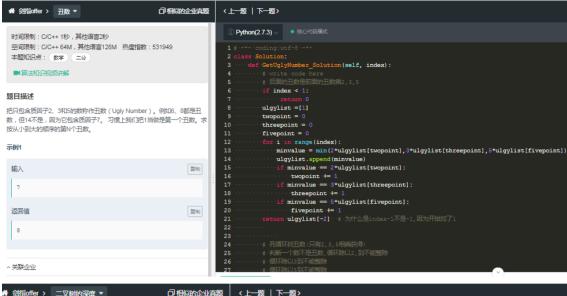


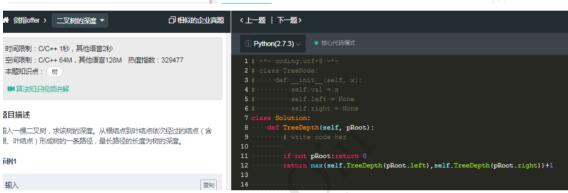


```
希 剑指offer > 二叉树中和为某一值的路径 ▼
                                                           口相似的企业真题
时间限制: C/C++ 1秒, 其他语言2秒 空间限制: C/C++ 64M, 其他语言128M 热度指数: 702700
本题知识点: 材
 ■算法知识视频讲解
野日描述
输入一颗二叉树的根节点和一个整数,按字典序打印出二叉树中结点值的和为输入整数的所有路径。路径定义为从树的根结点开
始往下一直到叶结点所经过的结点形成一条路径。
示例1
{10,5,12,4,7},22
返回值
                                                                  复制
[[10,5,7],[10,12]]
<del>∓</del>/5/12
输入
{10,5,12,4,7},15
返回值
                                                                  复制
```









```
● 剑指offer > 和为S的连续正数序列 ▼ □相似的企业有额时间限制:C/C++1秒,其他语言2秒空间限制:C/C++64M,其他语言128M 热度指数:462455本超知识点: 穷零
```

题目描述

小明很喜欢数学,有一天他在做数学作业时,要求计算出9~16的和,他马上就写出了正确答案是100。但是他并不满足于此,他在根究竟有多少种连续的正数序列的和为100(至少包括两个数)。没多久,他就得到另一组连续正数和为100的序列:18,19,20,21,22。现在把问题交给你,你能不能也很快的找出所有和为S的连续正数序列? Good Luck!

返回值描述:

输出所有和为s的连续正数序列。序列内按照从小至大的顺序,序列间 按照开始数字从小到大的顺序





题目描述

牛客最近来了一个新员工Fish,每天早晨总是会拿着一本英文杂志,写些句 子在本子上。同事Cat对Fish写的内容颇感兴趣,有一天他向Fish借来翻看, 但却读不懂它的意思。例如,"student. a am l"。后来才意识到,这家伙原来 把句子单词的顺序翻转了,正确的句子应该是"I am a student."。Cat对一 的翻转这些单词顺序可不在行,你能帮助他么? 🔺

示例1

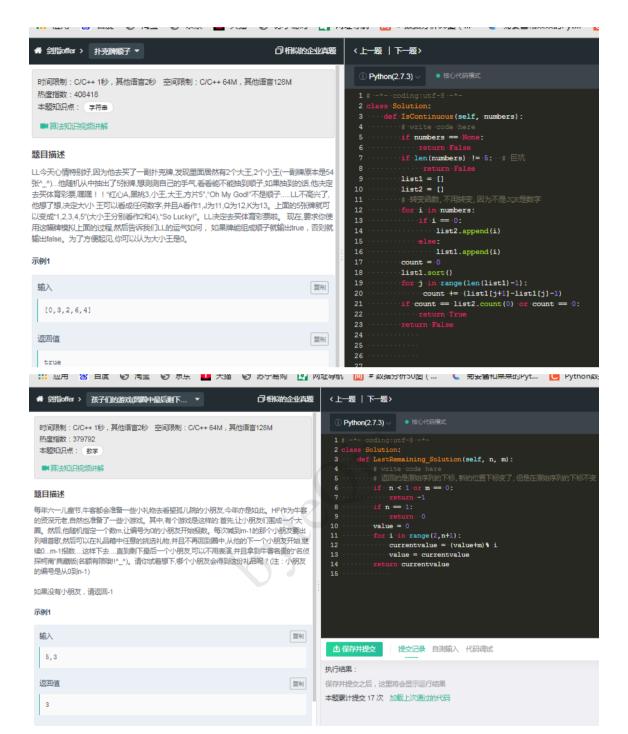


```
〈上—题 | 下—题〉
  ① Python(2.7.3) ~ 核心代码模式
  1 # --*- coding:utf-8 --*-
2 class Solution:
3 · · · ·def ReverseSentence(self, s):
              -s = -s.split()
              ····return a
·string = ''
           or for i in s[::-1]:

or i = '''+i

or string += i

return string.lstrip()
 12 -
 山 保存并提交
                     提交记录 自测输入 代码调试
保存并提交之后,这里将会显示运行结果
```





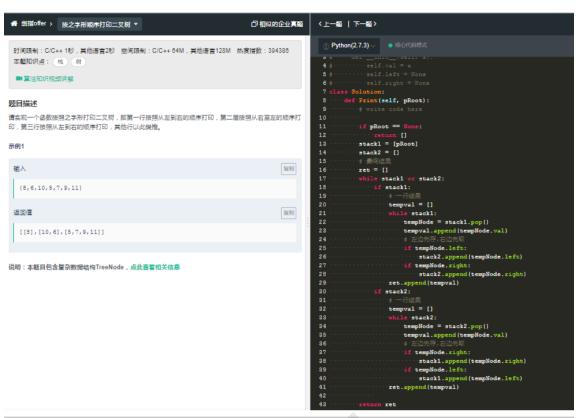


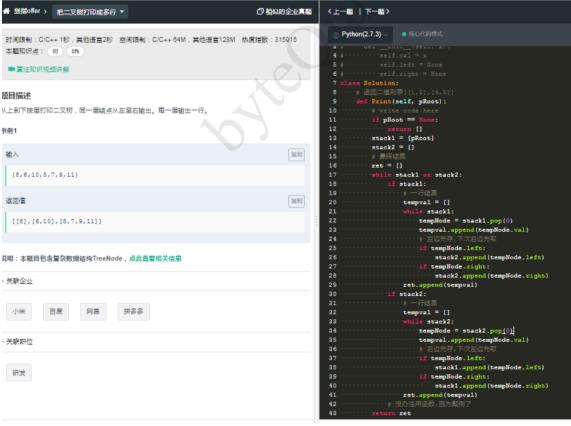
在一个排序的能表中,存在重复的结点,请删除该能表中重复的结点,重复的结点不保留,返回能表头指针。例如,能表1->2->3->4->4->5处理后为 1->2->5

示例1









例如,我们可以把一个只有根节点为1的二叉树序列化为"1,",然后通过自己的函数来解析回这 ヘーマ財

示例1

道明·太阳日有今日为晚保住均TreeMode 占业本美妇学信白

```
时间限制:C/C++ 1秒,其他语言2秒 空间限制:C/C++ 64M,其他语言128M
热度指数:496526
本题知识点: 材
```

题目描述

给定一棵二叉搜索树,请找出其中的第k小的TreeNode结点。

示例1

```
    輸入
    复制

    {5,3,7,2,4,6,8},3
    复制

    {4}
    資制

    接結点数值大小顺序第三小结点的值为4
```

```
〈上 题 | 下 题>
 ① Python(2.7.3) > · 核心代码模式
        def Serialize(self, root):
            retList = []
            def ·preOrder(root):
    if ·root == None
                     retList.append('#')
               retList.append(str(root.val))
repreOrder(root.left)
              preOrder(root.right)
            preOrder(root)
             return ''.'.join(retList)
        def Deserialize(self, s):
            resList = s.split()
               ef -depreOrder():
---if-resList == -[]:
                 rootval = resList.pop(0)
                 if rootval == '#':
                 node = TreeNode(int(rootval))
                 ·leftNode = depreOrder()
                 rightNode = depreOrder()
                 node.left = leftNode
                 node.right = rightNode
                       rn node
            ·pRoot ·= ·depreOrder()
·<mark>return</mark> ·pRoot
```

