Assignment #3: 惊蛰 Mock Exam

Updated 1641 GMT+8 Mar 5, 2025

2025 spring, Complied by 任宇桐 物理学院

> **说明：**
>
> 1. **惊蛰月考**：AC5（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：

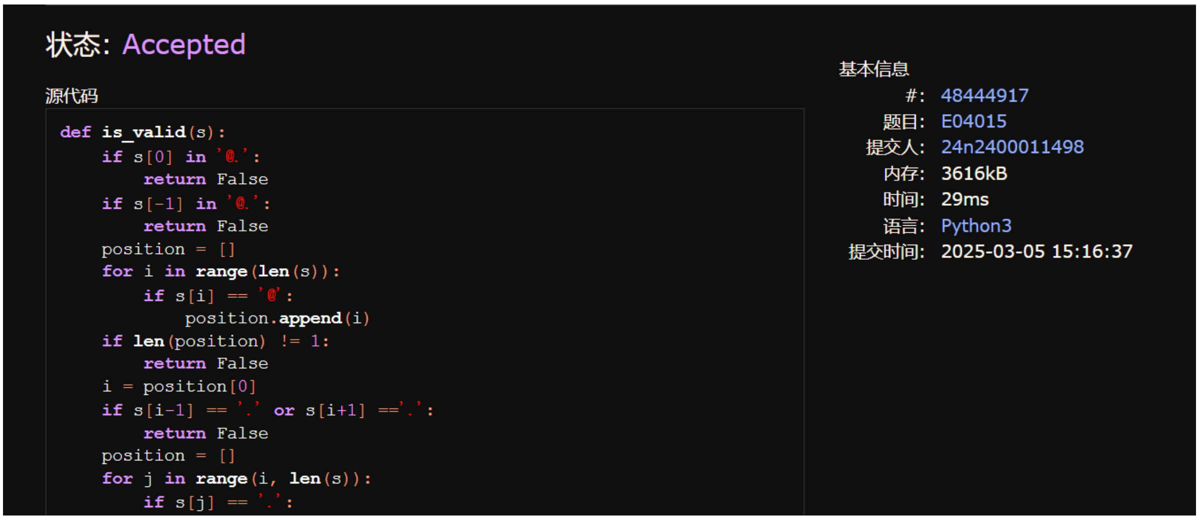
逐个判断条件即可。

代码：

```python
def is_valid(s):
    if s[0] in '@.':
        return False
    if s[-1] in '@.':
        return False
    position = []
    for i in range(len(s)):
        if s[i] == '@':
            position.append(i)
    if len(position) != 1:
        return False
    i = position[0]
    if s[i-1] == '.' or s[i+1] =='.':
        return False
    position = []
    for j in range(i, len(s)):
        if s[j] == '.':
            position.append(i)
    if not position:
        return False
    return True


while True:
    try:
        s = input()
    except EOFError:
        break
    if is_valid(s):
        print("YES")
    else:
        print('NO')
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```python
def is_valid(s):
    if s[0] in '@.':
        return False
    if s[-1] in '@.':
        return False
    position = []
    for i in range(len(s)):
        if s[i] == '@':
            position.append(i)
    if len(position) != 1:
        return False
    i = position[0]
    if s[i-1] == '.' or s[i+1] =='.':
        return False
    position = []
    for j in range(i, len(s)):
        if s[j] == '.':
```

基本信息

#:	48444917
题目:	E04015
提交人:	24n2400011498
内存:	3616kB
时间:	29ms
语言:	Python3
提交时间:	2025-03-05 15:16:37

M02039: 反反复复

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

思路：

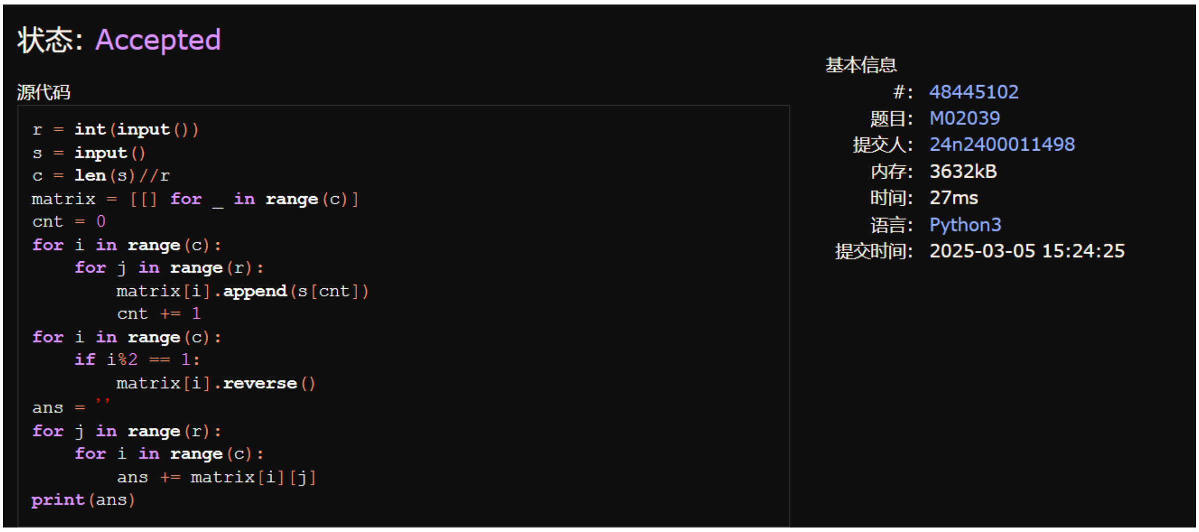
将矩阵还原直接实现即可。

代码：

```python
r = int(input())
s = input()
c = len(s)//r
matrix = [[] for _ in range(c)]
```

```
cnt = 0
for i in range(c):
    for j in range(r):
        matrix[i].append(s[cnt])
        cnt += 1
for i in range(c):
    if i%2 == 1:
        matrix[i].reverse()
ans = ''
for j in range(r):
    for i in range(c):
        ans += matrix[i][j]
print(ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
r = int(input())
s = input()
c = len(s)//r
matrix = [[] for _ in range(c)]
cnt = 0
for i in range(c):
    for j in range(r):
        matrix[i].append(s[cnt])
        cnt += 1
for i in range(c):
    if i%2 == 1:
        matrix[i].reverse()
ans = ''
for j in range(r):
    for i in range(c):
        ans += matrix[i][j]
print(ans)
```

基本信息

#: 48445102
题目: M02039
提交人: 24n2400011498
内存: 3632kB
时间: 27ms
语言: Python3
提交时间: 2025-03-05 15:24:25

M02092: Grandpa is Famous

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

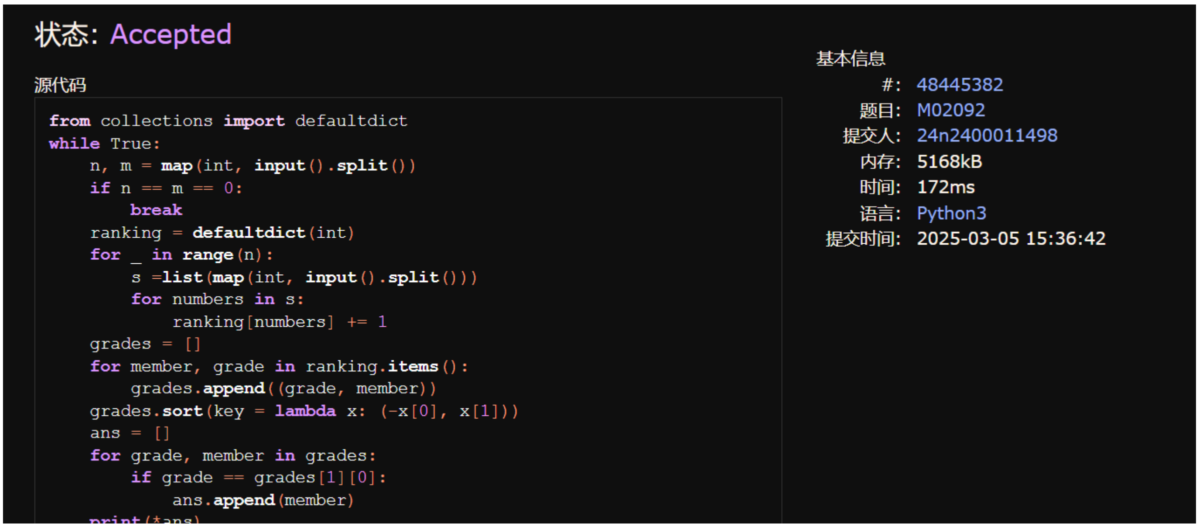
思路:

直接排序即可。

代码:

```
from collections import defaultdict
while True:
    n, m = map(int, input().split())
    if n == m == 0:
        break
    ranking = defaultdict(int)
    for _ in range(n):
        s =list(map(int, input().split()))
        for numbers in s:
            ranking[numbers] += 1
    grades = []
    for member, grade in ranking.items():
        grades.append((grade, member))
    grades.sort(key = lambda x: (-x[0], x[1]))
    ans = []
    for grade, member in grades:
        if grade == grades[1][0]:
            ans.append(member)
    print(*ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import defaultdict
while True:
    n, m = map(int, input().split())
    if n == m == 0:
        break
    ranking = defaultdict(int)
    for _ in range(n):
        s =list(map(int, input().split()))
        for numbers in s:
            ranking[numbers] += 1
    grades = []
    for member, grade in ranking.items():
        grades.append((grade, member))
    grades.sort(key = lambda x: (-x[0], x[1]))
    ans = []
    for grade, member in grades:
        if grade == grades[1][0]:
            ans.append(member)
    print(*ans)
```

基本信息

#:	48445382
题目:	M02092
提交人:	24n2400011498
内存:	5168kB
时间:	172ms
语言:	Python3
提交时间:	2025-03-05 15:36:42

M04133: 垃圾炸弹

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

思路：

逐个寻找即可，注意寻找的范围。

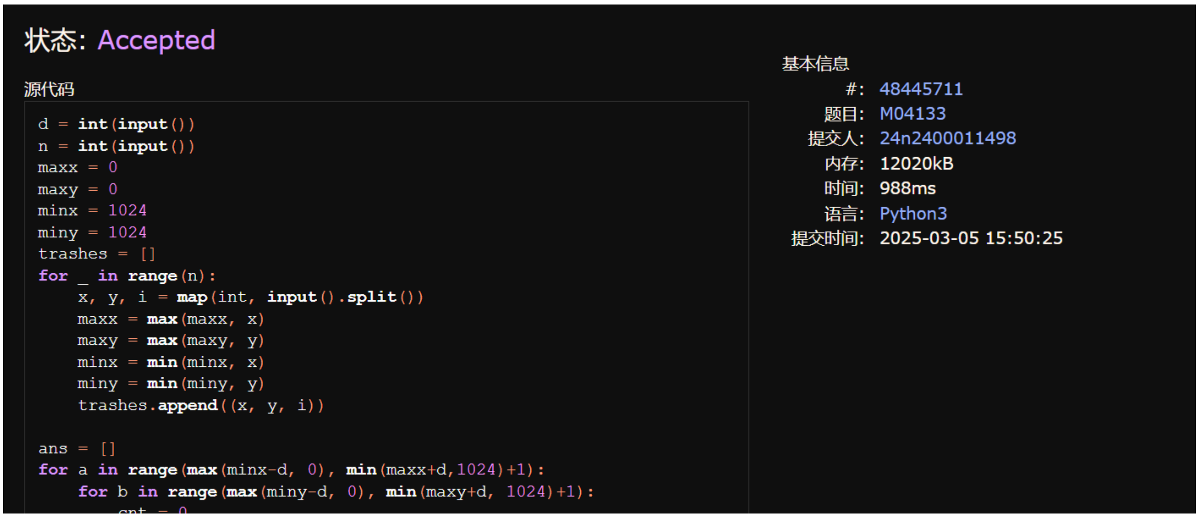
代码：

```
d = int(input())
n = int(input())
maxx = 0
maxy = 0
minx = 1024
miny = 1024
trashes = []
for _ in range(n):
    x, y, i = map(int, input().split())
    maxx = max(maxx, x)
    maxy = max(maxy, y)
    minx = min(minx, x)
    miny = min(miny, y)
    trashes.append((x, y, i))

ans = []
```

```
for a in range(max(minx-d, 0), min(maxx+d,1024)+1):
    for b in range(max(miny-d, 0), min(maxy+d, 1024)+1):
        cnt = 0
        for x, y, i in trashes:
            if abs(x-a) <= d and abs(y-b) <= d:
                cnt += i
        ans.append(cnt)
ans.sort(reverse=True)
num = 0
for i in ans:
    if i == ans[0]:
        num += 1
print(num, ans[0])
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
d = int(input())
n = int(input())
maxx = 0
maxy = 0
minx = 1024
miny = 1024
trashes = []
for _ in range(n):
    x, y, i = map(int, input().split())
    maxx = max(maxx, x)
    maxy = max(maxy, y)
    minx = min(minx, x)
    miny = min(miny, y)
    trashes.append((x, y, i))

ans = []
for a in range(max(minx-d, 0), min(maxx+d,1024)+1):
    for b in range(max(miny-d, 0), min(maxy+d, 1024)+1):
        cnt = 0
```

基本信息

#: 48445711

题目: M04133

提交人: 24n2400011498

内存: 12020kB

时间: 988ms

语言: Python3

提交时间: 2025-03-05 15:50:25

T02488: A Knight's Journey

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

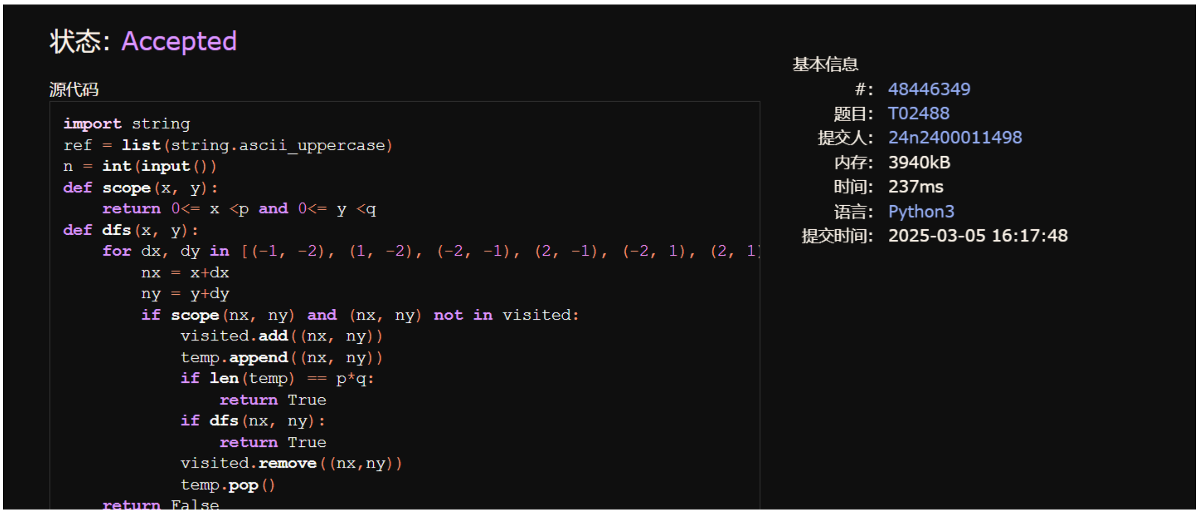
思路：

注意要按照字典序排列，所以要合理安排走的顺序。

代码：

```
import string
ref = list(string.ascii_uppercase)
n = int(input())
def scope(x, y):
    return 0<= x <p and 0<= y <q
def dfs(x, y):
    for dx, dy in [(-1, -2), (1, -2), (-2, -1), (2, -1), (-2, 1), (2, 1), (-1,
2), (1, 2)]:
        nx = x+dx
        ny = y+dy
        if scope(nx, ny) and (nx, ny) not in visited:
            visited.add((nx, ny))
            temp.append((nx, ny))
            if len(temp) == p*q:
                return True
            if dfs(nx, ny):
                return True
            visited.remove((nx,ny))
            temp.pop()
    return False
for i in range(n):
    p, q = map(int, input().split())
    print(f'Scenario #{i+1}: ')
    if p*q == 1:
        print('A1')
        print()
        continue
    visited = {(0, 0)}
    temp = [(0,0)]
    ans = ''
    if dfs(0, 0):
        for i , j in temp:
            ans += ref[j]+str(i+1)
        print(ans)
    else:
        print('impossible')
    print()
```

代码运行截图 (至少包含有"Accepted")



状态: Accepted

源代码

```
import string
ref = list(string.ascii_uppercase)
n = int(input())
def scope(x, y):
    return 0<= x <p and 0<= y <q
def dfs(x, y):
    for dx, dy in [(-1, -2), (1, -2), (-2, -1), (2, -1), (-2, 1), (2, 1
        nx = x+dx
        ny = y+dy
        if scope(nx, ny) and (nx, ny) not in visited:
            visited.add((nx, ny))
            temp.append((nx, ny))
            if len(temp) == p*q:
                return True
            if dfs(nx, ny):
                return True
            visited.remove((nx,ny))
            temp.pop()
    return False
```

基本信息

#:	48446349
题目:	T02488
提交人:	24n2400011498
内存:	3940kB
时间:	237ms
语言:	Python3
提交时间:	2025-03-05 16:17:48

T06648: Sequence

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

思路:

看了题解才明白思路，感觉自己太笨了，考场上快速想到用heap处理后30min+都没有想出来答案的思路。

下来修改了自己考场上的思路，发现代码稍微长了一些，但是时间复杂度差不多。考虑了对于现在已有的最小和，将所有一步增量的操作push到堆中，找到最小的值。感觉思路似乎更加直接一些？

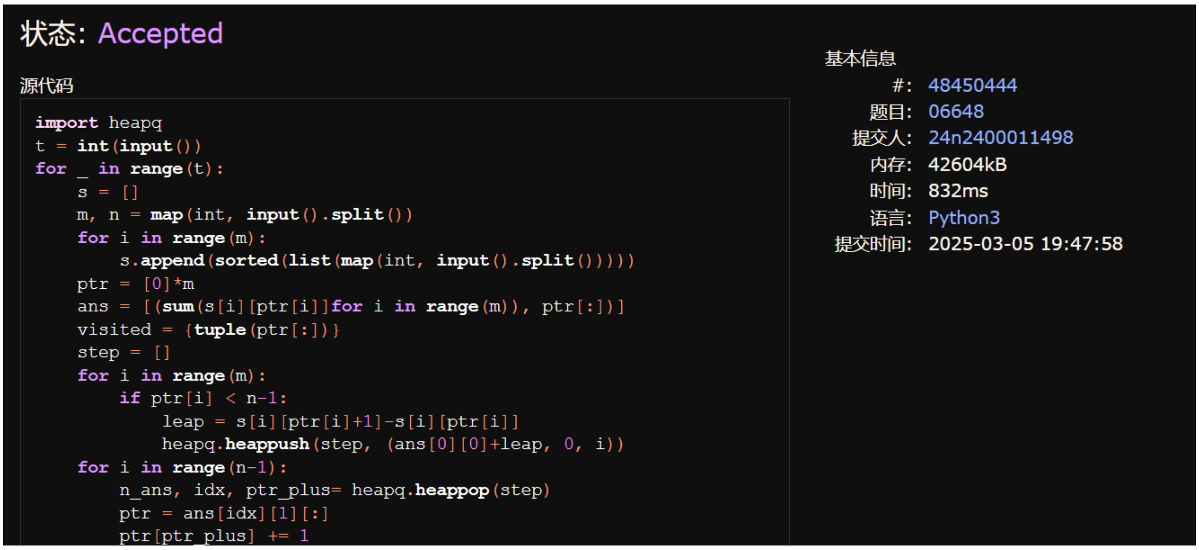
代码：

```
# 考试时想到的思路，832ms AC
import heapq
t = int(input())
for _ in range(t):
    s = []
    m, n = map(int, input().split())
    for i in range(m):
        s.append(sorted(list(map(int, input().split()))))
    ptr = [0]*m
    ans = [(sum(s[i][ptr[i]]for i in range(m)), ptr[:])]
    visited = {tuple(ptr[:])}
    step = []
    for i in range(m):
        if ptr[i] < n-1:
            leap = s[i][ptr[i]+1]-s[i][ptr[i]]
            heapq.heappush(step, (ans[0][0]+leap, 0, i))
    for i in range(n-1):
        n_ans, idx, ptr_plus= heapq.heappop(step)
        ptr = ans[idx][1][:]
        ptr[ptr_plus] += 1
        while tuple(ptr) in visited:
            n_ans, idx, ptr_plus = heapq.heappop(step)
            ptr = ans[idx][1][:]
            ptr[ptr_plus] += 1
        ans.append((n_ans, ptr[:]))
        visited.add(tuple(ptr))
        for j in range(m):
            if ptr[j] < n-1:
                leap = s[j][ptr[j] + 1] - s[j][ptr[j]]
                heapq.heappush(step, (n_ans+leap, len(ans)-1, j))
    ans_ = list([i[0] for i in ans])
    print(*ans_)

# 对照题解写的代码，942ms AC
import heapq

t = int(input())
for _ in range(t):
    m, n = map(int, input().split())
    s = sorted(list(map(int, input().split())))
    for i in range(m-1):
        new_s = sorted(list(map(int, input().split())))
        minheap = [(s[i]+new_s[0], i ,0) for i in range(n)]
        result = []
        for _ in range(n):
            current_num, i, j = heapq.heappop(minheap)
            result.append(current_num)
            if j+1 < len(new_s):
                heapq.heappush(minheap, (s[i]+new_s[j+1], i, j+1))
        s = result
    print(*s)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



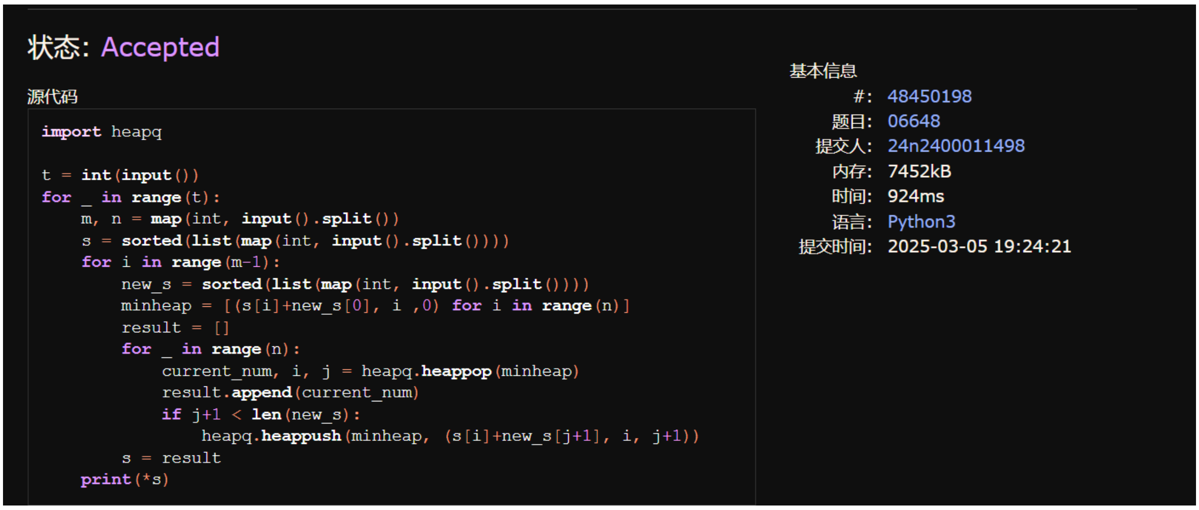
状态: Accepted

源代码

```
import heapq
t = int(input())
for _ in range(t):
    s = []
    m, n = map(int, input().split())
    for i in range(m):
        s.append(sorted(list(map(int, input().split()))))
    ptr = [0]*m
    ans = [(sum(s[i][ptr[i]]for i in range(m)), ptr[:])]
    visited = {tuple(ptr[:])}
    step = []
    for i in range(m):
        if ptr[i] < n-1:
            leap = s[i][ptr[i]+1]-s[i][ptr[i]]
            heapq.heappush(step, (ans[0][0]+leap, 0, i))
    for i in range(n-1):
        n_ans, idx, ptr_plus= heapq.heappop(step)
        ptr = ans[idx][1][:]
        ptr[ptr_plus] += 1
```

基本信息

#:	48450444
题目:	06648
提交人:	24n2400011498
内存:	42604kB
时间:	832ms
语言:	Python3
提交时间:	2025-03-05 19:47:58



状态: Accepted

源代码

```
import heapq

t = int(input())
for _ in range(t):
    m, n = map(int, input().split())
    s = sorted(list(map(int, input().split())))
    for i in range(m-1):
        new_s = sorted(list(map(int, input().split())))
        minheap = [(s[i]+new_s[0], i ,0) for i in range(n)]
        result = []
        for _ in range(n):
            current_num, i, j = heapq.heappop(minheap)
            result.append(current_num)
            if j+1 < len(new_s):
                heapq.heappush(minheap, (s[i]+new_s[j+1], i, j+1))
        s = result
    print(*s)
```

基本信息

#:	48450198
题目:	06648
提交人:	24n2400011498
内存:	7452kB
时间:	924ms
语言:	Python3
提交时间:	2025-03-05 19:24:21

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。

感觉月考做出来的题目都是之前计概做过或者类似的题目，之前没有做过的题目完全解决不了。还有就是英文题目考试的时候读起来太紧张了，半天都没读懂，希望正式考试的时候少些。