

剑指offer > 二维数组中的查找

相似的企业真题 下一题>

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：2044457
本题知识点： 数组
算法知识视频讲解

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该整数。

[

[1,2,8,9],

[2,4,9,12],

[4,7,10,13],

[6,8,11,15]

]

给定 target = 7，返回 true。

给定 target = 3，返回 false。

示例1

Python(2.7.3)

核心代码模式

```
1 #-*-coding:utf-8-*-
2 class Solution:
3     # array 二维列表
4     def Find(self, target, array):
5         # write code here
6         # 1:2:3
7         # 2:3:5
8         # 9:6:7
9         if array == []:
10             return False
11         i = 0
12         j = len(array[0])-1
13         while i < len(array) and j >= 0:
14             if array[i][j] == target:
15                 return True
16             elif array[i][j] < target:
17                 i += 1
18             else:
19                 j -= 1
20         return False
```

剑指offer > 替换空格

上一题 | 下一题>

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
算法知识视频讲解

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

输入

"We Are Happy"

复制

返回值

"We%20Are%20Happy"

复制

Python(2.7.3)

核心代码模式

```
1 #
2 # 代码中的类名、方法名、参数名已经指定，请勿修改
3 #
4 #
5 # @param s string字符串
6 # @return string字符串
7 #
8 class Solution:
9     def replaceSpace(self, s):
10         # write code here
11         slist = []
12         for i in s:
13             if i == ' ':
14                 slist.append('%20')
15             else:
16                 slist.append(i)
17         return ''.join(slist)
18
19
```

剑指offer > 从尾到头打印链表

相似的企业真题 上一题 | 下一题>

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：1520621
本题知识点： 链表
算法知识视频讲解

题目描述

输入一个链表，按链表从尾到头的顺序返回一个ArrayList。

示例1

输入

{ 67, 0, 24, 58 }

复制

返回值

[58, 24, 0, 67]

复制

说明：本题目包含复杂数据结构ListNode，点击查看相关信息

Python(2.7.3)

核心代码模式

```
1 #-*-coding:utf-8-*-
2 # class ListNode: ... # what链表
3 # def __init__(self, x):
4 #     self.val = x
5 #     self.next = None
6
7 class Solution:
8     # 返回从尾部到头部的列表值序列，例如 [1,2,3]
9     def printListFromTailToHead(self, listNode):
10         # write code here
11         '''
12         res = []
13         pTmp = listNode
14         while pTmp:
15             res.insert(0,pTmp.val) ... # 采用insert的形式形成倒叙
16             pTmp = pTmp.next
17         return res
18         '''
19         res = []
20         pTmp = listNode
21         while pTmp:
22             res.append(pTmp.val)
23             pTmp = pTmp.next
24         return res[::-1]
```

剑指offer > 用两个栈实现队列

相似的企业真题

< 上一题 | 下一题 >

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：826746

本题知识点：栈

算法知识视频讲解

题目描述

用两个栈来实现一个队列，完成队列的Push和Pop操作。 队列中的元素为int类型。

关联企业

字节跳动 米哈游 深信服 vivo

关联职位

研发 测试 前端

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2
3 # 定义两个栈
4 acceptStack = []
5 outputStack = []
6 class Solution:
7     def push(self, node):
8         # write code here
9         acceptStack.append(node)
10        # push操作不需要return
11    def pop(self):
12        # return xx
13        if outputStack == []:
14            while acceptStack:
15                outputStack.append(acceptStack.pop())
16        if outputStack:
17            return outputStack.pop()
18        else:
19            return None
```

剑指offer > 旋转数组的最小数字

我的提交

< 上一题 | 下一题 >

时间限制：C/C++ 3秒，其他语言6秒 空间限制：C/C++ 64M，其他语言128M

热度指数：1103957

本题知识点：二分

算法知识视频讲解

题目描述

把一个数组最开始的若干个元素搬到数组的末尾，我们称之为数组的旋转。
输入一个非递减排序的数组的一个旋转，输出旋转数组的最小元素。
NOTE：给出的所有元素都大于0，若数组大小为0，请返回0。

示例1

输入

[3, 4, 5, 1, 2]

复制

返回值

1

复制

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def minNumberInRotateArray(self, rotateArray):
4         # write code here
5         # 非递减排序的旋转... 有性质就是：
6         # 1. 若中间的数大于右边，那么最小值在右边
7         # 2. 若中间的数小于最右边，那么最小值在左边
8         # 3. 若中间值小于上一位的值，那么他就是最小的，若不是那他就是一个递增序列中右边不是最小
9         left = 0
10        right = len(rotateArray)-1
11        while left <= right:
12            mid = (left+right) // 2
13            if rotateArray[mid] < rotateArray[mid-1]:
14                # 正好是这个断点
15                return rotateArray[mid]
16            elif rotateArray[mid] > rotateArray[right]:
17                left = mid + 1
18            else:
19                right = mid - 1
20        return 0
```

剑指offer > 调整数组顺序使奇数位于偶数前面

< 上一题 | 下一题 >

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：23927

本题知识点：数组

算法知识视频讲解

题目描述

输入一个整数数组，实现一个函数来调整该数组中数字的顺序，使得所有的奇数位于数组的前半部分，所有的偶数位于数组的后半部分，并保证奇数和奇数，偶数和偶数之间的相对位置不变。

示例1

输入

[1, 2, 3, 4]

复制

返回值

[1, 3, 2, 4]

复制

Python(2.7.3)

核心代码模式

```
1 #
2 # 代码中的类名、方法名、参数名已经指定，请勿修改，直接返回方法规定的值即可
3 #
4 #
5 # @param array int整型一维数组
6 # @return int整型一维数组
7 #
8 class Solution:
9     def reOrderArray(self, array):
10        # write code here
11        if array == []:
12            return []
13        n = len(array)
14        indexodd = 0
15        for i in range(n):
16            if array[i] % 2 == 0:
17                continue
18            else:
19                array.insert(indexodd, array.pop(i))
20                indexodd += 1 # 插入后指针移动到一个要插入的位置上
21        return array
```

剑指offer > 链表中倒数第k个结点

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：48595
本题知识点： 链表
算法知识视频讲解

题目描述

输入一个链表，输出该链表中倒数第k个结点。
如果该链表长度小于k，请返回空。

示例1

输入

{1,2,3,4,5},1

返回值

{5}

说明：本题目包含复杂数据结构ListNode，[点击查看相关信息](#)

关联企业

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
3 #.....self.val:=x
4 #.....self.next:=None
5 #
6 # 代码中的类名、方法名、参数名已经指定，请勿修改，只
7 #
8 #
9 # @param pHead ListNode类
10 # @param k int整型
11 # @return ListNode类
12 #
13 class Solution:
14     def FindKthToTail(self, pHead, k):
15         # write code here
16         fastpro = pHead
17         slowpro = pHead
18         while k > 0 and fastpro != None:
19             # 走到fastpro == None,或k=0
20             fastpro = fastpro.next
21             k -= 1
22         if k != 0 and fastpro == None:
23             # 走完链表但是k!=0
24             return None
25         while fastpro != None:
26             # 走到fastpro为None为止
27             fastpro = fastpro.next
28             slowpro = slowpro.next
29         return slowpro # 不能偷进进去
```

剑指offer > 反转链表

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：1134422
本题知识点： 链表
算法知识视频讲解

题目描述

输入一个链表，反转链表后，输出新链表的表头。

示例1

输入

{1,2,3}

返回值

{3,2,1}

说明：本题目包含复杂数据结构ListNode，[点击查看相关信息](#)

关联企业

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
1 #-*-coding:utf-8 -*-
2 #class ListNode:
3 #.....def __init__(self, x):
4 #.....self.val:=x
5 #.....self.next:=None
6 class Solution:
7     # 返回ListNode
8     def ReverseList(self, pHead):
9         # write code here
10         # 表头的指向读为None,第二个的指向读到第一个,第三个的指向读到第二个,直到没有为止
11         # 链表从左右向右,表头在左
12         if pHead == None:
13             return None
14         if pHead.next == None:
15             return pHead
16         leftNode = pHead
17         midNode = pHead.next
18         rightNode = pHead.next.next
19         leftNode.next = None
20         while rightNode != None:
21             midNode.next = leftNode # 只有在这块方向
22             leftNode = midNode
23             midNode = rightNode
24             rightNode = rightNode.next
25         midNode.next = leftNode # 查最后一次块方向, right一直为游荡链
26         return midNode
```

剑指offer > 合并两个排序的链表

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：898821
本题知识点： 链表
算法知识视频讲解

题目描述

输入两个单调递增的链表，输出两个链表合成后的链表，当然我们需要合成后的链表满足单调不减规则。

示例1

输入

{1,3,5},{2,4,6}

返回值

{1,2,3,4,5,6}

说明：本题目包含复杂数据结构ListNode，[点击查看相关信息](#)

关联企业

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
9 #.....# write code here
10 #.....if pHead1 == None:
11 #.....return pHead2
12 #.....if pHead2 == None:
13 #.....return pHead1
14 #.....newHead = pHead1 if pHead1.val < pHead2.val else pHead2
15 #.....pTmp1 = pHead1
16 #.....pTmp2 = pHead2
17 #.....if newHead == pHead1:
18 #.....pTmp1 = pTmp1.next
19 #.....else:
20 #.....pTmp2 = pTmp2.next
21 #.....pTmp3 = newHead
22 #.....while pTmp2 and pTmp1:
23 #.....if pTmp1.val < pTmp2.val:
24 #.....pTmp3.next = pTmp1
25 #.....pTmp3 = pTmp1
26 #.....pTmp1 = pTmp1.next
27 #.....else:
28 #.....pTmp3.next = pTmp2
29 #.....pTmp3 = pTmp2
30 #.....pTmp2 = pTmp2.next
31 #.....if pTmp1 == None:
32 #.....pTmp3.next = pTmp2
33 #.....else:
34 #.....pTmp3.next = pTmp1
35 #.....return newHead
```

剑指offer > 树的子结构

相似的企业真题

< 上一题 | 下一题 >

我的提交

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：849647

本题知识点： 二叉树 树

算法知识视频讲解

题目描述

输入两棵二叉树A，B，判断B是不是A的子结构。（ps：我们约定空树不是任意一个树的子结构）

示例1

输入

[[8, 8, 9, 9, 2, 2, 5], [8, 9, 2, 2]]

返回值

true

说明：本题目包含复杂数据结构TreeNode，[点击查看相关信息](#)

Python(2.7.3)

核心代码模式

```
1. -*- coding:utf-8 -*-
2. class TreeNode:
3.     def __init__(self, x):
4.         self.val = x
5.         self.left = None
6.         self.right = None
7. class Solution:
8.     def HasSubtree(self, pRoot1, pRoot2):
9.         if pRoot2 == None or pRoot1 == None:
10.             return False
11.         def hasEqual(pRoot1, pRoot2):
12.             if pRoot2 == None:
13.                 return True
14.             if pRoot1 == None:
15.                 return False
16.             if pRoot1.val == pRoot2.val:
17.                 return hasEqual(pRoot1.left, pRoot2.left) and hasEqual(pRoot1.right, pRoot2.right)
18.             if pRoot1.val != pRoot2.val:
19.                 ret = hasEqual(pRoot1, pRoot2)
20.                 if ret:
21.                     return True
22.                 ret = self.HasSubtree(pRoot1.left, pRoot2)
23.                 if ret:
24.                     return True
25.                 ret = self.HasSubtree(pRoot1.right, pRoot2)
26.                 return ret
```

剑指offer > 顺时针打印矩阵

相似的企业真题

< 上一题 | 下一题 >

我的提交

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：903569

本题知识点： 数组

算法知识视频讲解

题目描述

输入一个矩阵，按照从外向里以顺时针的顺序依次打印出每一个数字，例如，如果输入如下4 X 4矩阵： 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 则依次打印出数字1,2,3,4,8,12,16,15,14,13,9,5,6,7,11,10.

示例1

输入

[[1,2],[3,4]]

返回值

[1,2,4,3]

Python(2.7.3)

核心代码模式

```
1. -*- coding:utf-8 -*-
2. class Solution:
3.     # matrix类型为二维列表，需要返回列表
4.     def printMatrix(self, matrix):
5.         # write code here
6.         result = []
7.         while matrix:
8.             result += matrix.pop(0) # 列表拼接
9.             result.extend(matrix.pop(0))
10.            if not matrix or not matrix[0]:
11.                break
12.            matrix = self.turn(matrix) # 旋转一下
13.        return result
14.
15.     def turn(self, matrix):
16.         # 完成装置
17.         num_r = len(matrix) # 打印行长
18.         num_c = len(matrix[0]) # 打印列长
19.         newmat = []
20.         for i in range(num_c):
21.             newmat2 = [] # 把一列存为一行
22.             for j in range(num_r):
23.                 newmat2.append(matrix[j][i]) # 顺的一列
24.             newmat.append(newmat2)
25.         newmat.reverse() # 每一行都逆一下
26.        return newmat
```

剑指offer > 栈的压入、弹出序列

相似的企业真题

< 上一题 | 下一题 >

我的提交

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：634473

本题知识点： 栈

算法知识视频讲解

题目描述

输入两个整数序列，第一个序列表示栈的压入顺序，请判断第二个序列是否可能为该栈的弹出顺序。假设压入栈的所有数字均不相等。例如序列1,2,3,4,5是某栈的压入顺序，序列4,5,3,2,1是该压栈序列对应的一个弹出序列，但4,3,5,1,2就不可能是该压栈序列的弹出序列。（注意：这两个序列的长度是相等的）

示例1

输入

[1,2,3,4,5], [4,3,5,1,2]

返回值

false

Python(2.7.3)

核心代码模式

```
1. -*- coding:utf-8 -*-
2. class Solution:
3.     def IsPopOrder(self, pushV, popV):
4.         # write code here
5.         # 首先要有一个空栈
6.         # 按照pushV的形式压入数据
7.         # 弹出的时候需要循环判断是否弹出，可连续弹出至不相等
8.         # 压入时就判断
9.         # 判断条件：压入的头是否和popV的数据是否相等
10.        stack = []
11.        index = 0
12.        for i in pushV:
13.            stack.append(i)
14.            while stack and stack[-1]==popV[index]:
15.                stack.pop()
16.                index += 1
17.        return True if len(stack)== 0 else False
18.
```

剑指offer > 从上往下打印二叉树

相似的企业真题

< 上一题 | 下一题 >

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: 640779

本题知识点: 队列 树

算法知识视频讲解

题目描述

从上往下打印出二叉树的每个节点，同层节点从左至右打印。

示例1

输入

{5,4,#,3,#,2,#,1}

复制

返回值

[5,4,3,2,1]

复制

说明: 本题目包含复杂数据结构TreeNode, 点击查看相关信息

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7 class Solution:
8     # 返回从上到下每个节点值列表，例：[1,2,3]
9     def PrintFromTopToBottom(self, root):
10         # write code here
11         if root == None:
12             return []
13         superNone = [root]
14         ret = []
15         while superNone:
16             temp = superNone[0]
17             ret.append(temp.val)
18             if temp.left:
19                 superNone.append(temp.left)
20             if temp.right:
21                 superNone.append(temp.right)
22             del superNone[0]
23         return ret
```

剑指offer > 二叉搜索树的后序遍历序列

相似的企业真题

< 上一题 | 下一题 >

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: 776802

本题知识点: 栈 树

算法知识视频讲解

题目描述

输入一个整数数组，判断该数组是不是某二叉搜索树的后序遍历的结果。如果是则返回true,否则返回false。假设输入的数组中的任意两个数字都互不相同。（ps：我们约定空树不是二叉搜索树）

示例1

输入

[4,8,6,12,16,14,10]

复制

返回值

true

复制

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def VerifySequenceOfBST(self, sequence):
4         # 左节点比根节点小,右节点比根节点大
5         # write code here
6         if sequence == []:
7             return False
8         rootNum = sequence[-1] # 根节点
9         del sequence[-1]
10         index = None
11         # 找左右子树分开的索引
12         for i in range(len(sequence)-1):
13             if index == None and sequence[i] > rootNum:
14                 index = i
15             if index != None and sequence[i] < rootNum:
16                 return False
17         # 左右可以为空
18         if sequence[:index] == []:
19             leftRet = True
20         else:
21             leftRet = self.VerifySequenceOfBST(sequence[:index])
22         if sequence[index:] == []:
23             rightRet = True
24         else:
25             rightRet = self.VerifySequenceOfBST(sequence[index:])
26         return leftRet and rightRet
```

剑指offer > 二叉树中和为某一值的路径

相似的企业真题

< 上一题 | 下一题 >

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: 树

算法知识视频讲解

题目描述

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

输入

{10,5,12,4,7},22

复制

返回值

[[10,5,7],[10,12]]

复制

示例2

输入

{10,5,12,4,7},15

复制

返回值

[]

复制

Python(2.7.3)

核心代码模式

```
6 # self.right = None
7 import copy
8 class Solution:
9     # 返回二维列表，内部每个列表表示找到的路径
10     def FindPath(self, root, expectNumber):
11         # write code here
12         if root == None:
13             return []
14         # 存放结果
15         ret = []
16         # 存放遍历节点
17         supportNode = [root]
18         # 存放路径值
19         supportArray = [[root.val]]
20         while supportNode:
21             tempNode = supportNode[0]
22             tempArray = supportArray[0]
23             if tempNode.left == None and tempNode.right == None:
24                 if sum(tempArray) == expectNumber:
25                     ret.insert(0,tempArray)
26             if tempNode.left:
27                 supportNode.append(tempNode.left)
28                 tempval = copy.copy(tempArray)
29                 tempval.append(tempNode.left.val)
30                 supportArray.append(tempval)
31             if tempNode.right:
32                 supportNode.append(tempNode.right)
33                 tempval = copy.copy(tempArray)
34                 tempval.append(tempNode.right.val)
35                 supportArray.append(tempval)
36             del supportNode[0]
37             del supportArray[0]
38         return ret
```

剑指offer > 复杂链表的复制

相似的企业真题

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
6 # 初始化 self.random = None
7 class Solution:
8     # 返回 RandomListNode
9     def Clone(self, pHead):
10         # write code here
11         # 复制一样的node,并且添加到一个node后面
12         if pHead==None:
13             return None
14         pTmp = pHead
15         while pTmp:
16             node = RandomListNode(pTmp.label) # 生成一个节点,有3个属性
17             node.next = pTmp.next
18             pTmp.next = node
19             pTmp = node.next
20         # 断开,将原node和新的node的连接
21         # 遍历新建的node的指向
22         pTmp = pHead
23         while pTmp:
24             if pTmp.random:
25                 pTmp.next.random = pTmp.random.next
26                 pTmp = pTmp.next.next
27             else:
28                 pTmp = pTmp.next
29         pNewHead = pHead.next
30         pNewTmp = pHead.next
31         newHead = pHead.next # 链头
32         while pTmp:
33             pTmp.next = pTmp.next.next
34             if pNewTmp.next:
35                 pNewTmp.next = pNewTmp.next.next
36             pNewTmp = pNewTmp.next
37             pTmp = pTmp.next
38         return newHead
```

题目描述

输入一个复杂链表（每个节点中有节点值，以及两个指针，一个指向下一个节点，另一个特殊指针random指向一个随机节点），请对此链表进行深拷贝，并返回拷贝后的头结点。（注意，输出结果中请不要返回参数中的节点引用，否则判题程序会直接返回空）

说明：本题目包含复杂数据结构ListNode、RandomListNode，[点击查看相关信息](#)

剑指offer > 二叉搜索树与双向链表

相似的企业真题

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
3 # 初始化 init (self, x):
4 self.val = x
5 self.left = None
6 self.right = None
7 class Solution:
8     def Convert(self, pRootOfTree):
9         # write code here
10         if pRootOfTree == None:
11             return None
12         # 找最右端,因为是排序的链表
13         def find_right(node):
14             while node.right:
15                 node = node.right
16             return node
17         # 找最左端
18         leftnode = self.Convert(pRootOfTree.left)
19         rightnode = self.Convert(pRootOfTree.right)
20         retnode = leftnode # 双向链表的头
21         # 找中序遍历
22         if leftnode:
23             leftnode = find_right(leftnode)
24         else:
25             retnode = pRootOfTree # 没有左节点的话把根节点作为头
26         # 找左子树
27         pRootOfTree.left = leftnode
28         pRootOfTree.right = rightnode
29         # 找左子树
30         if leftnode:
31             leftnode.right = pRootOfTree
32         if rightnode:
33             rightnode.left = pRootOfTree
34         return retnode
```

题目描述

输入一棵二叉搜索树，将该二叉搜索树转换成一个排序的双向链表。要求不能创建任何新的结点，只能调整树中结点指针的指向。

说明：本题目包含复杂数据结构TreeNode，[点击查看相关信息](#)

关联企业

字节跳动

猎聘网

关联职位

研发

剑指offer > 把数组排成最小的数

相似的企业真题

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def PrintMinNumber(self, numbers):
4         # write code here
5         if not numbers:
6             return ""
7         # 比较规则, 两两比较, 小的在前
8         lmb = lambda n1, n2: int(str(n1)+str(n2)) - int(str(n2)+str(n1))
9         array = sorted(numbers, cmp=lmb)
10        # cmp 比较的函数, 这个具有两个参数,
11        # 参数的值都是从可迭代对象中取出, 此函数必须遵守的规则为, 大于则返回1, 小于则返回-1, 等于则返回0
12        return ''.join([str(i) for i in array])
```

题目描述

输入一个正整数数组，把数组里所有数字拼接起来排成一个数，打印能拼接出的所有数字中最小的一个。例如输入数组[3, 32, 321]，则打印出这三个数字能排成的最小数字为321323。

示例1

输入

[3, 32, 321]

复制

返回值

"321323"

复制

剑指offer > 丑数

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：531949
本题知识点： 数学 二分

算法和知识视频讲解

题目描述

把只包含质因子2、3和5的数称作丑数（Ugly Number）。例如6、8都是丑数，但14不是，因为它包含质因子7，习惯上我们把1当做是第一个丑数。求按从小到大的顺序的第N个丑数。

示例1

输入

7

复制

返回值

8

复制

关联企业

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def GetUglyNumber_Solution(self, index):
4         # write code here
5         # 后面的丑数是前面的丑数乘2,3,5
6         if index < 1:
7             return 0
8         uglylist = [1]
9         twopoint = 0
10        threepoint = 0
11        fivepoint = 0
12        for i in range(index):
13            minvalue = min(2*uglylist[twopoint],3*uglylist[threepoint],5*uglylist[fivepoint])
14            uglylist.append(minvalue)
15            if minvalue == 2*uglylist[twopoint]:
16                twopoint += 1
17            if minvalue == 3*uglylist[threepoint]:
18                threepoint += 1
19            if minvalue == 5*uglylist[fivepoint]:
20                fivepoint += 1
21        return uglylist[-1] # 为什么是index-1不是-1,因为开始加了1
22
23
24 # 死循环找丑数 (只有2,3,5相乘获得)
25 # 判断一个数是不是丑数,循环除以2,到不能整除
26 # 循环除以3到不能整除
27 # 循环除以5到不能整除
```

剑指offer > 二叉树的深度

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：329477
本题知识点： 树

算法和知识视频讲解

题目描述

输入一棵二叉树，求该树的深度。从根结点开始，到叶结点依次经过的结点（含根、叶结点）形成树的一条路径，最长路径的长度为树的深度。

示例1

输入

复制

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7 class Solution:
8     def TreeDepth(self, pRoot):
9         # write code here
10
11        if not pRoot: return 0
12        return max(self.TreeDepth(pRoot.left), self.TreeDepth(pRoot.right)) + 1
13
14
```

剑指offer > 和为S的连续正数序列

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：462455
本题知识点： 穷举

算法和知识视频讲解

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,他在想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,你能不能很快的找出所有和为S的连续正数序列? Good Luck!

返回描述:

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

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def FindContinuousSequence(self, tsum):
4         # write code here
5         list1 = []
6         #lsum = 0
7         for i in range(1, tsum/2+2):
8             for j in range(1, i):
9                 if (j+i)*(i-j+1) == 2*tsum: # 推导式
10                    list1.append([u for u in range(j, i+1)])
11            return list1 #返回二维列表也可以
12
13
```

剑指offer > 左旋转字符串

相似的企业真题

< 上一题 | 下一题 >

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: 393379
本题知识点: 字符串
[算法知识视频讲解](#)

题目描述

汇编语言中有一种移位指令叫做循环左移(ROL)，现在有个简单的任务，就是用字符串模拟这个指令的运算结果。对于一个给定的字符序列S，请你把其循环左移K位后的序列输出。例如，字符序列S="abcXYZdef"，要求输出循环左移3位后的结果，即"XYZdefabc"。是不是很简单？OK，搞定它！

示例1

输入

"abcXYZdef",3

返回值

"XYZdefabc"

Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def LeftRotateString(self, s, n):
4         # write code here
5         if s == '':
6             return '' # return不是None
7         s = list(s)
8         string = ''
9         for i in range(n):
10             s.append(s.pop(0)) # 不能 s = s.append(s.pop(0)) 操作不赋值
11         for j in s:
12             string += j
13         return string
```

剑指offer > 翻转单词顺序列

相似的企业真题

< 上一题 | 下一题 >

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: 636014
本题知识点: 字符串
[算法知识视频讲解](#)

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 I”。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——的翻转这些单词顺序可不在行，你能帮助他么？

示例1

输入

"nowcoder. a am I"

返回值

"I am a nowcoder."

Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def ReverseSentence(self, s):
4         a = s
5         if s == '':
6             return ''
7         # write code here
8         s = s.split()
9         if s == []:
10             return a
11         string = ''
12         for i in s[::-1]:
13             i = '.' + i
14             string += i
15         return string.lstrip()
16
```

保存并提交

提交记录

自测输入

代码调试

执行结果:

保存并提交之后，这里将会显示运行结果

剑指offer > 扑克牌顺子

相似的企业真题

< 上一题 | 下一题 >

① Python(2.7.3) 核心代码模式

```
1 # -*- coding: utf-8 -*-
2 class Solution:
3     def IsContinuous(self, numbers):
4         # write code here
5         if numbers == None:
6             return False
7         if len(numbers) != 5: # 巨坑
8             return False
9         list1 = []
10        list2 = []
11        # 转变函数, 不用转变, 因为不是JqK是数字
12        for i in numbers:
13            if i == 0:
14                list2.append(i)
15            else:
16                list1.append(i)
17            count = 0
18            list1.sort()
19            for j in range(len(list1)-1):
20                count += (list1[j+1]-list1[j]-1)
21            if count == list2.count(0) or count == 0:
22                return True
23            return False
24
25
26
27
```

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: 408418

本题知识点: 字符串

算法知识视频讲解

题目描述

LL今天心情特别好,因为他去买了一副扑克牌,发现里面居然有2个大王,2个小王(一副牌原本是54张^_^)...他随机从中抽出了5张牌,想测测自己的手气,看看能不能抽到顺子,如果抽到的话,他就决定去买体育彩票,嘿嘿!!“红心A,黑桃3,小王,大王,方片5”,“Oh My God!”不是顺子.....LL不高兴了,他想了想,决定大\小\王可以看成任何数字,并且A看作1,J为11,Q为12,K为13。上面的5张牌就可以变成1,2,3,4,5(大小王分别看作2和4),“So Lucky!”。LL决定去买体育彩票啦。现在,要求你使用这副牌模拟上面的过程,然后告诉我们LL的运气如何,如果牌能组成顺子就输出true,否则就输出false。为了方便起见,你可以认为大小王是0。

示例1

输入

[0, 3, 2, 6, 4]

返回值

true

剑指offer > 孩子们的游戏(圆圈中最后剩下的数)

相似的企业真题

< 上一题 | 下一题 >

① Python(2.7.3) 核心代码模式

```
1 # -*- coding: utf-8 -*-
2 class Solution:
3     def LastRemaining_Solution(self, n, m):
4         # write code here
5         # 返回的是原始序列的下标, 新的位置下标变了, 但是在原始序列的下标不变
6         if n < 1 or m == 0:
7             return -1
8         if n == 1:
9             return 0
10        value = 0
11        for i in range(2, n+1):
12            currentvalue = (value+m)% i
13            value = currentvalue
14            return currentvalue
15
```

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: 379792

本题知识点: 数学

算法知识视频讲解

题目描述

每年六一儿童节,牛客都会准备一些小礼物去看望孤儿院的小朋友,今年亦是如此。HF作为牛客的资深元老,自然也准备了一些小游戏。其中,有个游戏是这样的:首先,让小朋友们围成一个大圈。然后,他随机指定一个数m,让编号为0的小朋友开始报数。每次喊到m-1的那个小朋友要出列唱首歌,然后可以在礼品箱中任意的挑选礼物,并且不再回到圈中,从他的下一个小朋友开始,继续0..m-1报数....这样下去.....直到剩下最后一个小朋友,可以不用表演,并且拿到牛客名贵的“名侦探柯南”典藏版(名额有限哦!!^_^)。请你试着想下,哪个小朋友会得到这份礼品呢？(注：小朋友的编号是从0到n-1)

如果没有小朋友，请返回-1

示例1

输入

5, 3

返回值

3

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：279930

本题知识点：数组

算法知识视频讲解

题目描述

给定一个数组A[0,1,...,n-1]，请构建一个数组B[0,1,...,n-1]，其中B中的元素B[i]=A[0]*A[1]*...*A[i-1]*A[i+1]*...*A[n-1]。不能使用除法。（注意：规定B[0] = A[1]*A[2]*...*A[n-1]，B[n-1] = A[0]*A[1]*...*A[n-2]。）

对于A长度为1的情况，B无意义，故而无法构建，因此该情况不会存在。

示例1

输入

[1,2,3,4,5]

返回值

[120,60,40,30,24]

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def multiply(self, A):
4         # write code here
5         B = []
6         for i in range(len(A)):
7             c = A.pop(i)
8             d = 1
9             for j in A:
10                 d *= j
11             A.insert(i, c)
12             B.append(d)
13         return B
```

保存并提交

提交记录

自测输入

代码调试

执行结果：

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：691540

本题知识点：链表

算法知识视频讲解

题目描述

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

示例1

输入

{1,2,3,3,4,4,5}

返回值

{1,2,5}

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class ListNode:
3     def __init__(self, x):
4         self.val = x
5         self.next = None
6 class Solution:
7     def deleteDuplication(self, pHead):
8         # write code here
9         if pHead == None:
10             return None
11         pTmp = pHead
12         newHead = ListNode(0)
13         npTmp = newHead
14         list1 = []
15         while pTmp:
16             list1.append(pTmp.val)
17             pTmp = pTmp.next
18         for i in list1:
19             if list1.count(i) == 1:
20                 npTmp.next = ListNode(i)
21                 npTmp = npTmp.next
22         return newHead.next
```

剑指offer > 二叉树的下一个结点

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：377698
本题知识点：树

算法知识视频讲解

题目描述

给定一个二叉树和其中的一个结点，请找出中序遍历顺序的下一个结点并且返回。注意，树中的结点不仅包含左右子结点，同时包含指向父结点的指针。

说明：本题目包含复杂数据结构TreeNode，点此查看相关信息

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2 # class TreeLinkNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7 #         self.next = None
8 class Solution:
9     def GetNext(self, pNode):
10         # write code here
11         # 如果存在右子树，找最左，就是下一个节点
12         # 如果没有右子树，看是不是父节点的左节点，是就返回，不是就继续找，找到为止
13         if pNode.right:
14             tempnode = pNode.right
15             while tempnode.left:
16                 tempnode = tempnode.left
17             return tempnode
18         else:
19             tempnode = pNode
20             while tempnode.next:
21                 if tempnode.next.left == tempnode:
22                     return tempnode.next
23                 tempnode = tempnode.next
24             return None
```

剑指offer > 对称的二叉树

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：338721
本题知识点：树

算法知识视频讲解

题目描述

请实现一个函数，用来判断一棵二叉树是不是对称的。注意，如果一个二叉树同此二叉树的镜像是一样的，那么二叉树是对称的。

示例1

输入

{8, 6, 6, 5, 7, 7, 5}

复制

返回值

true

复制

Python(2.7.3)

核心代码模式

```
1 # -*- coding:utf-8 -*-
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7 class Solution:
8     def isSymmetrical(self, pRoot):
9         # write code here
10         if pRoot == None:
11             return True
12         def isMirror(left, right):
13             if left == None and right == None:
14                 return True
15             elif left == None or right == None:
16                 return False
17             if left.val != right.val:
18                 return False
19             ret1 = isMirror(left.left, right.right)
20             ret2 = isMirror(left.right, right.left)
21             return ret1 and ret2
22         return isMirror(pRoot.left, pRoot.right)
23
```

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：394388

本题知识点： 栈 树

算法知识视频讲解

题目描述

请实现一个函数按照之字形打印二叉树，即第一行按照从左到右的顺序打印，第二层按照从右至左的顺序打印，第三行按照从左到右的顺序打印，其他行以此类推。

示例1

输入	复制
{8,6,10,5,7,9,11}	
返回值	复制
[[8],[10,6],[5,7,9,11]]	

说明：本题目包含复杂数据结构TreeNode，点此查看相关信息

< 上一题 | 下一题 >

```
Python(2.7.3) 核心代码模式
3 # -*- coding: utf-8 -*-
4 # 二叉树节点
5 # 二叉树节点
6 # 二叉树节点
7 class Solution:
8     def Print(self, pRoot):
9         # write code here
10
11         if pRoot == None:
12             return []
13         stack1 = [pRoot]
14         stack2 = []
15         # 最终结果
16         ret = []
17         while stack1 or stack2:
18             if stack1:
19                 # 一行结果
20                 tempval = []
21                 while stack1:
22                     tempNode = stack1.pop()
23                     tempval.append(tempNode.val)
24                     # 左孩子先存，右孩子先取
25                     if tempNode.left:
26                         stack2.append(tempNode.left)
27                     if tempNode.right:
28                         stack2.append(tempNode.right)
29                 ret.append(tempval)
30             if stack2:
31                 # 一行结果
32                 tempval = []
33                 while stack2:
34                     tempNode = stack2.pop()
35                     tempval.append(tempNode.val)
36                     # 左孩子先存，右孩子先取
37                     if tempNode.right:
38                         stack1.append(tempNode.right)
39                     if tempNode.left:
40                         stack1.append(tempNode.left)
41                 ret.append(tempval)
42         return ret
```

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：315015

本题知识点： 树 bfs

算法知识视频讲解

题目描述

从上到下按层打印二叉树，同一层结点从左至右输出。每一层输出一行。

示例1

输入	复制
{8,6,10,5,7,9,11}	
返回值	复制
[[8],[6,10],[5,7,9,11]]	

说明：本题目包含复杂数据结构TreeNode，点此查看相关信息

关联企业

小米 百度 网易 拼多多

关联职位

研发

< 上一题 | 下一题 >

```
Python(2.7.3) 核心代码模式
3 # -*- coding: utf-8 -*-
4 # 二叉树节点
5 # 二叉树节点
6 # 二叉树节点
7 class Solution:
8     # 返回二维列表 [[1,2],[4,5]]
9     def Print(self, pRoot):
10         # write code here
11         if pRoot == None:
12             return []
13         stack1 = [pRoot]
14         stack2 = []
15         # 最终结果
16         ret = []
17         while stack1 or stack2:
18             if stack1:
19                 # 一行结果
20                 tempval = []
21                 while stack1:
22                     tempNode = stack1.pop(0)
23                     tempval.append(tempNode.val)
24                     # 左孩子先存，下次右孩子先取
25                     if tempNode.left:
26                         stack2.append(tempNode.left)
27                     if tempNode.right:
28                         stack2.append(tempNode.right)
29                 ret.append(tempval)
30             if stack2:
31                 # 一行结果
32                 tempval = []
33                 while stack2:
34                     tempNode = stack2.pop(0)
35                     tempval.append(tempNode.val)
36                     # 右孩子先存，下次左孩子先取
37                     if tempNode.left:
38                         stack1.append(tempNode.left)
39                     if tempNode.right:
40                         stack1.append(tempNode.right)
41                 ret.append(tempval)
42         # 没办调用函数，因为限制了
43         return ret
```

剑指offer > 序列化二叉树

相似的企业真题

< 上一题 | 下一题 >

Python(2.7.3) 核心代码模式

```
class Solution:
    def Serialize(self, root):
        # write code here
        # 采用前序方法
        retList = []
        def preOrder(root):
            if root == None:
                retList.append('#')
                return
            retList.append(str(root.val))
            preOrder(root.left)
            preOrder(root.right)
        preOrder(root)
        return ''.join(retList)

    def Deserialize(self, s):
        # write code here
        resList = s.split()
        def depreOrder():
            if resList == []:
                return None
            rootval = resList.pop(0)
            if rootval == '#':
                return None
            node = TreeNode(int(rootval))
            leftNode = depreOrder()
            rightNode = depreOrder()
            node.left = leftNode
            node.right = rightNode
            return node
        pRoot = depreOrder()
        return pRoot
```

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：402848
本题知识点： 队列 树

算法知识视频讲解

题目描述

请实现两个函数，分别用来序列化和反序列化二叉树

二叉树的序列化是指：把一棵二叉树按照某种遍历方式的结果以某种格式保存为字符串，从而使得内存中建立起来的二叉树可以持久保存。序列化可以基于先序、中序、后序、层序的二叉树遍历方式来进行修改，序列化的结果是一个字符串，序列化时通过 某种符号表示空节点（#），以！表示一个结点值的结束（value!）。

二叉树的反序列化是指：根据某种遍历顺序得到的序列化字符串结果str，重构二叉树。

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个二叉树

示例1

输入

{8,6,10,5,7,9,11}

复制

返回值

{8,6,10,5,7,9,11}

复制

说明：本题目含有多组测试用例TreeNode 点此查看相关信息

Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 # class TreeNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7 class Solution:
8     # 返回对应节点TreeNode
9     def KthNode(self, pRoot, k):
10         # write code here
11         # 中序遍历存入列表，返回第k个
12         ret = []
13         # 中序遍历
14         def midOrder(pRoot):
15             if pRoot == None:
16                 return None
17             midOrder(pRoot.left)
18             ret.append(pRoot)
19             midOrder(pRoot.right)
20         midOrder(pRoot)
21         if len(ret) < k or k < 1:
22             return None
23         return ret[k-1]
```

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： 树

算法知识视频讲解

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

输入

{5,3,7,2,4,6,8},3

复制

返回值

{4}

复制

说明

按结点数值大小顺序第三小结点的值为4

剑指offer > 剪绳子

相似的企业真题

< 上一题

① Python(2.7.3) 核心代码模式

```
1 # -*- coding:utf-8 -*-
2 class Solution:
3     def cutRope(self, number):
4         # Write code here
5         def numsmup(num,k):
6             # 快手笔试灵感
7             s = num // k
8             d = num % k
9             return (pow(s,k-d)*pow(s+1,d))
10        maxnum = 1
11        for k in range(2,(number+3)//2):
12            if maxnum < numsmup(number,k):
13                maxnum = numsmup(number,k)
14        return maxnum
15
```

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：207978

本题知识点：数学

算法知识视频讲解

题目描述

给你一根长度为n的绳子，请把绳子剪成整数长的m段（m、n都是整数，n>1并且m>1，m<=n），每段绳子的长度记为k[1],...,k[m]。请问k[1]*...*k[m]可能的最大乘积是多少？例如，当绳子的长度是8时，我们把它剪成长度分别为2、3、3的三段，此时得到的最大乘积是18。

输入描述:

输入一个数n，意义见题面。（2 <= n <= 60）

返回值描述:

输出答案。

示例1

输入

8

复制

返回值

18

复制

保存并提交

提交记录

自测输入

代码调试

执行结果:

保存并提交之后，这里将会显示运行结果

本题累计提交 52 次 加载上次通过的代码