Assignment #3: 惊蛰 Mock Exam

Updated 1641 GMT+8 Mar 5, 2025

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

说明：

1. **惊蛰月考：**<mark>AC5</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

E04015: 邮箱验证

strings, http://cs101.openjudge.cn/practice/04015

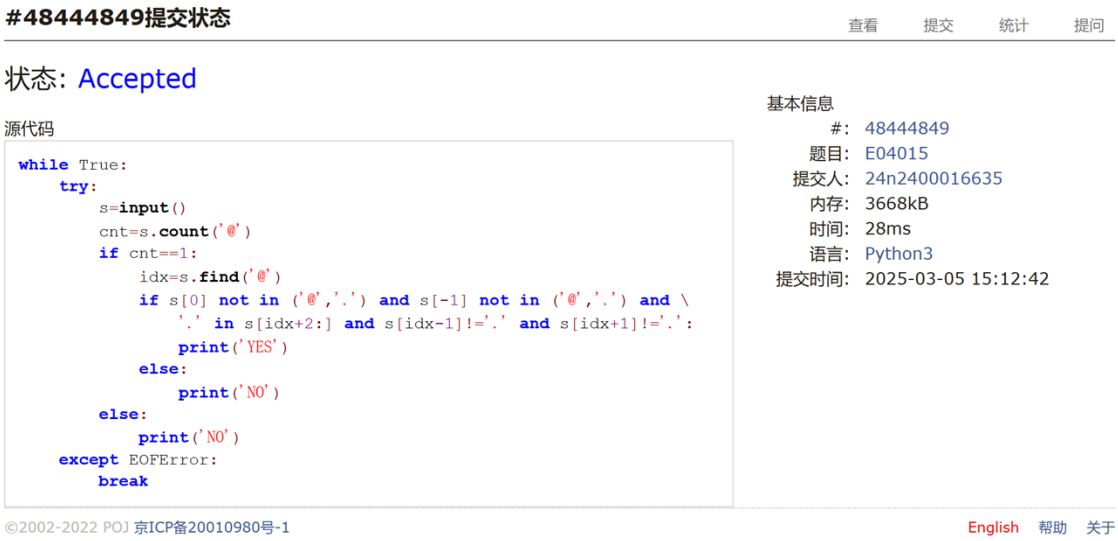
思路：
计概做过，这道题只需要按照题目要求列出 if 条件判断即可，有一个坑是.有可能在@的恰好前一位，不要忘记了。

代码：

```
### A ###
while True:
    try:
        s=input()
        cnt=s.count('@')
        if cnt==1:
            idx=s.find('@')
            if s[0] not in ('@','.') and s[-1] not in ('@','.') and \
                '.' in s[idx+2:] and s[idx-1]!='.' and s[idx+1]!='.':
                print('YES')
```

```
            else:
                print('NO')
        else:
            print('NO')
    except EOFError:
        break
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



M02039: 反反复复

implementation, http://cs101.openjudge.cn/practice/02039/

思路：
模拟题，先计算出矩阵的行数 n，然后创建矩阵，按照蛇形方向填充矩阵，方法是对行数 i 进行奇偶判断，如果 i 是偶数就顺着填 j，奇数就逆着填 j。
最后先遍历列再遍历行来得到答案。在代码中添加**打印矩阵**的代码来随时检查是否正确。

代码：

```
### B ###
m=int(input())
s=list(input())
l=len(s)
n=l//m
a=[['.']*m for _ in range(n)]
# for i in a: print(*i)

for i in range(n):
    if not i%2==1:
        for j in range(m):
            a[i][j]=s[i*m+j]
    else:
        for j in range(m-1,-1,-1):
            a[i][j]=s[i*m+(m-1-j)]

# for i in a: print(*i)
```

```
ans=''
for j in range(m):
    for i in range(n):
        ans+=a[i][j]
print(ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

M02092: Grandpa is Famous

implementation, http://cs101.openjudge.cn/practice/02092/

思路：
这道题其实很简单，就是英文题目太长了读起来很费时间，其实就是把输入的数据统计一下频次，然后输出频次第二位的所有编号即可。使用 collections 中的 Counter 函数非常方便。

代码：

```
### C ###
from collections import Counter
while True:
    n,m=map(int,input().split())
    if {n,m}=={0}:
        break
    s=[]
    for i in range(n):
        l=list(map(int,input().split()))
        s.extend(l)
    c=Counter(s)
    ans=[]
    cnts=[v for v in c.values()]
    cnts.sort(reverse=True)
    ans=[]
```

```
    for k,v in c.items():
        if v==cnts[1]:
            ans.append(k)
    ans.sort()
    print(*ans)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

M04133: 垃圾炸弹

matrices, http://cs101.openjudge.cn/practice/04133/

思路：
计概做过，这道题主要在于 range 中的 min 和 max 的使用，保证把垃圾数量填到矩阵中时不越界。设置一个全局遍历 maxv 来更新某个位置的累积最大垃圾数目，最后再遍历一次矩阵得到用于 maxv 数目的格子数量即可。

代码：

```
### D ###
d=int(input())
n=int(input())
s=[[0]*1025 for _ in range(1025)]
maxv=0
for _ in range(n):
    x,y,num=map(int,input().split())
    for i in range(max(0,x-d),min(1025,x+1+d)):
        for j in range(max(0,y-d),min(1025,y+1+d)):
            s[i][j]+=num
            maxv=max(maxv,s[i][j])
cnt=0
for i in range(1025):
    for j in range(1025):
```

```
        if s[i][j]==maxv:
            cnt+=1
print(cnt,maxv)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

T02488: A Knight's Journey

backtracking, http://cs101.openjudge.cn/practice/02488/

思路：
其实就是很简单的 dfs，只不过变形的地方是：可以从任意位置作为起点遍历棋盘，且输出字典序最小的路径。因此可以初始化一个 paths 列表来放所有可能的路径，我们想要字典序从小到大排列，考虑到列编号是字母，所以在遍历起点的时候按照**先遍历列再遍历行**的方式来进行。这样，只要得到了完整的路径，这些路径就是所有可能路径的首字母和首数字最小的路径了（比如 A3），因此可以直接剪枝。
然后对得到的 paths 排序取第一个就是答案。
这道题的重点是遍历顺序和剪枝策略，否则会超时。

代码：

```
### E ###
dx,dy=[-1,1,-2,2,-2,2,-1,1],[-2,-2,-1,-1,1,1,2,2]
def dfs(s,p,q,x,y,path):
    target=p*q
    s[x][y]=1
    xx=str(x+1)
    yy=chr(ord('A')+y)
    path+=(yy+xx)
    if len(path)==target*2:
        paths.append(path)
        return
    for i in range(8):
        nx,ny=x+dx[i],y+dy[i]
        if 0<=nx<p and 0<=ny<q and s[nx][ny]==0:
            s[nx][ny]=1
```

```
            dfs(s,p,q,nx,ny,path)
            s[nx][ny]=0

n=int(input())
for case in range(1,n+1):
    paths=[]
    p,q=map(int,input().split()) #1,2,3...,p; a,b,c,,,,,,q
    s=[[0]*q for _ in range(p)]
    f=True
    for j in range(q):
        if not f:
            break
        for i in range(p):
            dfs(s,p,q,i,j,'')
            if paths:
                f=False
                break

    print(f'Scenario #{case}:')
    if not paths:
        print('impossible')
    else:
        paths.sort()
        print(paths[0])
    print()
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
### E ###
dx,dy=[-1,1,-2,2,-2,2,-1,1],[-2,-2,-1,-1,1,1,2,2]
def dfs(s,p,q,x,y,path):
    target=p*q
    s[x][y]=1
    xx=str(x+1)
    yy=chr(ord('A')+y)
    path+=(yy+xx)
    if len(path)==target*2:
        paths.append(path)
        return
    for i in range(8):
        nx,ny=x+dx[i],y+dy[i]
        if 0<=nx<p and 0<=ny<q and s[nx][ny]==0:
            s[nx][ny]=1
            dfs(s,p,q,nx,ny,path)
            s[nx][ny]=0

n=int(input())
for case in range(1,n+1):
    paths=[]
    p,q=map(int,input().split()) #1,2,3...,p;;;;a,b,c,,,,,,q
    s=[[0]*q for _ in range(p)]
    f=True
    for j in range(q):
        if not f:
            break
        for i in range(p):
            dfs(s,p,q,i,j,'')
            if paths:
                f=False
                break

    print(f'Scenario #{case}:')
    if not paths:
        print('impossible')
    else:
        paths.sort()
        print(paths[0])
    print()
```

基本信息

#:	48447587
题目:	T02488
提交人:	24n2400016635
内存:	3720kB
时间:	7166ms
语言:	Python3
提交时间:	2025-03-05 16:54:18

T06648: Sequence

heap, http://cs101.openjudge.cn/practice/06648/

思路:

考后看的题解，heap 考场上能想到，但是答案好像有点贪心的感觉，而且两两合并的思路可以有效降低时间复杂度，很难想到。设置 visited 的集合是防止存入相同的和，这个地方 debug 几次没看出来。

代码:

```
### F ###
from heapq import heappop,heappush
def merge(a,b,n):
    heap,result,visited=[],[], {(0,0)}
    heappush(heap,(a[0]+b[0],0,0))
    while len(result)<n:
        res,i,j=heappop(heap)
        result.append(res)
        if i<n-1 and (i+1,j) not in visited:
            heappush(heap,(a[i+1]+b[j],i+1,j))
            visited.add((i+1,j))
```

```
        if j<n-1 and (i,j+1) not in visited:
            heappush(heap,(a[i]+b[j+1],i,j+1))
            visited.add((i,j+1))
    return result[:n]


t=int(input())
for _ in range(t):
    m,n=map(int,input().split())
    curr=sorted(list(map(int,input().split())))
    for _ in range(m-1):
        other=sorted(list(map(int,input().split())))
        curr=merge(curr,other,n)
    print(*curr)
```

代码运行截图 ==（AC 代码截图，至少包含有"Accepted"）==

#48455155提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
### F ###
from heapq import heappop,heappush
def merge(a,b,n):
    heap,result,visited=[],[], {(0,0)}
    heappush(heap,(a[0]+b[0],0,0))
    while len(result)<n:
        res,i,j=heappop(heap)
        result.append(res)
        if i<n-1 and (i+1,j) not in visited:
            heappush(heap,(a[i+1]+b[j],i+1,j))
            visited.add((i+1,j))
        if j<n-1 and (i,j+1) not in visited:
            heappush(heap,(a[i]+b[j+1],i,j+1))
            visited.add((i,j+1))
    return result[:n]


t=int(input())
for _ in range(t):
    m,n=map(int,input().split())
    curr=sorted(list(map(int,input().split())))
    for _ in range(m-1):
        other=sorted(list(map(int,input().split())))
        curr=merge(curr,other,n)
    print(*curr)
```

基本信息

#: 48455155
题目: 06648
提交人: 24n2400016635
内存: 5612kB
时间: 1292ms
语言: Python3
提交时间: 2025-03-06 12:38:55

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>

做题感受：第 3 和 5 题都是英文题目，难度不大，读题时间很长。我做到第 3 题看到一堆英文直接跳过去做第 4 题了，然后再回来做第 3 题。然后第 5 题也跳过了先看第 6 题，结果花了 40min 写代码并发现思路是错的，又折返到第 5 题，结果发现第 5 题其实也还好。最后 AC5。感觉这些题目用 c++做起来可能会很复杂吧……希望老师能和上学期一样每次分享一下大佬的 cpp 作业代码！想学习学习别人的写法！

前五题都是考试源码，命名可能没那么讲究，不过足够自己在紧张的情况下反应了。第六题参考了题解。每日选做进度有点落后了，因为其中有些题目难度好像异常大了（对

于我来说），比如 27256 当前队列中位数，在 sunnywhy 中有一个简单版本的，不涉及删除数字：https://sunnywhy.com/sfbj/9/7/370，就比 oj 的那个题好做一些。感觉自己需要多刷题，目前的情况是模板题大概率能做，难题新题一点做不出来。。。