Assignment #7: 20250402 Mock Exam

Updated 1624 GMT+8 Apr 2, 2025

2025 spring, Complied by <mark>同学的姓名、院系</mark>

1. 题目

E05344:最后的最后

http://cs101.openjudge.cn/practice/05344/

思路:

代码:

#48800089提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n,k=map(int,input().split())
nums=[i for i in range(1,n+1)]
start=0
while len(nums)>1:
    kill=(start+k-1)%len(nums)
    t=nums.pop(kill)
    start=kill

    print(t,end=' ')
```

基本信息

#: 48800089
题目: E5344
提交人: 24n2300093007
内存: 3636kB
时间: 21ms
语言: Python3
提交时间: 2025-04-02 16:25:03

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M02774: 木材加工

binary search, http://cs101.openjudge.cn/practice/02774/

思路:

代码:

#48812989提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def max_wood_length():
    N, K = map(int, input().split())
    logs = []
    for _ in range(N):
        logs.append(int(input()))

    left = 1
    right = max(logs)
    answer = 0

    while left <= right:
        mid = (left + right) // 2
        total = 0
        for log in logs:
            total += log // mid
        if total >= K:
            answer = mid
            left = mid + 1
        else:
            right = mid - 1

    print(answer)

max_wood_length()
```

基本信息

#: 48812989
题目: 02774
提交人: 24n2300093007
内存: 3932kB
时间: 38ms
语言: Python3
提交时间: 2025-04-03 18:08:51

M07161:森林的带度数层次序列存储

tree, http://cs101.openjudge.cn/practice/07161/

思路:

代码:

#48816847提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

class Node:
    def __init__(self, value):
        self.value = value
        self.children = []

def build_tree(sequence):
    if not sequence:
        return None

    nodes = []
    # 将序列分成节点和度数的对
    for i in range(0, len(sequence), 2):
        char = sequence[i]
        degree = int(sequence[i+1])
        nodes.append((char, degree))

    if not nodes:
        return None

    root = Node(nodes[0][0])
    queue = deque()
    queue.append((root, nodes[0][1]))

    index = 1
    while queue and index < len(nodes):
        current_node, degree = queue.popleft()
        # 接下来的degree个节点是当前节点的子节点
        children_nodes = nodes[index: index + degree]
        index += degree
        for char, child_degree in children_nodes:
            child_node = Node(char)
            current_node.children.append(child_node)
            queue.append((child_node, child_degree))
    return root

def postorder_traversal(root):
    result = []
    def traverse(node):
        if node:
            for child in node.children:
                traverse(child)
            result.append(node.value)
    traverse(root)
    return result

n = int(input())
trees = []
for _ in range(n):
    parts = input().split()
    tree = build_tree(parts)
    trees.append(tree)

postorder = []
for tree in trees:
    postorder.extend(postorder_traversal(tree))

print(' '.join(postorder))
```

基本信息

#:	48816847
题目:	07161
提交人:	24n2300093007
内存:	3704kB
时间:	22ms
语言:	Python3
提交时间:	2025-04-04 14:29:45

M18156:寻找离目标数最近的两数之和

two pointers, http://cs101.openjudge.cn/practice/18156/

思路:

代码:

#48817160提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n = int(input())
nums = list(map(int, input().split()))
nums = sorted(nums)

def search_mid(nums, target):
    left, right = 0, len(nums) - 1
    closest_sum = nums[left] + nums[right]
    min_diff = abs(target - closest_sum)

    while left < right:
        current_sum = nums[left] + nums[right]
        current_diff = abs(target - current_sum)

        if current_diff < min_diff:
            min_diff = current_diff
            closest_sum = current_sum
        elif current_diff == min_diff:
            if current_sum < closest_sum:
                closest_sum = current_sum

        if current_sum < target:
            left += 1
        elif current_sum > target:
            right -= 1
        else:
            return current_sum

    return closest_sum

print(search_mid(nums, n))
```

基本信息

#: 48817160
题目: 18156
提交人: 24n2300093007
内存: 15876kB
时间: 79ms
语言: Python3
提交时间: 2025-04-04 15:17:10

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M18159: 个位为 1 的质数个数

sieve, http://cs101.openjudge.cn/practice/18159/

思路:

代码:

#48826290提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

def get_numbers_ending_with_1(n):
    numbers = []
    k = 1
    while True:
        num = 10 * k + 1
        if num >= n:
            break
        numbers.append(num)
        k += 1
    return numbers

T = int(input())
for case in range(1, T + 1):
    n = int(input())
    numbers = get_numbers_ending_with_1(n)
    primes = [num for num in numbers if is_prime(num)]
    print(f"Case{case}:")
    if not primes:
        print("NULL")
    else:
        print(' '.join(map(str, primes)))
```

基本信息

#: 48826290
题目: 18159
提交人: 24n2300093007
内存: 11444kB
时间: 7746ms
语言: Python3
提交时间: 2025-04-05 20:10:10

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

M28127:北大夺冠

hash table, http://cs101.openjudge.cn/practice/28127/

思路:

代码:

#48817717提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
M=int(input())
dic1={}
data=[]
all_data=[]
for i in range(M):
    univ,test_no,p_f=map(str,input().split(','))
    data.append(univ)

    if p_f=='no':
        pass
    else:
        if univ not in dic1:
            dic1[univ]=[]
        if test_no not in dic1[univ]:
            dic1[univ].append(test_no)

for y in dic1.keys():
    all_data.append((y,len(dic1[y]),data.count(y)))
for y1 in data:
    if y1 not in dic1.keys():
        all_data.append((y1,0,data.count(y1)))

all_data=sorted(all_data,key =lambda x:(-x[1],x[2],x[0]))
counting=0

for (x,y,z) in all_data:
    counting+=1
    if counting<=12:
        print(counting,x,y,z)
```

基本信息

#: 48817717
题目: 28127
提交人: 24n2300093007
内存: 3688kB
时间: 24ms
语言: Python3
提交时间: 2025-04-04 16:44:53

2. 学习总结和收获

这次月考让我意识到自己在树结构处理和算法细节上还存在不足。森林的带度数层次序列存储问题反映出我对复杂数据结构的转换掌握不够熟练，需要加强树和森林相关算法的练习。个位为 1 的质数统计题第一次提交的代码出错，而到现在，还没找到代码中的问题。。