

Assignment #C: 202505114 Mock Exam

Updated 1518 GMT+8 May 14, 2025

2025 spring, Compiled by 任宇桐 物理学院

说明:

1. **月考:** AK (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

E06364: 牛的选举

<http://cs101.openjudge.cn/practice/06364/>

思路:

直接排序即可。

代码:

```
n, k = map(int, input().split())
votes = []
for i in range(n):
    a, b = map(int, input().split())
    votes.append((a, b, i))
votes.sort(reverse=True)
votes = votes[:k]
votes.sort(key=lambda x: x[1])
print(votes[-1][-1]+1)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n, k = map(int, input().split())
votes = []
for i in range(n):
    a, b = map(int, input().split())
    votes.append((a, b, i))
votes.sort(reverse=True)
votes = votes[:k]
votes.sort(key = lambda x: x[1])
print(votes[-1][-1]+1)
```

基本信息

#: 49159305
题目: E06364
提交人: 24n2400011498
内存: 12080kB
时间: 150ms
语言: Python3
提交时间: 2025-05-14 15:11:20

M04077: 出栈序列统计

<http://cs101.openjudge.cn/practice/04077/>

思路:

果然考试的时候想不到数学方法。。。只能乖乖按照题目提示模拟了。

代码:

```
n = int(input())
s = [i for i in range(n)]
stack = []
output = []
cnt = 0
def dfs():
    global cnt
    if len(output) == n:
        cnt += 1
        return
    if s:
        stack.append(s.pop())
        dfs()
        s.append(stack.pop())
    if stack:
        output.append(stack.pop())
        dfs()
        stack.append(output.pop())
dfs()
print(cnt)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n = int(input())
s = [i for i in range(n)]
stack = []
output = []
cnt = 0
def dfs():
    global cnt
    if len(output) == n:
        cnt += 1
        return
    if s:
        stack.append(s.pop())
        dfs()
        s.append(stack.pop())
    if stack:
        output.append(stack.pop())
        dfs()
        stack.append(output.pop())
dfs()
print(cnt)
```

基本信息

#: 49159440
题目: M04077
提交人: 24n2400011498
内存: 3704kB
时间: 361ms
语言: Python3
提交时间: 2025-05-14 15:19:17

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

M05343:用队列对扑克牌排序

<http://cs101.openjudge.cn/practice/05343/>

思路:

直接模拟即可。考试的时候没有注意到f-string的嵌套引号要用不同的引号做区分，导致错的莫名其妙

...

代码:

```
n = int(input())
from collections import deque
qs = [deque([]) for i in range(9)]
q_alpha = {'A': deque([]), 'B': deque([]), 'C': deque([]), 'D': deque([])}
ans = []
s = list(input().split())
for i in range(n):
    qs[int(s[i][-1])-1].append(s[i])
for i in range(9):
    print(f'Queue{i+1}:{' ".join(qs[i])}')
    for _ in range(len(qs[i])):
        char = qs[i].popleft()
        q_alpha[char[0]].append(char)
for x in ['A', 'B', 'C', 'D']:
    print(f'Queue{x}:{' ".join(q_alpha[x])}')
    for _ in range(len(q_alpha[x])):
        ans.append(q_alpha[x].popleft())
print(*ans)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n = int(input())
from collections import deque
qs = [deque([]) for i in range(9)]
q_alpha = {'A':deque([]), 'B':deque([]), 'C':deque([]), 'D':deque([])}
ans = []
s = list(input().split())
for i in range(n):
    qs[int(s[i][-1]-1)].append(s[i])
for i in range(9):
    print(f'Queue{i+1}:{" ".join(qs[i])}')
    for _ in range(len(qs[i])):
        char = qs[i].popleft()
        q_alpha[char[0]].append(char)
for x in ['A', 'B', 'C', 'D']:
    print(f'Queue{x}:{" ".join(q_alpha[x])}')
    for _ in range(len(q_alpha[x])):
        ans.append(q_alpha[x].popleft())
print(*ans)
```

基本信息

#: 49162483
题目: 05343
提交人: 24n2400011498
内存: 3644kB
时间: 22ms
语言: Python3
提交时间: 2025-05-14 17:45:53

M04084: 拓扑排序

<http://cs101.openjudge.cn/practice/04084/>

思路:

一开始没能正确理解题意，没看出来谁是起点，谁是终点。看懂以后发现直接每次入队以后排序就行？

代码:

```
v, a = map(int, input().split())
degrees = [0]*v
nodes = [[] for i in range(v)]
for _ in range(a):
    p, q = map(int, input().split())
    p -= 1
    q -= 1
    degrees[q] += 1
    nodes[p].append(q)
ans = []

q = []
for i in range(v):
    if degrees[i] == 0:
        q.append(i)
q.sort(reverse = True)
while q:
    node = q.pop()
    ans.append(node)
    nodes[node].sort()
    for neighbor in nodes[node]:
        degrees[neighbor] -= 1
        if degrees[neighbor] == 0:
            q.append(neighbor)
    q.sort(reverse = True)

temp = []
```

```

for i in ans:
    temp.append(f'v{i+1}')
print(*temp)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

v, a = map(int, input().split())
degrees = [0]*v
nodes = [[] for i in range(v)]
for _ in range(a):
    p, q = map(int, input().split())
    p -= 1
    q -= 1
    degrees[q] += 1
    nodes[p].append(q)
ans = []

q = []
for i in range(v):
    if degrees[i] == 0:
        q.append(i)
q.sort(reverse = True)
while q:
    node = q.pop()
    ans.append(node)
    nodes[node].sort()
    for neighbor in nodes[node]:
        degrees[neighbor] -= 1
        if degrees[neighbor] == 0:
            q.append(neighbor)
q.sort(reverse = True)

```

基本信息

#: 49159752

题目: M04084

提交人: 24n2400011498

内存: 3676kB

时间: 21ms

语言: Python3

提交时间: 2025-05-14 15:36:26

M07735:道路

Dijkstra, <http://cs101.openjudge.cn/practice/07735/>

思路:

开始的时候非常纠结应该如何处理双重约束的问题, 后来决定直接穷举+剪枝, 然后就直接过了?

代码:

```

from collections import defaultdict, deque
k = int(input())
n = int(input())
r = int(input())
information = defaultdict(list)
current_information = [[] for _ in range(n+1)]
for _ in range(r):
    s, d, l, t = map(int, input().split())
    information[s].append((d, l, t))
q = deque([(1, 0, 0)])
while q:
    start, length, tax = q.popleft()
    for d, l, t in information[start]:
        if tax+t <= k:
            if current_information[d]:
                for current_length, current_cost in current_information[d]:
                    if current_cost <= tax+t and current_length <= length+l:

```

```

        break
    else:
        current_information[d].append((length+1, tax+t))
        q.append((d, length+1, tax+t))
    else:
        current_information[d].append((length + 1, tax + t))
        q.append((d, length + 1, tax + t))

temp = current_information[-1][:]
temp.sort()
for i in range(len(temp)):
    if temp[i][1] <= k:
        print(temp[i][0])
        break

```

代码运行截图 (至少包含有"Accepted")

#49160601提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

from collections import defaultdict, deque
k = int(input())
n = int(input())
r = int(input())
information = defaultdict(list)
current_information = [[] for _ in range(n+1)]
for _ in range(r):
    s, d, l, t = map(int, input().split())
    information[s].append((d, l, t))
q = deque([(1, 0, 0)])
while q:
    start, length, tax = q.popleft()
    for d, l, t in information[start]:
        if tax+t <= k:
            if current_information[d]:
                for current_length, current_cost in current_information[d]:
                    if current_cost <= tax+t and current_length <= length:
                        break
            else:
                current_information[d].append((length+1, tax+t))
                q.append((d, length+1, tax+t))

```

基本信息

: 49160601
题目: M07735
提交人: 24n2400011498
内存: 4596kB
时间: 265ms
语言: Python3
提交时间: 2025-05-14 16:21:55

T24637:宝藏二叉树

dp, <http://cs101.openjudge.cn/practice/24637/>

思路:

虽然是tough题目, 但是感觉就是套了一个树的壳子, 比上学期的dp简单不少。。。

代码:

```
n = int(input())
s = list(map(int, input().split()))
maxv = 0
temp = 0
def dfs(i, flag):
    if i >= n:
        return 0
    if flag == 0:
        return max(dfs(2*i+1, 1), dfs(2*i+1, 0))+max(dfs(2*i+2, 1),dfs(2*i+2, 0))
    if flag == 1:
        return s[i]+dfs(2*i+1, 0)+dfs(2*i+2, 0)
print(max(dfs(0, 0), dfs(0, 1)))
```

代码运行截图 (至少包含有"Accepted")

#49161248提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n = int(input())
s = list(map(int, input().split()))
maxv = 0
temp = 0
def dfs(i, flag):
    if i >= n:
        return 0
    if flag == 0:
        return max(dfs(2*i+1, 1), dfs(2*i+1, 0))+max(dfs(2*i+2, 1),dfs(
    if flag == 1:
        return s[i]+dfs(2*i+1, 0)+dfs(2*i+2, 0)
print(max(dfs(0, 0), dfs(0, 1)))
```

基本信息

#: 49161248
题目: T24637
提交人: 24n2400011498
内存: 3648kB
时间: 74ms
语言: Python3
提交时间: 2025-05-14 16:46:35

©2002-2022 POJ 京ICP备20010980号-1 [English](#) [帮助](#) [关于](#)

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。

第一次成功AK，还是蛮有成就感的。但是感觉整体题目思维量还是相对小，造成的假象？