

# Assignment #C: 202505114 Mock Exam

Updated 1518 GMT+8 May 14, 2025

2025 spring, Compiled by 颜鼎堃 工学院

## 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

### E06364: 牛的选举

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

思路:

- 排序

代码:

```
1 N, K = map(int, input().split())
2 ticket = [list(map(int, input().split())) + [i] for i in range(N)]
3 r2 = sorted(ticket, reverse=True)[:K]
4 print(sorted(r2, key=lambda t: t[1])[-1][2] + 1)
```

Python

代码运行截图 (至少包含有“Accepted”)



## #49159391提交状态

[查看](#)[提交](#)[统计](#)[提问](#)状态: **Accepted**

源代码

```
N, K = map(int, input().split())
ticket = [list(map(int, input().split())) + [i] for i in range(N)]
r2 = sorted(ticket, reverse=True)[:K]
print(sorted(r2, key=lambda t: t[1][-1][2] + 1))
```

基本信息

#: 49159391

题目: E06364

提交人: 颜鼎盛(24n2400011125)

内存: 14200kB

时间: 187ms

语言: Python3

提交时间: 2025-05-14 15:16:03

OJ-06364

## M04077: 出栈序列统计

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

思路:

- 卡特兰数

代码:

```
1 from math import comb
2 n = int(input())
3 print(comb(2*n, n) - comb(2*n, n - 1))
```

Python

代码运行截图 (至少包含有"Accepted")



## #49159506提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```
from math import comb
n = int(input())
print(comb(2*n, n) - comb(2*n, n - 1))
```

基本信息

#: 49159506

题目: M04077

提交人: 颜鼎盛(24n2400011125)

内存: 3544kB

时间: 20ms

语言: Python3

提交时间: 2025-05-14 15:22:19

OJ-04077

## M05343:用队列对扑克牌排序

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

思路:

- 队列模拟
- 考场上改了半天居然是测试数据没删

代码:

```
1  from collections import deque
2  qy = [deque() for i in range(10)]
3  d1 = "xABCD"
4  d2 = {"A":1, "B":2, "C":3, "D":4}
5  qx = [deque() for i in range(5)]
6  n = int(input())
7  for i in input().split():
8      qy[int(i[1])].append(i)
9  for i in range(1, 10):
10     print(f"Queue{i}", end=":")
11     print(*qy[i])
12     while qy[i]:
13         p = qy[i].popleft()
14         qx[d2[p[0]]].append(p)
15  ans = deque()
16  for i in range(1, 5):
17     print(f"Queue{d1[i]}", end=":")
18     print(*qx[i])
19     while qx[i]:
20         ans.append(qx[i].popleft())
21  print(*ans)
```

Python

代码运行截图 (至少包含有"Accepted")



## #49160398提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```
from collections import deque
qy = [deque() for i in range(10)]
d1 = "xABCD"
d2 = {"A":1, "B":2, "C":3, "D":4}
qx = [deque() for i in range(5)]
n = int(input())
for i in input().split():
    qy[int(i[1])].append(i)
for i in range(1, 10):
    print(f"Queue({i})", end=":")
    print(*qy[i])
    while qy[i]:
        p = qy[i].popleft()
        qx[d2[p[0]]].append(p)
ans = deque()
for i in range(1, 5):
    print(f"Queue({d1[i]})", end=":")
    print(*qx[i])
    while qx[i]:
        ans.append(qx[i].popleft())
print(*ans)
```

基本信息

#: 49160398

题目: M05343

提交人: 颜鼎堃(24n2400011125)

内存: 3652kB

时间: 24ms

语言: Python3

提交时间: 2025-05-14 16:11:46

OJ-05343

## M04084: 拓扑排序

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

思路:

- 实名讨厌重边
- 考场上知道思路但是写太慢，耽误太多时间，最后还没写出来

代码:

```
1  from heapq import heappush, heappop
2  class Vertex:
3      def __init__(self, n):
4          self.num = n
5          self.name = f"v{n + 1}"
6          self.ind = 0
7  v, a = map(int, input().split())
8  ans = []
9  heap = []
10 adj_mat = [[False for i in range(v)] for j in range(v)]
11 nodes = [Vertex(i) for i in range(v)]
12 for i in range(a):
13     n1, n2 = map(int, input().split())
14     if not adj_mat[n1 - 1][n2 - 1]:
15         nodes[n2 - 1].ind += 1
16         adj_mat[n1 - 1][n2 - 1] = True
17 for i in nodes:
18     if not i.ind:
19         heappush(heap, i.num)
20 while heap:
21     node = heappop(heap)
22     for i in range(v):
23         if adj_mat[node][i]:
24             nodes[i].ind -= 1
25             if nodes[i].ind == 0:
```

Python

```

26         heappush(heap, i)
27     ans.append(nodes[node].name)
28
29     print(*ans)

```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#49176848提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

from heapq import heappush, heappop
class Vertex:
    def __init__(self, n):
        self.num = n
        self.name = f"v{n + 1}"
        self.ind = 0
v, a = map(int, input().split())
ans = []
heap = []
adj_mat = [[False for i in range(v)] for j in range(v)]
nodes = [Vertex(i) for i in range(v)]
for i in range(a):
    n1, n2 = map(int, input().split())
    if not adj_mat[n1 - 1][n2 - 1]:
        nodes[n2 - 1].ind += 1
        adj_mat[n1 - 1][n2 - 1] = True
for i in nodes:
    if not i.ind:
        heappush(heap, i.num)
while heap:
    node = heappop(heap)
    for i in range(v):
        if adj_mat[node][i]:
            nodes[i].ind -= 1
            if nodes[i].ind == 0:
                heappush(heap, i)
    ans.append(nodes[node].name)
print(*ans)

```

基本信息

#: 49176848

题目: 04084

提交人: 颜鼎堃(24n2400011125)

内存: 3724kB

时间: 21ms

语言: Python3

提交时间: 2025-05-16 14:18:39

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

OJ-04084

## M07735:道路

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

思路:

- 开二维数组, 确实没想到

代码:

```

1  from heapq import heappush, heappop
2  from collections import defaultdict
3  K = int(input())
4  N = int(input())
5  R = int(input())
6  dist = defaultdict(list)
7  length = [[1e9 for j in range(K + 1)] for i in range(N + 1)]
8  length[1][0] = 0
9  for i in range(R):
10     S, D, L, T = map(int, input().split())
11     dist[S].append((L, T, D))
12  heap = [(0, 0, 1)]
13  ans = -1
14  while heap:
15     distance, cost, node = heappop(heap)
16     if node == N:
17         ans = distance
18         break
19     if distance > length[node][cost]:
20         continue
21     for l, c, n in dist[node]:

```

Python

```

22     if c + cost ≤ K and l + distance < length[n][c + cost]:
23         length[n][c + cost] = l + distance
24         heappush(heap, (length[n][c + cost], c + cost, n))
25
26 print(ans)

```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号

CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#49219454提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

from heapq import heappush, heappop
from collections import defaultdict
K = int(input())
N = int(input())
R = int(input())
dist = defaultdict(list)
length = [[1e9 for j in range(K + 1)] for i in range(N + 1)]
length[1][0] = 0
for i in range(R):
    S, D, L, T = map(int, input().split())
    dist[S].append((L, T, D))
heap = [(0, 0, 1)]
ans = -1
while heap:
    distance, cost, node = heappop(heap)
    if node == N:
        ans = distance
        break
    if distance > length[node][cost]:
        continue
    for l, c, n in dist[node]:
        if c + cost ≤ K and l + distance < length[n][c + cost]:
            length[n][c + cost] = l + distance
            heappush(heap, (length[n][c + cost], c + cost, n))
print(ans)

```

基本信息

#: 49219454

题目: 07735

提交人: 颜鼎堃(24n2400011125)

内存: 5444kB

时间: 49ms

语言: Python3

提交时间: 2025-05-20 22:48:53

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

OJ-07735

## T24637:宝藏二叉树

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

思路:

- 树上的动态规划

代码:

```

1  from math import log2, ceil
2  N = int(input())
3  layer = ceil(log2(N + 1)) - 1
4  treasure = list(map(int, input().split()))
5  dp = [[0 for i in range(2**layer)] for j in range(2)]
6  for i in range(2**layer - 1, N):
7      dp[1][i] = treasure[i]
8      dp[0][i] = 0
9  for i in range(2**layer - 2, -1, -1):
10     dp[0][i] = max(dp[0][2*i+1], dp[1][2*i+1]) + max(dp[0][2*i+2], dp[1][2*i+2])
11     dp[1][i] = treasure[i] + dp[0][2 * i + 1] + dp[0][2 * i + 2]
12 print(max(dp[0][0], dp[1][0]))

```

Python

代码运行截图 (至少包含有"Accepted")



#49162339提交状态

查看

提交

统计

提问

状态: Accepted

源代码

```
from math import log2, ceil
N = int(input())
layer = ceil(log2(N + 1)) - 1
treasure = list(map(int, input().split()))
dp = [[0 for i in range(2**(layer+1))] for j in range(2)]
for i in range(2**layer - 1, N):
    dp[1][i] = treasure[i]
    dp[0][i] = 0
for i in range(2**layer - 2, -1, -1):
    dp[0][i] = max(dp[0][2*i+1], dp[1][2*i+1]) + max(dp[0][2*i+2], dp[1][2*i+2])
    dp[1][i] = treasure[i] + dp[0][2 * i + 1] + dp[0][2 * i + 2]
print(max(dp[0][0], dp[1][0]))
```

基本信息

#: 49162339

题目: 24637

提交人: 颜鼎堃(24n2400011125)

内存: 3664kB

时间: 28ms

语言: Python3

提交时间: 2025-05-14 17:34:15

OJ-24637

## 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。

考试本来以为能做对五题吧，最后只做对了三题。在各种细节上花费大量时间调整。第三题知道思路，但总是会在细微的地方出bug，以及没考虑重边。最后一题本来都写出来了，一个下标标错了，调不出来。

熟练度不够

这几天忙飞了