

Assignment #D: 图 & 散列表

Updated 2042 GMT+8 May 20, 2025

2025 spring, Compiled by 郑涵予 物理学院

说明：

1. 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。

2. **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。

3. **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

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：

直接按照要求实现即可，注意键可能重复。最坑的点无疑是输入数据的问题，在这上面花了不少时间WA了十几次。而且很神奇的是居然是WA不是RE导致没有第一时间发现，不然根据上次水淹七军那题的经验应该能很快意识到的。（用时约30min，输入数据害的（））

代码：

```

import sys
input=sys.stdin.read
data=input().split()
pos=0
n=int(data[pos])
pos+=1
m=int(data[pos])
pos+=1
a=[int(x) for x in data[pos:pos+n]]
mp={}
visited=[False]*m
for x in a:
    if x in mp:
        continue
    index=x%m
    if not visited[index]:
        visited[index]=True
        mp[x]=index
    else:
        i=1
        while 1:
            index1=(index-i*i)%m
            index2=(index+i*i)%m
            if not visited[index2]:
                visited[index2]=True
                mp[x]=index2
                break
            elif not visited[index1]:
                visited[index1]=True
                mp[x]=index1
                break
            i+=1
b=[mp[x] for x in a]
print(' '.join(map(str,b)))

```

代码运行截图 (至少包含有"Accepted")

CS101 / 题库（包括计概、数算题目）

题目排名状态提问

#49063242提交状态

查看提交统计提问

状态: Accepted

源代码

```
import sys
input=sys.stdin.read
data=input().split()
pos=0
n=int(data[pos])
pos+=1
m=int(data[pos])
pos+=1
a=[int(x) for x in data[pos:pos+n]]
mp={}
visited=[False]*m
for x in a:
    if x in mp:
        continue
    index=x%m
    if not visited[index]:
        visited[index]=True
        mp[x]=index
    else:
        i=1
        while 1:
            index1=(index-i*i)%m
            index2=(index+i*i)%m
            if not visited[index2]:
                visited[index2]=True
                mp[x]=index2
                break
            elif not visited[index1]:
```

基本信息

#: 49063242
题目: 17975
提交人: 24n2400011325
内存: 7536kB
时间: 23ms
语言: Python3
提交时间: 2025-05-05 10:58:24

M01258: Agri-Net

MST, <http://cs101.openjudge.cn/practice/01258/>

思路:

经典的最小生成树（用时约10min）

代码:

```
import sys
data=sys.stdin.read().split()
index=0
while index<len(data):
    n=int(data[index])
    index+=1
    a=[]
    for i in range(n):
        for j in range(n):
            if j>=i:
                a.append((int(data[index]),i,j))
            index+=1
    root=list(range(n))
    a.sort()

    def find_root(x):
        if x!=root[x]:
            root[x]=find_root(root[x])
        return root[x]

    ans=0
    count=0
    for w,i,j in a:
        if count==n-1:
            break
        ri,rj=find_root(i),find_root(j)
        if ri!=rj:
            root[ri]=rj
            ans+=w
```

```
count+=1  
print(ans)
```

代码运行截图 (至少包含有"Accepted")



OpenJudge 题目ID, 标题, 描述 24n2400011325 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#49144876提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
import sys  
data=sys.stdin.read().split()  
index=0  
while index<len(data):  
    n=int(data[index])  
    index+=1  
    a=[]  
    for i in range(n):  
        for j in range(n):  
            if j>=i:  
                a.append((int(data[index]),i,j))  
                index+=1  
    root=list(range(n))  
    a.sort()  
  
    def find_root(x):  
        if x!=root[x]:  
            root[x]=find_root(root[x])  
        return root[x]  
  
    ans=0  
    count=0  
    for w,i,j in a:  
        if count==n-1:  
            break
```

基本信息

#: 49144876
题目: 01258
提交人: 24n2400011325
内存: 5488kB
时间: 34ms
语言: Python3
提交时间: 2025-05-13 23:56:59

M3552.网络传送门旅游

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

思路:

直接用Dijkstra即可 (用时约10min)

代码:

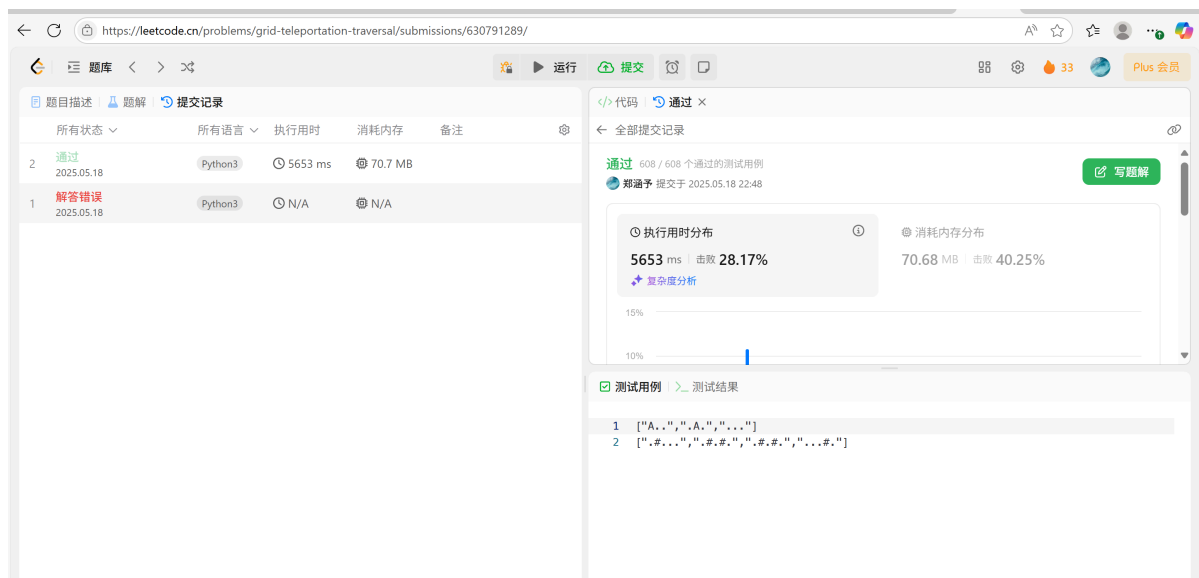
```
class Solution:  
    def minMoves(self, matrix: List[str]) -> int:  
        m,n=len(matrix),len(matrix[0])  
        directions=[(1,0),(-1,0),(0,1),(0,-1)]  
        q=[]  
        dist=[[float('inf')]*n for _ in range(m)]  
        dist[0][0]=0  
        used=[False]*26  
        heapq.heappush(q,(0,0,0))  
        mp=defaultdict(list)  
        ans=-1  
        for i in range(m):  
            for j in range(n):  
                if matrix[i][j].isalpha():  
                    mp[ord(matrix[i][j])-ord('A')].append((i,j))  
        while q:  
            step,x,y=heapq.heappop(q)  
            if x==m-1 and y==n-1:  
                return step  
            if matrix[x][y].isalpha() and not used[ord(matrix[x][y])-ord('A')]:  
                used[ord(matrix[x][y])-ord('A')]=True  
                for nx,ny in mp[ord(matrix[x][y])-ord('A')]:
```

```

        if dist[nx][ny]>step:
            dist[nx][ny]=step
            heapq.heappush(q,(step,nx,ny))
    for dx,dy in directions:
        nx,ny=x+dx,y+dy
        if 0<=nx<m and 0<=ny<n and matrix[nx][ny]!='#' and step+1<dist[nx][ny]:
            dist[nx][ny]=step+1
            heapq.heappush(q,(step+1,nx,ny))
    return -1

```

代码运行截图 (至少包含有"Accepted")



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:

用Bellman Ford进行k+1次松弛, 代码其实很简洁。但是自己做的时候如果没看标签大概率是不知道能这么做的,一开始写错了好几次。(用时约30min)

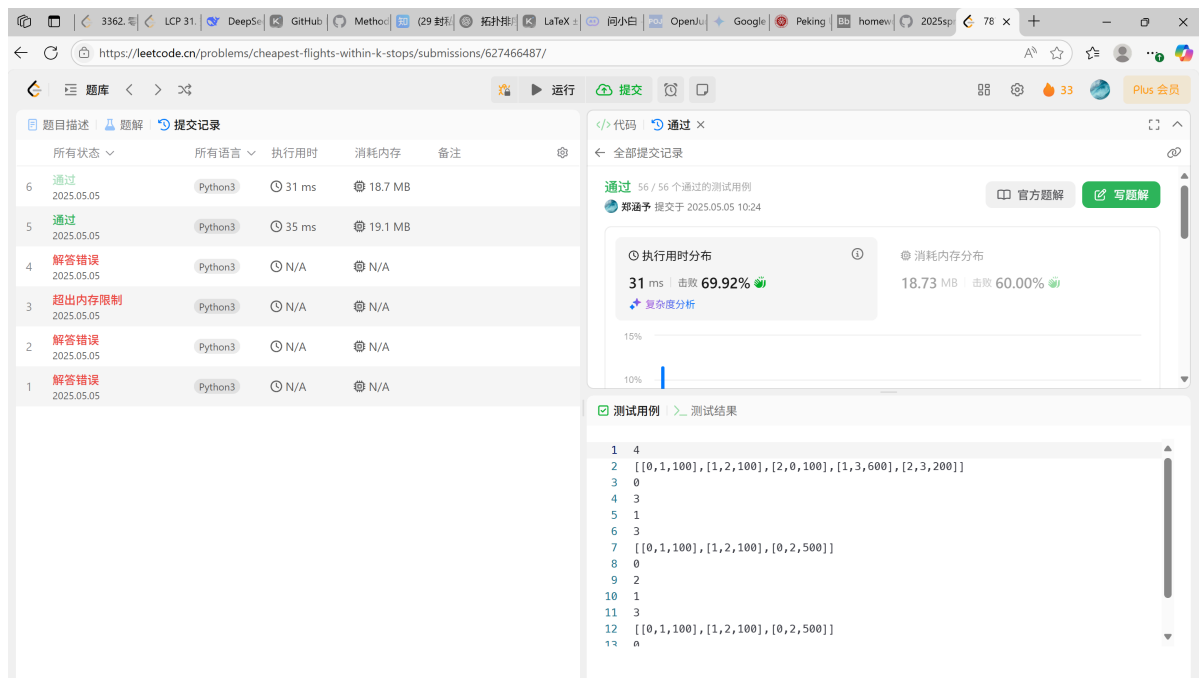
代码:

```

class Solution:
    def findCheapestPrice(self, n, flights, src, dst, K):
        dp = [float('inf')] * n
        dp[src] = 0
        for _ in range(K+1):
            tmp = dp[:]
            for u, v, w in flights:
                if dp[u] + w < tmp[v]:
                    tmp[v] = dp[u] + w
            dp = tmp
        return dp[dst] if dp[dst] < float('inf') else -1

```

代码运行截图 (至少包含有"Accepted")



M03424: Candies

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

思路：

看到标签后直接就秒了。但是要仔细想证明还是有难度的，能大概感觉得出来但不怎么会证。（用时约5min）

代码：

```
from heapq import *
n,m=map(int,input().split())
a=[[[] for _ in range(n+1)]
for _ in range(m):
    u,v,w=map(int,input().split())
    a[u].append((v,w))
q=[]
dist=[float('inf')]*(n+1)
dist[1]=0
heappush(q,(0,1))
while q:
    d,index=heappop(q)
    if d>dist[index]:
        continue
    if index==n:
        print(d)
        exit()
    for v,w in a[index]:
        if d+w<dist[v]:
            dist[v]=d+w
            heappush(q,(dist[v],v))
```

代码运行截图 (至少包含有"Accepted")

CS101 / 题库（包括计概、数算题目）

题目

排名

状态

提问

#49215090提交状态

查看提交统计提问

状态: Accepted

源代码

```
from heapq import *
n,m=map(int,input().split())
a=[[ ] for _ in range(n+1)]
for _ in range(m):
    u,v,w=map(int,input().split())
    a[u].append((v,w))
q=[]
dist=[float('inf')]*(n+1)
dist[1]=0
heappush(q,(0,1))
while q:
    d,index=heappop(q)
    if d>dist[index]:
        continue
    if index==n:
        print(d)
        exit()
    for v,w in a[index]:
        if d+w<dist[v]:
            dist[v]=d+w
            heappush(q,(dist[v],v))
```

基本信息

#: 49215090
题目: 03424
提交人: 24n2400011325
内存: 24448kB
时间: 357ms
语言: Python3
提交时间: 2025-05-20 15:33:24

©2002-2022 POJ 京ICP备20010980号-1

English帮助关于

M22508:最小奖金方案

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

思路：

可以直接使用lru_cache，这样连拓扑排序都省了，直接递归就行。（用时约5min）

代码：

```
from functools import lru_cache
n,m=map(int,input().split())
a=[[ ] for _ in range(n)]
for _ in range(m):
    u,v=map(int,input().split())
    a[u].append(v)
@lru_cache(maxsize=None)
def dp(i):
    res=100
    for x in a[i]:
        res=max(res,dp(x)+1)
    return res
ans=0
for i in range(n):
    ans+=dp(i)
print(ans)
```

状态: Accepted

源代码

```
from functools import lru_cache
n,m=map(int,input().split())
a=[[] for _ in range(n)]
for i in range(m):
    u,v=map(int,input().split())
    a[u].append(v)
@lru_cache(maxsize=None)
def dp(i):
    res=100
    for x in a[i]:
        res=max(res,dp(x)+1)
    return res
ans=0
for i in range(n):
    ans+=dp(i)
print(ans)
```

基本信息

#: 49062171
题目: 22508
提交人: 24n2400011325
内存: 3732kB
时间: 26ms
语言: Python3
提交时间: 2025-05-05 09:16:57

©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。

这周没打力扣周赛但跑去参加了信科的九坤杯，题目难度还是相当大的（考五个小时简直累死），我们小组三个人最后切出来四个题，拿了个三等奖（听说程设能加分，数算能不能（（（）。这种比赛对python还是有些不友好，我写题使用了调整过一些细节的Dijkstra，用python一交直接MLE,换C++就成功AC了。包括我们A题的dp也是python被卡超时，C++就过了。马上就机考了，还是希望机考顺利点别犯低级错误。