

# Assignment #7: 20250402 Mock Exam

Updated 1624 GMT+8 Apr 2, 2025

2025 spring, Compiled by <mark>汤伟杰，信息管理系</mark>

## 说明：

1. **月考：**AC?<mark>6</mark>。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：

考试时候没看题目用的 deque，课后用的循环链表，搓了半天才 ac，好麻烦。。

代码：

```
class Node:
    def __init__(self,value,next=None):
        self.value=value
        self.next=next

class CircleLinkedList:
    def __init__(self):
        self.tail=Node(0)
        self.size=0

    def isEmpty(self):
        return self.size==0
```

```

def append(self,value):
    newNode=Node(value)
    if self.isEmpty():
        self.tail=newNode
        newNode.next=self.tail
    else:
        newNode.next=self.tail.next
        self.tail.next=newNode
        self.tail=newNode
    self.size+=1

def popFront(self):
    if self.isEmpty():
        return None
    old_head=self.tail.next
    if self.size==1:
        self.tail=None
    else:
        self.tail.next=old_head.next
    self.size-=1
    return old_head.value

n,k=map(int,input().split())
ans=[]
linkedList=CircleLinkedList()
for i in range(1,n+1):
    linkedList.append(i)

for _ in range(k-1):
    linkedList.append(linkedList.popFront())

while linkedList.size>1:
    ans.append(linkedList.popFront())
    for _ in range(k-1):
        linkedList.append(linkedList.popFront())

print(*ans)

```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

状态: Accepted

源代码

```
class Node:
    def __init__(self, value, next=None):
        self.value=value
        self.next=next

class CircleLinkedList:
    def __init__(self):
        self.tail=Node(0)
        self.size=0

    def isEmpty(self):
        return self.size==0

    def append(self, value):
        newNode=Node(value)
        if self.isEmpty():
            self.tail=newNode
            newNode.next=self.tail
        else:
            newNode.next=self.tail.next
```

基本信息

#: 48803031  
题目: 05344  
提交人: 24n2400016635  
内存: 3680kB  
时间: 23ms  
语言: Python3  
提交时间: 2025-04-02 18:47:22

## M02774: 木材加工

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

思路:

这道题又是二分加贪心，不过这个检查 mid 值是否合理的函数部分比较好写，每一段木材对于 mid 值能切割的个数就是  $\text{length}/\text{mid}$ ，然后比较最后所有的段数是否大于 k 即可。

代码:

```
### B ###
def cancutsuch(mid,s,k):
    ans=0
    for i in s:
        ans+=i//mid
    return ans>=k

n,k=map(int,input().split())
s=[]
for _ in range(n):
    s.append(int(input()))

l,r=1,sum(s)//k
while l<=r:
    mid=(l+r)//2
    if mid==0: r=0;break
    if cancutsuch(mid,s,k):
        l=mid+1
    else:
        r=mid-1
print(r)
```

代码运行截图 &lt;mark&gt;（至少包含有"Accepted"）&lt;/mark&gt;

状态: **Accepted**

源代码

```
### B ###
def cancutsuch(mid, s, k):
    ans=0
    for i in s:
        ans+=i//mid
    return ans>=k

n, k=map(int, input().split())
s=[]
for _ in range(n):
    s.append(int(input()))

l, r=1, sum(s)//k
while l<=r:
    mid=(l+r)//2
    if mid==0: r=0; break
    if cancutsuch(mid, s, k):
        l=mid+1
    else:
        r=mid-1
print(r)
```

基本信息

#: 48803080  
题目: 02774  
提交人: 24n2400016635  
内存: 3920kB  
时间: 42ms  
语言: Python3  
提交时间: 2025-04-02 18:50:17

## M07161:森林的带度数层次序列存储

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

思路:

第一遍读题看成前序遍历了用 **stack** 做，然后发现问题改成了 **bfs** 的思路，反而简单了，第一次读入第一个节点，然后 **for** 循环一次其孩子数量；接着对各个孩子再 **bfs** 一遍，用这个思路建立树。然后后序遍历就先对该节点的各个孩子递归一下，然后再取该节点的值。

代码:

```
### C ###
from collections import deque

class TreeNode:
    def __init__(self, val, num=0, child=None):
        self.val=val
        self.num=num
        self.child=child if child!=None else []

def build(s):
    if not s:
        return None
    q=deque()
    n=len(s)
    i=0
    root, child_num=TreeNode(s[i]), int(s[i+2])
    root.num=child_num
    q.append(root)

    while q:
        for _ in range(len(q)):
```

```

        curr=q.popleft()
        for _ in range(curr.num):
            i+=4
            node,child_num = TreeNode(s[i]),int(s[i+2])
            node.num=child_num
            curr.child.append(node)
            q.append(node)

    return root

def post(root,ans):
    if root:
        for c in root.child:
            post(c,ans)
        ans.append(root.val)
    return ans

all=[]
n=int(input())
for _ in range(n):
    s=input()
    root=build(s)
    ans=[]
    ans=post(root,ans)
    # print(*ans)
    all.extend(ans)
print(*all)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

#48803123提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

### C ###
from collections import deque

class TreeNode:
    def __init__(self, val, num=0, child=None):
        self.val=val
        self.num=num
        self.child=child if child!=None else []

def build(s):
    if not s:
        return None
    q=deque()
    n=len(s)
    i=0
    root, child_num=TreeNode(s[i]),int(s[i+2])
    root.num=child_num
    q.append(root)

    while q:
        for _ in range(len(q)):

```

基本信息

#: 48803123  
 题目: 07161  
 提交人: 24n2400016635  
 内存: 3680kB  
 时间: 21ms  
 语言: Python3  
 提交时间: 2025-04-02 18:53:16

M18156:寻找离目标数最近的两数之和

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

思路：

记忆里以前好像做过但是考场上一点没想起来，，于是现场想到的是：在双指针移动的过程中，对最终答案 **ans** 和差值 **dif** 进行维护（这个灵感来源于树的遍历，因为常常会遇到这样的题：框架是树的遍历，但是会在遍历过程中对一些变量的值进行更新）。比较恶心的地方是在有多个答案的时候取较小值，于是多了一个 **dif** 是否相等的判断来对 **ans** 取较小值。。总之这段代码思路一般，写起来很丑陋，**debug** 也花了很长时间。。。

代码：

```
### D ###
def do(s):
    global dif,ans

    i, j = 0, len(s) - 1
    while i < j: #3 3 7 8 9
        if s[i] + s[j] == t:
            return t
        elif (curr:=(s[i] + s[j])) > t:
            if abs(curr-t)<dif:
                dif=abs(curr-t)
                ans=curr
            elif abs(curr-t)==dif:
                ans=min(curr,ans)
            j -= 1
        elif (curr := (s[i] + s[j])) < t:
            if abs(curr-t)<dif:
                dif=abs(curr-t)
                ans=curr
            elif abs(curr-t)==dif:
                ans=min(curr,ans)
            i += 1
    return ans

t=int(input())
s=list(map(int,input().split()))

s.sort()
ans=float('inf')
dif=float('inf')
print(do(s))
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
### D ###
def do(s):
    global dif, ans

    i, j = 0, len(s) - 1
    while i < j: #3 3 7 8 9
        if s[i] + s[j] == t:
            return t
        elif (curr := (s[i] + s[j])) > t:
            if abs(curr - t) < dif:
                dif = abs(curr - t)
                ans = curr
            elif abs(curr - t) == dif:
                ans = min(curr, ans)
            j -= 1
        elif (curr := (s[i] + s[j])) < t:
            if abs(curr - t) < dif:
                dif = abs(curr - t)
                ans = curr
            elif abs(curr - t) == dif:
                ans = min(curr, ans)
            i += 1
```

基本信息

#: 48803167  
题目: M18156  
提交人: 24n2400016635  
内存: 15260kB  
时间: 109ms  
语言: Python3  
提交时间: 2025-04-02 18:56:18

## M18159:个位为 1 的质数个数

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

思路:

欧拉筛宣传片（）但是考场上还真忘了欧拉筛怎么打，换了另一个记得的筛法（好像是叫埃拉托斯特尼筛法？）

代码:

```
def chose(n):
    prime=[]
    isprime=[True]*(n+1)
    isprime[0]=isprime[1]=False

    for i in range(2,n+1):
        if isprime[i]:
            prime.append(i)
            for k in range(i**2,n+1,i):
                isprime[k]=False
    return prime

p=chose(10001)
case=0
for _ in range(int(input())):
    case+=1
    s=int(input())
    print(f'Case{case}:')
    ans=[]
    for i in p:
        if i<s and str(i)[-1]=='1':
            ans.append(i)
    if ans: print(*ans)
```

```
else:
    print('NULL')
```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

#48803197提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
def chose(n):
    prime=[]
    isprime=[True]*(n+1)
    isprime[0]=isprime[1]=False

    for i in range(2,n+1):
        if isprime[i]:
            prime.append(i)
            for k in range(i**2,n+1,i):
                isprime[k]=False
    return prime

p=chose(10001)
case=0
for _ in range(int(input())):
    case+=1
    s=int(input())
    print(f'Case {case}:')
    ans=[]
    for i in p:
        if i<s and str(i)[-1]=='1':
            ans.append(i)
    if ans: print(*ans)
    else:
        print('NULL')
```

基本信息

#: 48803197  
题目: 18159  
提交人: 24n2400016635  
内存: 11328kB  
时间: 2463ms  
语言: Python3  
提交时间: 2025-04-02 18:59:56

## M28127:北大夺冠

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

思路:

`defaultdict` 和 `lambda` 排序 的魅力不用多说。。。用 `sub_cnt` 来存每个队伍提交次数, 用 `ac_cnt` 来存每个队伍 ac 了的题目编号, 这里采用了 `defaultdict(set)` 方便一点, 最后只需要取一下 `len` 就知道 ac 了多少题目了。关键是最后的 `lambda` 函数的排序。

代码:

```
from collections import defaultdict
m=int(input())
sub_cnt,ac_cnt=defaultdict(int),defaultdict(set)
names=set()
for _ in range(m):
    name,id,ac=input().split(',')
    names.add(name)
    sub_cnt[name]+=1
    if ac=='yes' and id not in ac_cnt[name]:
        ac_cnt[name].add(id)

ss=[]
for n in names:
    ss.append((len(ac_cnt[n]),sub_cnt[n],n))
```



```
ss.sort(key=lambda x:(-x[0],x[1],x[2]))
for i in range(min(12,len(names))):
    print(i+1,ss[i][2],ss[i][0],ss[i][1])
```

代码运行截图 == (AC 代码截图, 至少包含有"Accepted") ==

#48803227提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
from collections import defaultdict
m=int(input())
sub_cnt,ac_cnt=defaultdict(int),defaultdict(set)
names=set()
for _ in range(m):
    name,id,ac=input().split(',')
    names.add(name)
    sub_cnt[name]+=1
    if ac=='yes' and id not in ac_cnt[name]:
        ac_cnt[name].add(id)

ss=[]
for n in names:
    ss.append((len(ac_cnt[n]),sub_cnt[n],n))

ss.sort(key=lambda x:(-x[0],x[1],x[2]))
for i in range(min(12,len(names))):
    print(i+1,ss[i][2],ss[i][0],ss[i][1])
```

基本信息

#: 48803227  
题目: M28127  
提交人: 24n2400016635  
内存: 3624kB  
时间: 21ms  
语言: Python3  
提交时间: 2025-04-02 19:03:00

## 2. 学习总结和收获

<mark>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。</mark>

本次做题好几道都 debug 半天, 估计是感冒的原因, 比如二分那题本来应该是

```
l,r=1,sum(s)//k
while l<=r:
    mid=(l+r)//2
    if mid==0: r=0;break
    if cancutsuch(mid,s,k):
        l=mid+1
    else:
        r=mid-1
```

被我硬生生写成了

```
l,r=1,sum(s)//k
while l<=r:
    mid=(l+r)//2
    if mid==0: r=0;break
    if cancutsuch(mid,s,k):
        l+=1
    else:
        r-=1
```

l 和 r 的每次都加 1 减 1, 等于没二分, 就这一点点的错误我在考试最后才发现, 服啦!

最近仍然在补每日选做，然后在跟进树的课件，后面的 **avl** 树在寒假时候没看懂，现在要加把劲学了！