

1. 题目

M17975: 用二次探查法建立散列表

<http://cs101.openjudge.cn/practice/17975/>

需要用这样接收数据。因为输入数据可能分行了，不是题面描述的形式。OJ 上面有的题目是给 C++设计的，细节考虑不周全。

```
import sys
input = sys.stdin.read
data = input().split()
index = 0
n = int(data[index])
index += 1
m = int(data[index])
index += 1
num_list = [int(i) for i in data[index:index+n]]
思路:
```

代码:

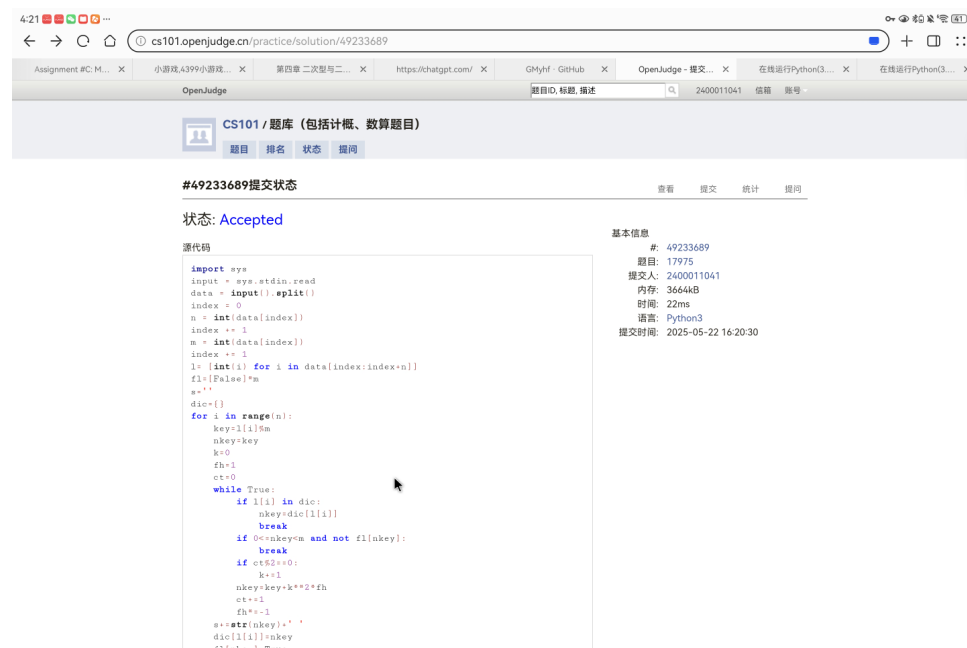
```
import sys
input = sys.stdin.read
data = input().split()
index = 0
n = int(data[index])
index += 1
m = int(data[index])
index += 1
l = [int(i) for i in data[index:index+n]]
fl = [False]*m
s = ""
dic = {}
for i in range(n):
    key = l[i] % m
    nkey = key
    k = 0
    fh = 1
    ct = 0
    while True:
        if l[i] in dic:
            nkey = dic[l[i]]
            break
        if 0 <= nkey < m and not fl[nkey]:
            break
        if ct % 2 == 0:
```

```

        k+=1
        nkey=key+k**2*fh
        ct+=1
        fh*=-1
        s+=str(nkey)+' '
        dic[l[i]]=nkey
        fl[nkey]=True
print(s[:-1])

```

代码运行截图（至少包含有"Accepted"）



M01258: Agri-Net
MST, <http://cs101.openjudge.cn/practice/01258/>

思路:

代码:

```

import heapq
while True:
    try:
        n=int(input())
        l=[]
        for i in range(n):
            l.append(list(map(int,input().split())))
        visit=set()
        path=[]
        heapq.heappush(path,(0,0))
        cur=[float('inf') for i in range(n)]
        op=0
        while len(visit)<n:

```

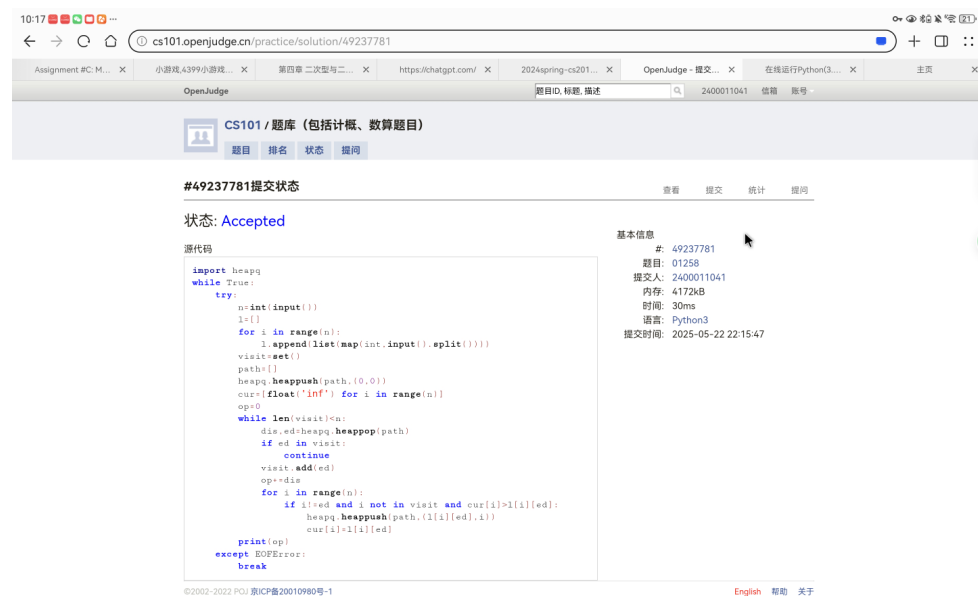
```

dis,ed=heapq.heappop(path)
if ed in visit:
    continue
visit.add(ed)
op+=dis
for i in range(n):
    if i!=ed and i not in visit and cur[i]>l[i][ed]:
        heapq.heappush(path,(l[i][ed],i))
        cur[i]=l[i][ed]

print(op)
except EOFError:
    break

```

代码运行截图 （至少包含有"Accepted"）



M3552.网络传送门旅游

bfs, <https://leetcode.cn/problems/grid-teleportation-traversal/>

思路:

代码:

class Solution(object):

```
def minMoves(self, matrix):
```

```
    """
```

```
:type matrix: List[str]
```

```
:rtype: int
```

```
"""
```

```
import heapq
```

```
n=len(matrix)
```

```
m=len(matrix[0])
```

```
l=[[k for k in s]for s in matrix]
```

```
dic={}
```

```
for i in range(n):
```

```
    for j in range(m):
```

```
        if 'A'<=l[i][j]<='Z':
```

```
            if l[i][j] in dic:
```

```
                dic[l[i][j]].append((i,j))
```

```
            else:
```

```
                dic[l[i][j]]=[(i,j)]
```

```
path=[]
```

```
heapq.heappush(path,(0,0,0))
```

```
step=[[0,-1],[0,1],[1,0],[-1,0]]
```

```
op=-1
```

```
while path:
```

```
    d,x,y=heapq.heappop(path)
```

```
    if x==n-1 and y==m-1:
```

```
        return d
```

```
    if l[x][y] in dic:
```

```

for (x1,y1) in dic[l[x][y]]:

    if (x1,y1)!=(x,y) and l[x1][y1]!='#':

        heapq.heappush(path,(d,x1,y1))

        l[x1][y1]='#'

l[x][y]='#'

for k in step:

    nx=x+k[0]

    ny=y+k[1]

    if 0<=nx<n and 0<=ny<m and l[nx][ny]!='#':

        heapq.heappush(path,(d+1,nx,ny))

        if l[nx][ny]=='.':

            l[nx][ny]='#'

return -1

```

代码运行截图 （至少包含有"Accepted"）

4:15

leetcode.cn/problems/grid-teleportation-traversal/description

OpenJudge - 提交... X 在线运行Python3... X Assignment #C: M... X 小游戏.4399小游戏... X 第四章 二次型与二... X 2024spring-cs201... X 3552. 网格传送门... X 在线运行Python3...

题库 < > 运行 提交 通过

3552. 网格传送门旅游

已解答

给你一个大小为 $m \times n$ 的二维字符串网格 `matrix`，用字符串数组表示，其中 `matrix[i][j]` 表示第 i 行和第 j 列处的单元格。每个单元格可以是以下几种字符之一：

- `'.'` 表示一个空单元格。
- `'g'` 表示一个障碍物。
- 一个大写字母（`'A'` 到 `'Z'`）表示一个传送门。

你从左上角单元格 $(0, 0)$ 出发，目标是到达右下角单元格 $(m - 1, n - 1)$ 。你可以从当前位置移动到相邻的单元格（上、下、左、右），移动后的单元格必须在网格边界内且不是障碍物。

如果你踏入一个包含传送门字母的单元格，并且你之前没有使用过该传送门字母，你可以立即传送到网格中另一个具有相同字母的单元格。这次传送不计入移动次数，但每个字母对应的传送门在旅程中最多只能使用一次。

返回到达右下角单元格所需的最少移动次数。如果无法到达目的地，则返回 -1 。

示例 1：

输入: `matrix = ["A..",".A..","..."]`

输出: 2

解释:

A	.	.
.	A	.
.	.	.

通过 608 / 608 个通过的测试用例

Mystifying Sutherland9A 提交于 2025.05.23 16:10

执行用时分布 8229 ms 击败 11.11%

消耗内存分布 70.42 MB 击败 66.67%

全部提交记录

笔记 x 测试用例 > 测试结果

Case 1 Case 2 +

matrix =

```
["A..",".A..","..."]
```

M787.K 站中转内最便宜的航班

Bellman Ford, <https://leetcode.cn/problems/cheapest-flights-within-k-stops/>

思路:

代码:

class Solution(object):

```
def findCheapestPrice(self, n, flights, src, dst, k):
```

```
    """
```

```
    :type n: int
```

```
    :type flights: List[List[int]]
```

```
    :type src: int
```

```
    :type dst: int
```

```
    :type k: int
```

```
    :rtype: int
```

```
    """
```

```
import heapq
```

```

prc={}

for q in flights:

    prc[(q[0],q[1])]=q[2]

cur={}

path=[]

heapq.heappush(path,(0,0,src))

while path:

    p,t,name=heapq.heappop(path)

    if name==dst:

        return p

    if (name,t) in cur and p>=cur[(name,t)]:

        continue

    cur[(name,t)]=p

    for (st,ed) in prc:

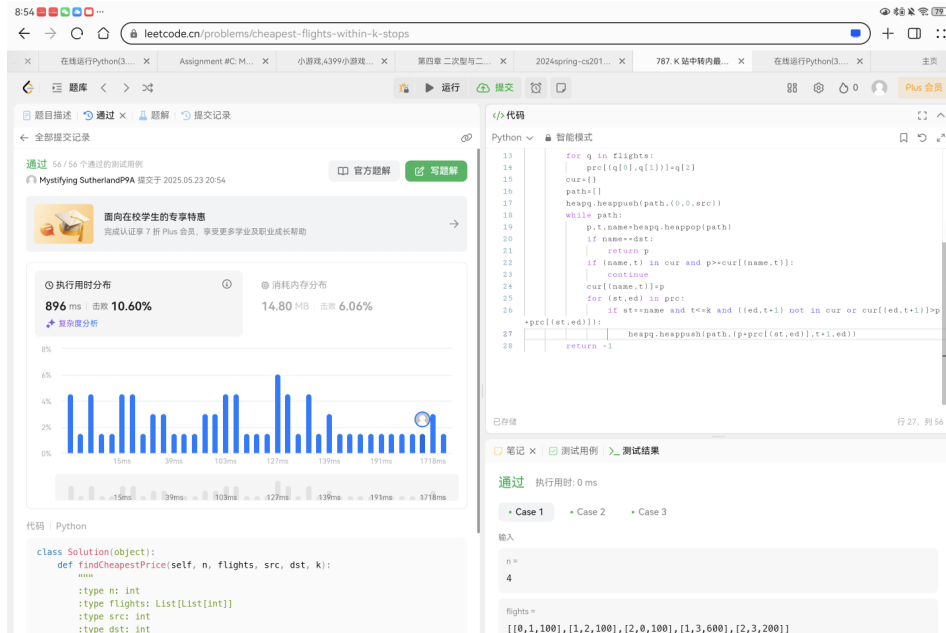
        if st==name and t<=k :

            heapq.heappush(path,(p+prc[(st,ed)],t+1,ed))

return -1

```

代码运行截图 （至少包含有"Accepted"）



M03424: Candies

Dijkstra, <http://cs101.openjudge.cn/practice/03424/>

思路:

代码:

```
import heapq
```

```
n,m=map(int,input().split())
```

```
link=[[]for i in range(n)]
```

```
for j in range(m):
```

```
    st,ed,d=map(int,input().split())
```

```
    link[st-1].append((ed-1,d))
```

```
path=[(0,0)]
```

```
cur=[float('inf') for i in range(n)]
```

```
v=[False]*n
```

```
cur[0]=0
```

```
while path:
```



```

d,name=heapq.heappop(path)

if name==n-1:

    print(d)

    break

if v[name]:

    continue

v[name]=True

for (ed,ds) in link[name]:

    if cur[ed]>ds+d:

        cur[ed]=ds+d

        heapq.heappush(path,(ds+d,ed))

```

代码运行截图 （至少包含有"Accepted"）

The screenshot shows a web browser window with the URL `cs101.openjudge.cn/practice/solution/49245085`. The page displays the submission status for problem #49245085 as "Accepted". The code is as follows:

```

import heapq
n,m=map(int,input().split())
link=[[]for i in range(n)]
for j in range(m):
    st,ed,d=map(int,input().split())
    link[st-1].append((ed-1,d))
path=[(0,0)]
cur=[float('inf') for i in range(n)]
v=[False]*n
cur[0]=0
while path:
    d,name=heapq.heappop(path)
    if name==n-1:
        print(d)
        break
    if v[name]:
        continue
    v[name]=True
    for (ed,ds) in link[name]:
        if cur[ed]>ds+d:
            cur[ed]=ds+d
            heapq.heappush(path,(ds+d,ed))

```

On the right side of the submission page, the following basic information is displayed:

- #: 49245085
- 题目: 03424
- 提交人: 2400011041
- 内存: 27176kB
- 时间: 370ms
- 语言: Python3
- 提交时间: 2025-05-23 21:21:08

M22508:最小奖金方案

topological order, <http://cs101.openjudge.cn/practice/22508/>

思路:

代码:

```
n,m=map(int,input().split())
```

```
d={}
```

```
num=[0 for i in range(n)]
```

```
link=[[] for i in range(n)]
```

```
v=[False]*n
```

```
for k in range(n):
```

```
    d[k]=0
```

```
for i in range(m):
```

```
    win,lose=map(int,input().split())
```

```
    link[lose].append(win)
```

```
    d[win]+=1
```

```
ct=0
```

```
while ct<n:
```

```
    for k in d:
```

```
        if not v[k] and d[k]==0:
```

```
            v[k]=True
```

```
            for j in link[k]:
```

```
                num[j]=max(num[j],num[k]+1)
```

```
                d[j]-=1
```

```
            ct+=1
```

```
op=int(100*n+sum(num))
```

```
print(op)
```

代码运行截图（至少包含有"Accepted"）



2. 学习总结和收获

本次作业还是有点复杂的，同时也反映出自己对 djstl 模板的不熟悉之处。在机考前还需要熟悉各类经典算法的经典写法。

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。

复习以往代码中.....