

# Assignment #D: 图 & 散列表

Updated 2042 GMT+8 May 20, 2025

2025 spring, Compiled by 任宇桐 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:

对照题目的提示, 其实可以直接实现。

代码:

```
import sys
```

```

n, m, *rest = map(int, sys.stdin.read().split())
s = rest[:n]
table = [None]*m
ans = []
for value in s:
    key = value%m
    if table[key] == value or table[key] is None:
        table[key] = value
        ans.append(key)
    else:
        i = 1
        signs = [1, -1]
        flag = 1
        while flag:
            for sign in signs:
                temp = (key + sign*i**2)%m
                if table[temp] == value or table[temp] is None:
                    table[temp] = value
                    ans.append(temp)
                    flag = 0
                    break
            i += 1
print(*ans)

```

代码运行截图 (至少包含有"Accepted")

#49140623提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

import sys

n, m, *rest = map(int, sys.stdin.read().split())
s = rest[:n]
table = [None]*m
ans = []
for value in s:
    key = value%m
    if table[key] == value or table[key] is None:
        table[key] = value
        ans.append(key)
    else:
        i = 1
        signs = [1, -1]
        flag = 1
        while flag:
            for sign in signs:
                temp = (key + sign*i**2)%m
                if table[temp] == value or table[temp] is None:
                    table[temp] = value
                    ans.append(temp)
                    flag = 0
                    break
            i += 1
print(*ans)

```

基本信息

#: 49140623  
 题目: 17975  
 提交人: 24n2400011498  
 内存: 3612kB  
 时间: 22ms  
 语言: Python3  
 提交时间: 2025-05-13 17:53:39

## M01258: Agri-Net

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

思路:

从任意一个节点出发, 开始找最小的边进行处理。

代码:

```
import heapq
while True:
    try:
        n = int(input())
    except EOFError:
        break
    matrix = []
    for _ in range(n):
        matrix.append(list(map(int, input().split())))

    edges = []
    visited = {0}
    ans = 0

    for i, edge in enumerate(matrix[0]):
        heapq.heappush(edges, (edge, i))
    while len(visited) < n:
        edge, node = heapq.heappop(edges)
        if node in visited:
            continue
        ans += edge
        visited.add(node)
        for i, edge in enumerate(matrix[node]):
            heapq.heappush(edges, (edge, i))
    print(ans)
```

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:

感觉算是bfs变式? 但是看了提示以后才想到, 如果遇到了“传送门”不用讨论, 直接全部入队即可, 不断更新。

代码:

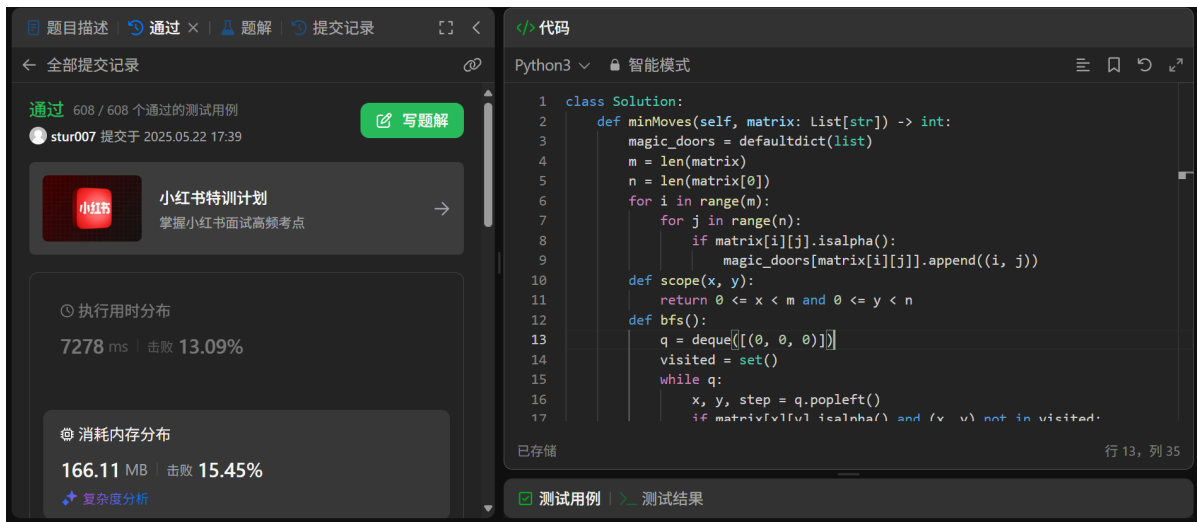
```
class Solution:
    def minMoves(self, matrix: List[str]) -> int:
```

```

magic_doors = defaultdict(list)
m = len(matrix)
n = len(matrix[0])
for i in range(m):
    for j in range(n):
        if matrix[i][j].isalpha():
            magic_doors[matrix[i][j]].append((i, j))
def scope(x, y):
    return 0 <= x < m and 0 <= y < n
def bfs():
    q = deque([(0, 0, 0)])
    visited = set()
    while q:
        x, y, step = q.popleft()
        if matrix[x][y].isalpha() and (x, y) not in visited:
            for x, y in magic_doors[matrix[x][y]]:
                if x == m-1 and y == n-1:
                    return step
                if (x, y) not in visited:
                    visited.add((x, y))
                    for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
                        nx = x + dx
                        ny = y + dy
                        if scope(nx, ny) and matrix[nx][ny] != '#' and
(nx, ny) not in visited:
                            q.append((nx, ny, step+1))
                    magic_doors[matrix[x][y]].clear()
            else:
                if x == m-1 and y == n-1:
                    return step
                if (x, y) not in visited:
                    visited.add((x, y))
                    for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
                        nx = x + dx
                        ny = y + dy
                        if scope(nx, ny) and matrix[nx][ny] != '#' and (nx,
ny) not in visited:
                            q.append((nx, ny, step+1))
                    return -1
    return bfs()

```

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/>

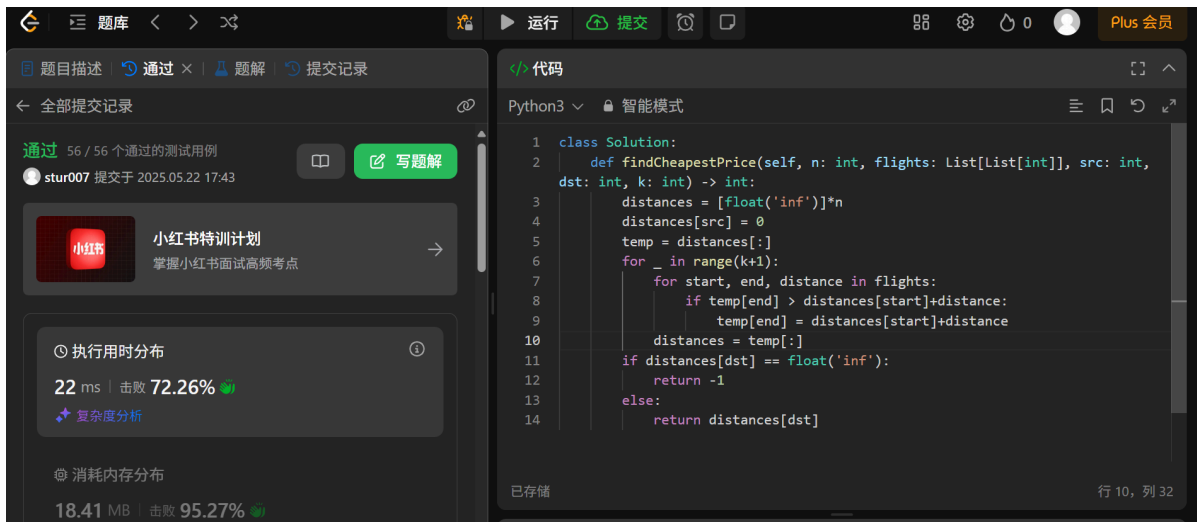
思路:

如果直接Dijkstra似乎过不了, 还需要剪枝。不如Bellman的方法代码简单。

代码:

```
class Solution:
    def findCheapestPrice(self, n: int, flights: List[List[int]], src: int, dst: int, k: int) -> int:
        distances = [float('inf')] * n
        distances[src] = 0
        temp = distances[:]
        for _ in range(k+1):
            for start, end, distance in flights:
                if temp[end] > distances[start] + distance:
                    temp[end] = distances[start] + distance
            distances = temp[:]
        if distances[dst] == float('inf'):
            return -1
        else:
            return distances[dst]
```

代码运行截图 (至少包含有"Accepted")



## M03424: Candies

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

思路:

虽然写的非常新鲜，但是就是直接Dijkstra的基本类型。

代码:

```
from collections import defaultdict
import heapq

n, m = map(int, input().split())
paths = defaultdict(list)
for _ in range(m):
    a, b, c = map(int, input().split())
    paths[a-1].append((b-1, c))
distance = [float('inf') for _ in range(n)]
distance[0] = 0
q = [(0, 0)]
while q:
    cd, cn = heapq.heappop(q)
    if cn == n-1:
        print(cd)
        break
    if cd > distance[cn]:
        continue
    for nn, d in paths[cn]:
        nd = cd + d
        if nd < distance[nn]:
            distance[nn] = nd
            heapq.heappush(q, (nd, nn))
```

代码运行截图 (至少包含有"Accepted")

## M22508:最小奖金方案

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

思路:

直接按照入度进行逐步处理即可。

代码:

```
from collections import deque

class Node:
    def __init__(self):
        self.win_times = 0
        self.winners = []

n, m = map(int, input().split())
nodes = [Node() for _ in range(n)]
for _ in range(m):
    a, b = map(int, input().split())
    a -= 1
    b -= 1
    nodes[a].win_times += 1
    nodes[b].winners.append(nodes[a])

q = deque([])
for node in nodes:
    if node.win_times == 0:
        q.append(node)

cnt = 0
ans = 0
while q:
    s = len(q)
    ans += cnt*s
    for _ in range(s):
        node = q.popleft()
        for winner in node.winners:
            winner.win_times -= 1
            if winner.win_times == 0:
                q.append(winner)
    cnt += 1
print(ans+100*n)
```

代码运行截图 (至少包含有"Accepted")

状态: **Accepted**

源代码

```
from collections import deque

class Node:
    def __init__(self):
        self.win_times = 0
        self.winners = []

n, m = map(int, input().split())
nodes = [Node() for _ in range(n)]
for _ in range(m):
    a, b = map(int, input().split())
    a -= 1
    b -= 1
    nodes[a].win_times += 1
    nodes[b].winners.append(nodes[a])

q = deque([])
for node in nodes:
    if node.win_times == 0:
```

基本信息

#: 49138795  
题目: 22508  
提交人: 24n2400011498  
内存: 3792kB  
时间: 30ms  
语言: Python3  
提交时间: 2025-05-13 16:21:22

## 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。

虽然模板掌握的还算可以，但是对于变形还是很难一次写对，还需要继续练习。