

Assignment #C: 202505114 Mock Exam

Updated 1518 GMT+8 May 14, 2025

2025 spring, Compiled by 郑涵予 物理学院

说明:

1. **月考**: AC6。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:

读入后先按照第一次的票数排序，取前k个，然后找到第二次票数最大的就行。(用时约3min)

代码:

```
n,k=map(int,input().split())
a=[]
for _ in range(n):
    x,y=map(int,input().split())
    a.append((-x,y,_+1))
a.sort()
pos=0
temp=-1
for i in range(k):
    if a[i][1]>temp:
        temp=a[i][1]
        pos=a[i][2]
print(pos)
```

代码运行截图 (至少包含有"Accepted")

#49159304提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n,k=map(int,input().split())
a=[]
for _ in range(n):
    x,y=map(int,input().split())
    a.append((-x,y,_+1))
a.sort()
pos=0
temp=-1
for i in range(k):
    if a[i][1]>temp:
        temp=a[i][1]
        pos=a[i][2]
print(pos)
```

基本信息

#: 49159304
题目: E06364
提交人: 24n2400011325
内存: 13164kB
时间: 150ms
语言: Python3
提交时间: 2025-05-14 15:11:16

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M04077: 出栈序列统计

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

思路:

按照提示直接搜索就行, 事实上这个是卡特兰数, 可以直接输出 $\text{math.comb}(2*n,n)/(n+1)$.(用时约5min)

代码:

```
n=int(input())
ans=0//
def dfs(i,k):
    if i==n:
        global ans
        ans+=1
        return
    dfs(i+1,k+1)
    if k>0:
        dfs(i,k-1)
dfs(0,0)
print(ans)
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
n=int(input())
ans=0
def dfs(i,k):
    if i==n:
        global ans
        ans+=1
        return
    dfs(i+1,k+1)
    if k>0:
        dfs(i,k-1)
dfs(0,0)
print(ans)
```

基本信息

#: 49159438
题目: M04077
提交人: 24n2400011325
内存: 3592kB
时间: 94ms
语言: Python3
提交时间: 2025-05-14 15:19:06

©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:

直接按要求操作就行 (用时约7min)

代码:

```
from collections import deque
q=[deque() for _ in range(9)]
n=int(input())
s=list(input().split())
for x in s:
    num=int(x[1])
    q[num-1].append(x)
temp=[]
for i in range(9):
    print('Queue{:}'.format(i+1),end="")
    print(' '.join(q[i]))
    while q[i]:
        temp.append(q[i].popleft())
for x in temp:
    pos=ord(x[0])-ord('A')
    q[pos].append(x)
ans=[]
l=['A','B','C','D']
for i in range(4):
    ch=l[i]
    print('Queue{:}'.format(ch),end="")
    print(' '.join(q[i]))
    while q[i]:
        ans.append(q[i].popleft())
print(' '.join(ans))
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
from collections import deque
q=[deque() for _ in range(9)]
n=int(input())
s=list(input().split())
for x in s:
    num=int(x[1])
    q[num-1].append(x)
temp=[]
for i in range(9):
    print('Queue{}:'.format(i+1),end=" ")
    print(' '.join(q[i]))
    while q[i]:
        temp.append(q[i].popleft())
for x in temp:
    pos=ord(x[0])-ord('A')
    q[pos].append(x)
ans=[]
l=['A','B','C','D']
for i in range(4):
    ch=l[i]
    print('Queue{}:'.format(i+1),end=" ")
    print(' '.join(q[i]))
    while q[i]:
        ans.append(q[i].popleft())
print(' '.join(ans))
```

基本信息

#: 49159583
题目: M05343
提交人: 24n2400011325
内存: 3648kB
时间: 21ms
语言: Python3
提交时间: 2025-05-14 15:27:16

M04084: 拓扑排序

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

思路:

相比正常的拓扑排序多要求了编号小的先访问，把普通队列改成优先队列就行（用时约5min）

代码:

```
n,m=map(int,input().split())
a=[[ ] for _ in range(n+1)]
indegree=[0]*(n+1)
for _ in range(m):
    u,v=map(int,input().split())
    a[u].append(v)
    indegree[v]+=1
import heapq
q=[]
for i in range(1,n+1):
    if not indegree[i]:
        heapq.heappush(q,i)
ans=[]
while q:
    p=heapq.heappop(q)
    ans.append('v'+str(p))
    for x in a[p]:
        indegree[x]-=1
        if not indegree[x]:
            heapq.heappush(q,x)
print(' '.join(ans))
```

状态: Accepted

源代码

```
n,m=map(int,input().split())
a=[[ ] for _ in range(n+1)]
indegree=[0]*(n+1)
for _ in range(m):
    u,v=map(int,input().split())
    a[u].append(v)
    indegree[v]+=1
import heapq
q=[]
for i in range(1,n+1):
    if not indegree[i]:
        heapq.heappush(q,i)
ans=[]
while q:
    p=heapq.heappop(q)
    ans.append(' '+str(p))
    for x in a[p]:
        indegree[x]-=1
        if not indegree[x]:
            heapq.heappush(q,x)
print(' '.join(ans))
```

基本信息

#: 49159378
题目: M04084
提交人: 24n2400011325
内存: 3648kB
时间: 27ms
语言: Python3
提交时间: 2025-05-14 15:15:17

M07735:道路

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

思路:

感觉这题应该算T了其实,比后面一题难.一开始直接写了暴力的bfs果然MLE了.最后的思路大概是用一个best字典存给定编号与花费下能达到的最小距离,用Dijkstra不断更新,同时花费如果大于k就不加入队列,最后运行起来出奇地快.(用时约25min)

代码:

```
k=int(input())
n=int(input())
m=int(input())
a=[[ ] for _ in range(n+1)]
for _ in range(m):
    u,v,w,c=map(int,input().split())
    a[u].append((v,w,c))
import heapq
best=dict()
q=[]
heapq.heappush(q,(0,0,1))#distance cost index
best[(1,0)]=0
ans=-1
while q:
    dist,cost,index=heapq.heappop(q)
    if index==n:
        ans=dist
        break
    for v,w,c in a[index]:
        ncost=cost+c
        ndist=dist+w
```

```

        if ncost<=k and ndist<best.get((v,ncost),float('inf')):
            best[(v,ncost)]=ndist
            heapq.heappush(q,(ndist,ncost,v))
print(ans)

```

代码运行截图 (至少包含有"Accepted")

#49160081提交状态

[查看](#) [提交](#) [统计](#)

状态: Accepted

源代码

```

k=int(input())
n=int(input())
m=int(input())
a=[[ ] for _ in range(n+1)]
for _ in range(m):
    u,v,w,c=map(int,input().split())
    a[u].append((v,w,c))
import heapq
best=dict()
q=[]
heapq.heappush(q,(0,0,1))#distance cost index
best[(1,0)]=0
ans=-1
while q:
    dist,cost,index=heapq.heappop(q)
    if index==n:
        ans=dist
        break
    for v,w,c in a[index]:
        ncost=cost+c
        ndist=dist+w
        if ncost<=k and ndist<best.get((v,ncost),float('inf')):
            best[(v,ncost)]=ndist
            heapq.heappush(q,(ndist,ncost,v))
print(ans)

```

基本信息

#: 49160081
 题目: M07735
 提交人: 24n2400011325
 内存: 6724kB
 时间: 56ms
 语言: Python3
 提交时间: 2025-05-14 15:56:38

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#)

T24637:宝藏二叉树

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

思路:

直接dp就行, 但是考场上写法有些诡异, 在两个函数之间来回递归调用, 没想到还一次就过了.后面发现其实可以直接逆序递推就行的.(用时约5min)

代码:

```

n=int(input())
a=list(map(int,input().split()))
a.insert(0,0)
from functools import lru_cache
@lru_cache(maxsize=None)
def dp0(i):
    if i>n:
        return 0
    return
max(dp1(2*i)+dp1(2*i+1),dp0(2*i)+dp0(2*i+1),dp0(2*i)+dp1(2*i+1),dp1(2*i)+dp0(2*i+1))
@lru_cache(maxsize=None)
def dp1(i):
    if i>n:

```

```

        return 0
    return dp0(2*i)+dp0(2*i+1)+a[i]
ans=0
for i in range(1,n+1):
    ans=max(ans,dp0(i),dp1(i))
print(ans)

```

代码运行截图 (至少包含有"Accepted")

#49159895提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

n=int(input())
a=list(map(int,input().split()))
a.insert(0,0)
from functools import lru_cache
@lru_cache(maxsize=None)
def dp0(i):
    if i>n:
        return 0
    return max(dp1(2*i)+dp1(2*i+1),dp0(2*i)+dp0(2*i+1),dp0(2*i)+dp1(2*i+1))
@lru_cache(maxsize=None)
def dp1(i):
    if i>n:
        return 0
    return dp0(2*i)+dp0(2*i+1)+a[i]
ans=0
for i in range(1,n+1):
    ans=max(ans,dp0(i),dp1(i))
print(ans)

```

基本信息

#: 49159895
 题目: T24637
 提交人: 24n2400011325
 内存: 3924kB
 时间: 22ms
 语言: Python3
 提交时间: 2025-05-14 15:44:27

©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。

这次月考还是比较顺利 (希望期末机考也这样) .但是对机房的环境似乎还是有点不熟, 第五题一开始MLE后心急之下想用C++试试, 结果发现机房没有Visual Studio,只有我不熟悉的VS code和dev C++, 连怎么创建C++文件都没搞懂, 最后只好作罢.好在最后用python写也过了.