Assignment #3: 惊蛰 Mock Exam

Updated 1641 GMT+8 Mar 5, 2025

2025 spring, Complied by 同学的姓名、院系

> **说明：**
>
> 1. **惊蛰月考**：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

E04015: 邮箱验证

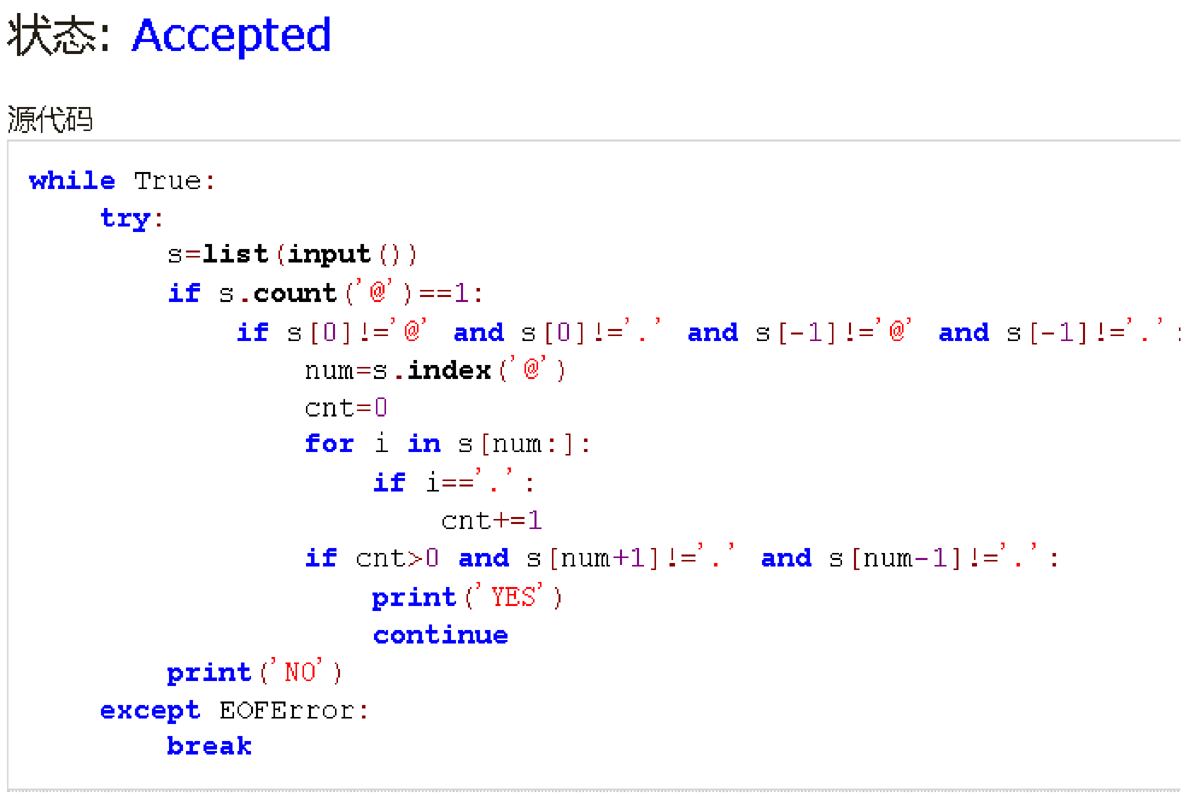
strings, http://cs101.openjudge.cn/practice/04015

思路：啊啊啊考试时把至少一个看成只有一个了浪费了不少时间

代码：

```python
while True:
    try:
        s=list(input())
        if s.count('@')==1:
            if s[0]!='@' and s[0]!='.' and s[-1]!='@' and s[-1]!='.':
                num=s.index('@')
                cnt=0
                for i in s[num:]:
                    if i=='.':
                        cnt+=1
                if cnt>0 and s[num+1]!='.' and s[num-1]!='.':
                    print('YES')
                    continue
        print('NO')
    except EOFError:
        break
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
while True:
    try:
        s=list(input())
        if s.count('@')==1:
            if s[0]!='@' and s[0]!='.' and s[-1]!='@' and s[-1]!='.':
                num=s.index('@')
                cnt=0
                for i in s[num:]:
                    if i=='.':
                        cnt+=1
                if cnt>0 and s[num+1]!='.' and s[num-1]!='.':
                    print('YES')
                    continue
        print('NO')
    except EOFError:
        break
```

M02039: 反反复复

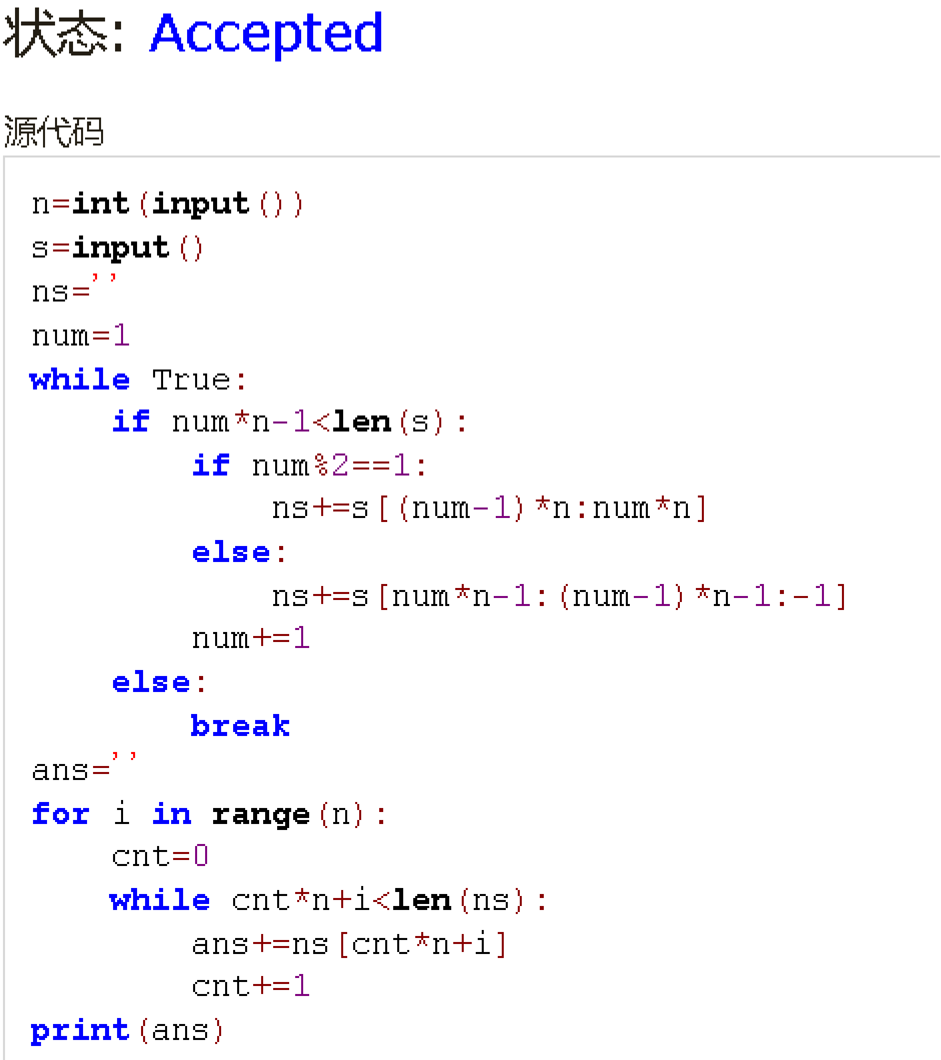
implementation, http://cs101.openjudge.cn/practice/02039/

思路：

代码：

```
n=int(input())
s=input()
ns=''
num=1
while True:
    if num*n-1<len(s):
        if num%2==1:
            ns+=s[(num-1)*n:num*n]
        else:
            ns+=s[num*n-1:(num-1)*n-1:-1]
        num+=1
    else:
        break
ans=''
for i in range(n):
    cnt=0
    while cnt*n+i<len(ns):
        ans+=ns[cnt*n+i]
        cnt+=1
print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
n=int(input())
s=input()
ns=''
num=1
while True:
    if num*n-1<len(s):
        if num%2==1:
            ns+=s[(num-1)*n:num*n]
        else:
            ns+=s[num*n-1:(num-1)*n-1:-1]
        num+=1
    else:
        break
ans=''
for i in range(n):
    cnt=0
    while cnt*n+i<len(ns):
        ans+=ns[cnt*n+i]
        cnt+=1
print(ans)
```

M02092: Grandpa is Famous

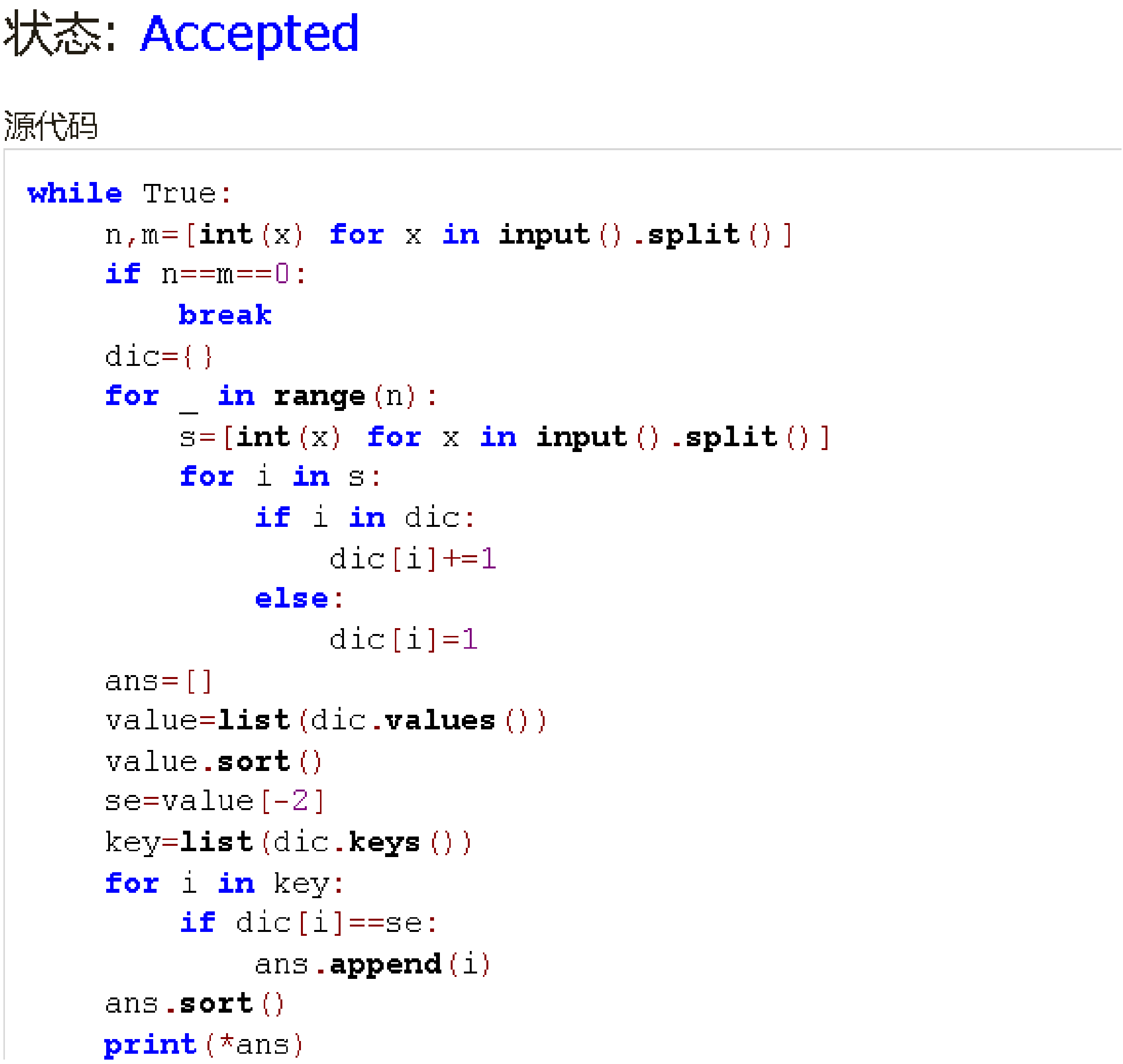
implementation, http://cs101.openjudge.cn/practice/02092/

思路：考试时读题和理解题目意思花了不少时间，可能因为英语太差了吧......但理解题目意思之后就很快做出来了

代码：

```
while True:
    n,m=[int(x) for x in input().split()]
    if n==m==0:
        break
    dic={}
    for _ in range(n):
        s=[int(x) for x in input().split()]
        for i in s:
            if i in dic:
                dic[i]+=1
            else:
                dic[i]=1
    ans=[]
    value=list(dic.values())
    value.sort()
    se=value[-2]
    key=list(dic.keys())
    for i in key:
        if dic[i]==se:
            ans.append(i)
    ans.sort()
    print(*ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
while True:
    n,m=[int(x) for x in input().split()]
    if n==m==0:
        break
    dic={}
    for _ in range(n):
        s=[int(x) for x in input().split()]
        for i in s:
            if i in dic:
                dic[i]+=1
            else:
                dic[i]=1
    ans=[]
    value=list(dic.values())
    value.sort()
    se=value[-2]
    key=list(dic.keys())
    for i in key:
        if dic[i]==se:
            ans.append(i)
    ans.sort()
    print(*ans)
```

M04133: 垃圾炸弹

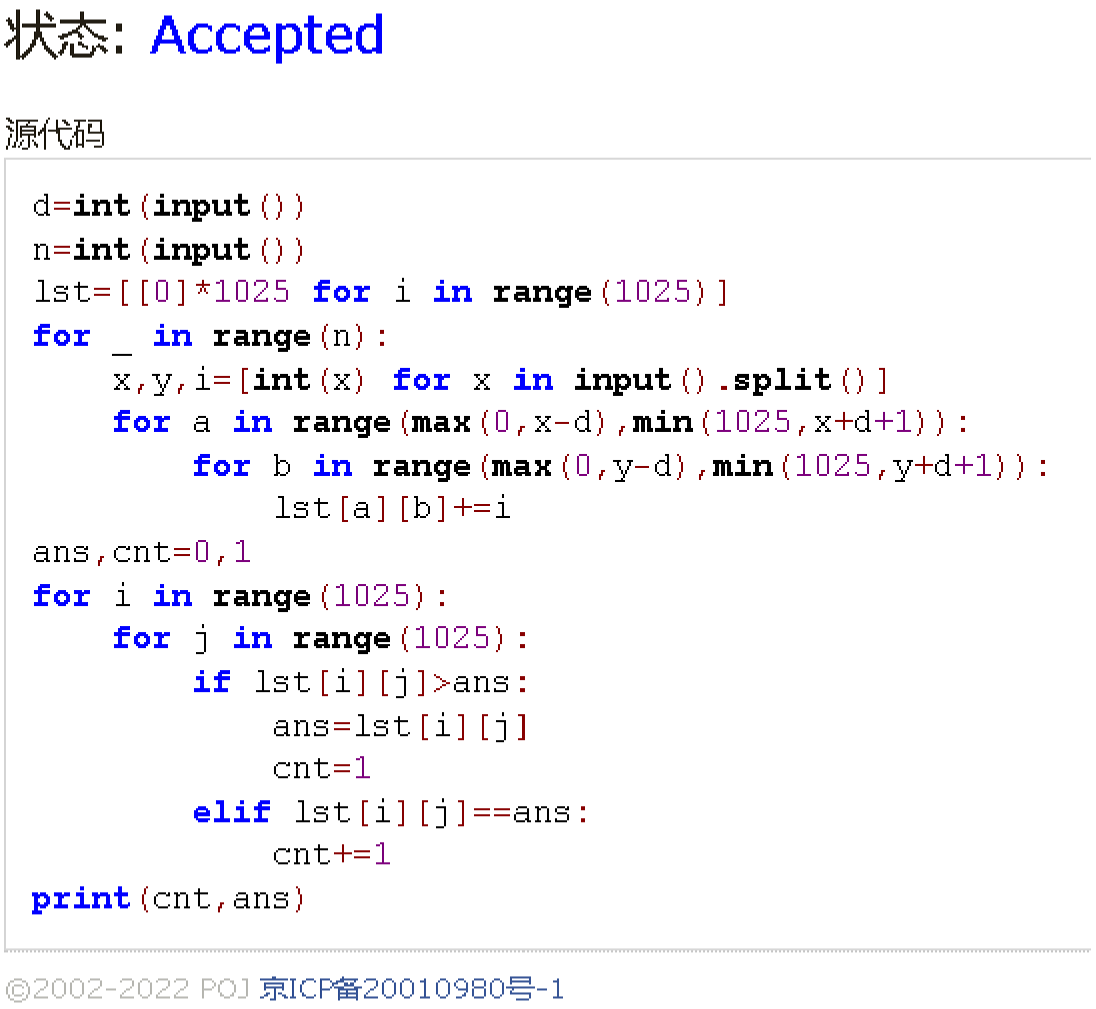
matrices, http://cs101.openjudge.cn/practice/04133/

思路：写过的题目还是不会写，啊啊感觉之前写过那么多题目都白费了记不住

代码：

```
d=int(input())
n=int(input())
lst=[[0]*1025 for i in range(1025)]
for _ in range(n):
    x,y,i=[int(x) for x in input().split()]
    for a in range(max(0,x-d),min(1025,x+d+1)):
        for b in range(max(0,y-d),min(1025,y+d+1)):
            lst[a][b]+=i
ans,cnt=0,1
for i in range(1025):
    for j in range(1025):
        if lst[i][j]>ans:
            ans=lst[i][j]
            cnt=1
        elif lst[i][j]==ans:
            cnt+=1
print(cnt,ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
d=int(input())
n=int(input())
lst=[[0]*1025 for i in range(1025)]
for _ in range(n):
    x,y,i=[int(x) for x in input().split()]
    for a in range(max(0,x-d),min(1025,x+d+1)):
        for b in range(max(0,y-d),min(1025,y+d+1)):
            lst[a][b]+=i
ans,cnt=0,1
for i in range(1025):
    for j in range(1025):
        if lst[i][j]>ans:
            ans=lst[i][j]
            cnt=1
        elif lst[i][j]==ans:
            cnt+=1
print(cnt,ans)
```

©2002-2022 POJ 京ICP备20010980号-1

T02488: A Knight's Journey

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

思路:

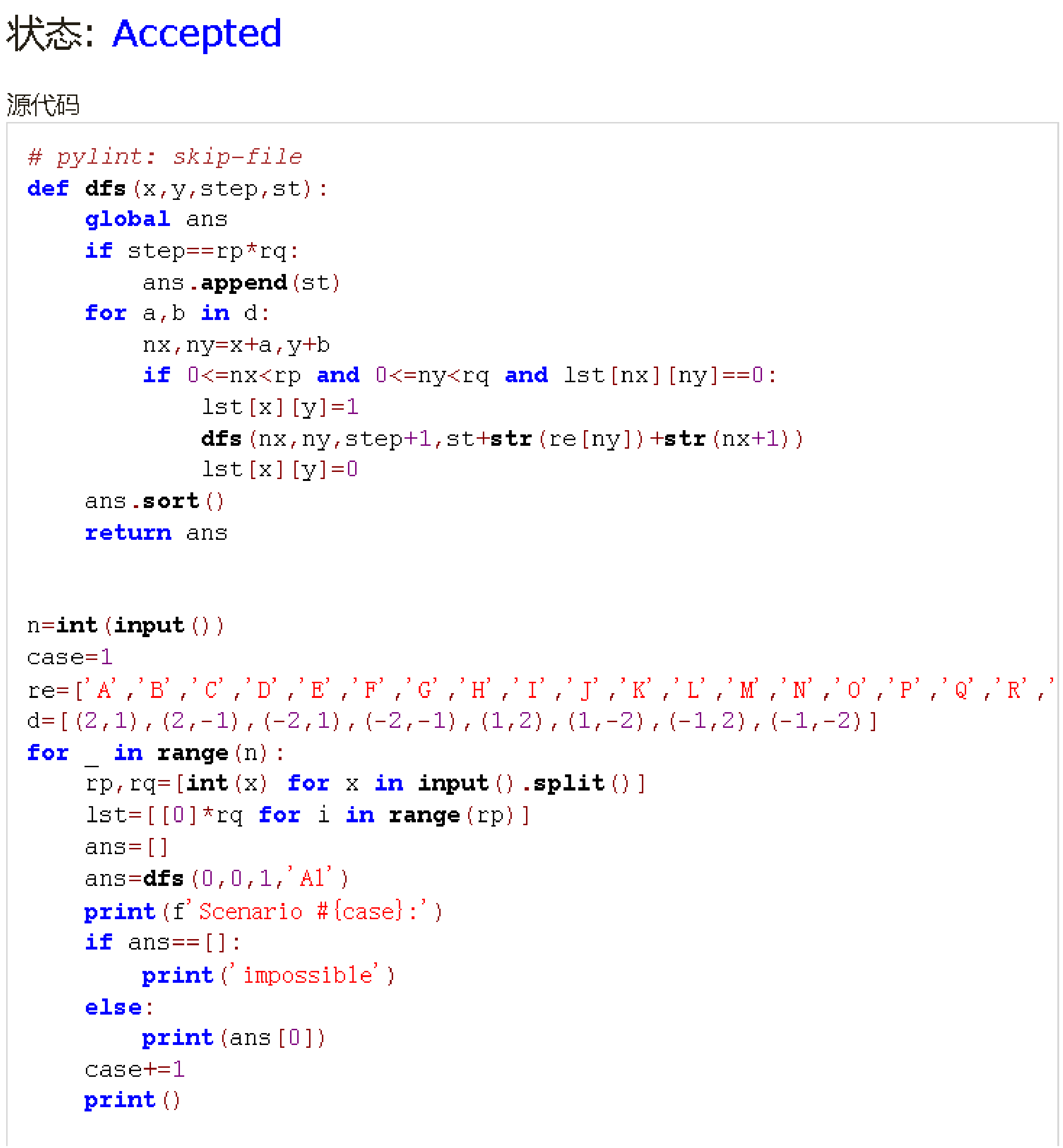
代码:

```
# pylint: skip-file
def dfs(x,y,step,st):
    global ans
    if step==rp*rq:
        ans.append(st)
    for a,b in d:
        nx,ny=x+a,y+b
        if 0<=nx<rp and 0<=ny<rq and lst[nx][ny]==0:
            lst[x][y]=1
            dfs(nx,ny,step+1,st+str(re[ny])+str(nx+1))
            lst[x][y]=0
    ans.sort()
    return ans

n=int(input())
case=1
```

```
re=
['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
d=[(2,1),(2,-1),(-2,1),(-2,-1),(1,2),(1,-2),(-1,2),(-1,-2)]
for _ in range(n):
    rp,rq=[int(x) for x in input().split()]
    lst=[[0]*rq for i in range(rp)]
    ans=[]
    ans=dfs(0,0,1,'A1')
    print(f'Scenario #{case}:')
    if ans==[]:
        print('impossible')
    else:
        print(ans[0])
    case+=1
    print()
```

代码运行截图 (至少包含有"Accepted")



状态: Accepted

源代码

```
# pylint: skip-file
def dfs(x,y,step,st):
    global ans
    if step==rp*rq:
        ans.append(st)
    for a,b in d:
        nx,ny=x+a,y+b
        if 0<=nx<rp and 0<=ny<rq and lst[nx][ny]==0:
            lst[x][y]=1
            dfs(nx,ny,step+1,st+str(re[ny])+str(nx+1))
            lst[x][y]=0
    ans.sort()
    return ans


n=int(input())
case=1
re=['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','
d=[(2,1),(2,-1),(-2,1),(-2,-1),(1,2),(1,-2),(-1,2),(-1,-2)]
for _ in range(n):
    rp,rq=[int(x) for x in input().split()]
    lst=[[0]*rq for i in range(rp)]
    ans=[]
    ans=dfs(0,0,1,'A1')
    print(f'Scenario #{case}:')
    if ans==[]:
        print('impossible')
    else:
        print(ans[0])
    case+=1
    print()
```

T06648: Sequence

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

思路：这道题有答案吗，在题解里面没有找到

太难太难了，写不出来......用ai还一直超内存

呜呜呜参考了cyk同学的代码后进行了修正，终于AC了

代码：

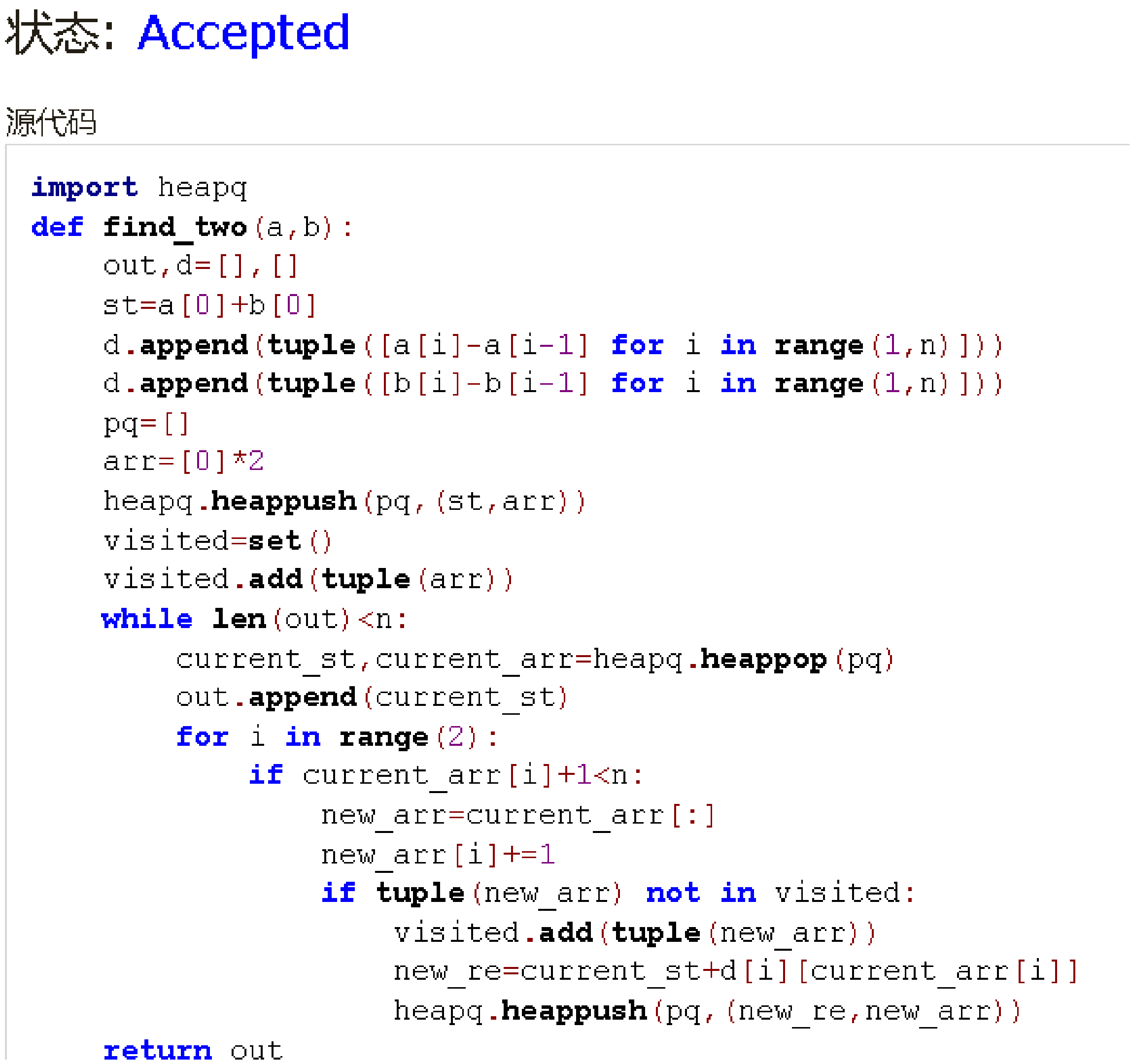
```python
import heapq
def find_two(a,b):
    out,d=[],[]
    st=a[0]+b[0]
    d.append(tuple([a[i]-a[i-1] for i in range(1,n)]))
    d.append(tuple([b[i]-b[i-1] for i in range(1,n)]))
    pq=[]
    arr=[0]*2
    heapq.heappush(pq,(st,arr))
    visited=set()
    visited.add(tuple(arr))
    while len(out)<n:
        current_st,current_arr=heapq.heappop(pq)
        out.append(current_st)
        for i in range(2):
            if current_arr[i]+1<n:
                new_arr=current_arr[:]
                new_arr[i]+=1
                if tuple(new_arr) not in visited:
                    visited.add(tuple(new_arr))
                    new_re=current_st+d[i][current_arr[i]]
                    heapq.heappush(pq,(new_re,new_arr))
    return out

t=int(input())

def find(m,n):
    ans,lst=[],[]
    for __ in range(m):
        s=sorted([int(x) for x in input().split()])
        lst.append(s)
    if len(lst)==1:
        return lst[0]
    else:
        out=find_two(lst[0],lst[1])
        for i in lst[2:]:
            out=find_two(out,i)
        return sorted(out)

for _ in range(t):
    m,n=[int(x) for x in input().split()]
    ans=find(m,n)
    print(*ans)
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



状态: Accepted

源代码

```python
import heapq
def find_two(a,b):
    out,d=[],[]
    st=a[0]+b[0]
    d.append(tuple([a[i]-a[i-1] for i in range(1,n)]))
    d.append(tuple([b[i]-b[i-1] for i in range(1,n)]))
    pq=[]
    arr=[0]*2
    heapq.heappush(pq,(st,arr))
    visited=set()
    visited.add(tuple(arr))
    while len(out)<n:
        current_st,current_arr=heapq.heappop(pq)
        out.append(current_st)
        for i in range(2):
            if current_arr[i]+1<n:
                new_arr=current_arr[:]
                new_arr[i]+=1
                if tuple(new_arr) not in visited:
                    visited.add(tuple(new_arr))
                    new_re=current_st+d[i][current_arr[i]]
                    heapq.heappush(pq,(new_re,new_arr))
    return out
```

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。

感觉是不是应该把以前写过的题目拿出来重新写一遍了......考试时垃圾炸弹和马走日similar都没写出来