

Assignment #7: 20250402 Mock Exam

Updated 1624 GMT+8 Apr 2, 2025

2025 spring, Compiled by 叶靖、信管

说明:

1. **月考:** AC3 (请改为同学的通过数)。考试题目都在“题库 (包括计概、数算题目)”里面, 按照数字题号能找到, 可以重新提交。作业中提交自己最满意版本的代码和截图。
2. **解题与记录:**

对于每一个题目, 请提供其解题思路 (可选), 并附上使用Python或C++编写的源代码 (确保已在OpenJudge, Codeforces, LeetCode等平台上获得Accepted)。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。(推荐使用Typora <https://typoraio.cn> 进行编辑, 当然你也可以选择Word。)无论题目是否已通过, 请标明每个题目大致花费的时间。
3. ****提交安排:** **提交时, 请首先上传PDF格式的文件, 并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像, 提交的文件为PDF格式, 并且“作业评论”区包含上传的.md或.doc附件。
4. ****延迟提交:** **如果你预计无法在截止日期前提交作业, 请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。

请按照上述指导认真准备和提交作业, 以保证顺利完成课程要求。

1. 题目

E05344:最后的最后

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

思路:

代码:

```
n, k = map(int, input().split()) queue = list(range(1, n + 1)) output = []  
  
index = 0 k -= 1  
  
while len(queue) > 1: index = (index + k) % len(queue) output.append(queue.pop(index))  
  
print(' '.join(map(str, output)))
```

代码运行截图（至少包含有"Accepted"）



M02774: 木材加工

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

思路：

代码：

```
def max_wood(K, lengths): left, right = 1, max(lengths) ans = 0 while left <= right: mid = (left + right) // 2 num_wood = sum(wood // mid for wood in lengths) if num_wood >= K: ans = mid left = mid + 1 else: right = mid - 1 return ans
```

```
N, K = map(int, input().split()) lengths = [int(input()) for _ in range(N)]
```

```
print(max_wood(K, lengths))
```

代码运行截图（至少包含有"Accepted"）



M07161: 森林的带度数层次序列存储

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

思路：

代码：

```
def build_tree(level_order): nodes = # 存储所有节点 queue = [] tokens = level_order.split()
```

```
    i = 0
    root = tokens[i]
    nodes[root] = []
    queue.append((root, int(tokens[i + 1])))
    i += 2

    while i < len(tokens):
        parent, children_count = queue.pop(0)
        for _ in range(children_count):
            child = tokens[i]
            nodes[child] = []
            nodes[parent].append(child)
            queue.append((child, int(tokens[i + 1])))
            i += 2
```

```
return root, nodes
```

```
def postorder_traversal(root, nodes, result): for child in nodes[root]: postorder_traversal(child, nodes, result) result.append(root)
```

```
n = int(input().strip()) forests = [input().strip() for _ in range(n)]
```

```
result = [] for tree in forests: root, nodes = build_tree(tree) postorder_traversal(root, nodes, result)
```

```
print(" ".join(result))
```

代码运行截图 （至少包含有"Accepted"）



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:

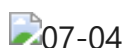
```
T = int(input()) nums = list(map(int, input().split())) nums.sort()
```

```
left, right = 0, len(nums) - 1 ans = nums[left] + nums[right] while left < right: sum = nums[left] + nums[right] if sum == T: print(sum) exit() if abs(sum - T) < abs(ans - T): ans = sum elif abs(sum - T) == abs(ans - T): ans = min(sum, ans)
```

```
if sum > T:
    right -= 1
elif sum < T:
    left += 1
```

```
print(ans)
```

代码运行截图 （至少包含有"Accepted"）



M18159:个位为 1 的质数个数

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

思路:

代码:

```
import math
def prime_num(num):
    if num < 2: return False
    if num == 2: return True
    if num % 2 == 0: return False
    for i in range(3, int(math.sqrt(num)) + 1, 2):
        if num % i == 0: return False
    return True
```

```
def sieve(limit=10000):
    primes = [True] * (limit + 1)
    primes[0], primes[1] = False, False
    for i in range(2, int(math.sqrt(limit)) + 1):
        if primes[i]:
            for j in range(i * i, limit + 1, i):
                primes[j] = False
    return [i for i in range(2, limit + 1) if primes[i]]
def prime_1(primes):
    return [p for p in primes if p % 10 == 1]
```

```
T = int(input())
primes = sieve(10000)
for case_num in range(1, T + 1):
    n = int(input())
    ans = []
    for p in prime_1(primes):
        if p < n:
            print(f'Case:')
            if ans:
                print(' '.join(ans))
            else:
                print('NULL')
```

代码运行截图 (至少包含有"Accepted")



M28127:北大夺冠

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

思路:

代码:

```
def analyst(records):
    data = {}
    for rec in records:
        team, title, result = map(str, rec.split(','))
        if team not in data:
            data[team] = {'num': 0, 'title': set()}
        data[team]['num'] += 1
        if result == 'yes':
            data[team]['title'].add(title)
```

```
ranking = sorted(data.items(), key=lambda x: (-len(x[1]['title']), x[1]['num'], x[0]))
for ranks, (team, datas) in enumerate(ranking[:12], start=1):
    print(ranks, team, len(datas['title']), datas['num'])
```

```
M = int(input())
records = []
for _ in range(M):
    records.append(input())
analyst(records)
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")



2. 学习总结和收获

如果发现作业题目相对简单, 有否寻找额外的练习题目, 如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

本次月考题目比起上次月考而言在解题时会相比较容易点了, 但是还是缺乏训练, 月考时需要不断尝试和调整才能在时限内完成题目需求。月考题目里有题是关于树的, 果然非常难, 一开始完全

无从下手，后来自己尝试的思路还是会有bug，且提供的案例不够，消耗了不少时间还是没能解出来。北大夺冠的题目虽然月考时没时间去解，但考后自己尝试时思路完全正确，确实有让我感到非常有成就感。