Assignment #D: 图 & 散列表

Updated 2042 GMT+8 May 20, 2025

2025 spring, Complied by 胡新璞，工学院

1. 题目

M17975: 用二次探查法建立散列表

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

思路：读懂题意之后比较简单。一次性读入的写法也给了（期末应该不会有这样的吧）

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
num_list = [int(i) for i in data[index:index+n]]

ans = []
table = [None] * m
for i in range(len(num_list)):
    k = num_list[i] % m
    if table[k] == None or table[k] == num_list[i]:
        table[k] = num_list[i]
        ans.append(k)
    else:
        j = 1
        found = False
        while not found:
            if not table[(k + j ** 2) % m]:
                table[(k + j ** 2) % m] = num_list[i]
                found = True
                ans.append((k + j ** 2) % m)
            elif not table[(k - j ** 2) % m]:
                table[(k - j ** 2) % m] = num_list[i]
                found = True
                ans.append((k - j ** 2) % m)
            else:
                j += 1
print(" ".join(map(str, ans)))
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
import sys
input = sys.stdin.read
data = input().split()
index = 0
n = int(data[index])
index += 1
m = int(data[index])
```

基本信息

#: 49279292
题目: 17975
提交人: 2400011037
内存: 3680kB
时间: 22ms
语言: Python3
提交时间: 2025-05-27 15:25:08

M01258: Agri-Net

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

代码：

```
import heapq

while True:
    try:
        n = int(input())
    except EOFError:
        break
    matrix = [list(map(int, input().split())) for i in range(n)]
    q = []
    dis = [0] + [100001] * (n - 1)
    val = set()
    cnt = 0
    heapq.heappush(q,(dis[0],0))
    while q:
        x, y = heapq.heappop(q)
        if y in val:
            continue
        val.add(y)
        cnt += dis[y]
        for i in range(n):
            if matrix[y][i] < dis[i]:
                dis[i] = matrix[y][i]
                heapq.heappush(q,(dis[i],i))
    print(cnt)
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
import heapq

while True:
    try:
        n = int(input())
    except EOFError:
        break
```

基本信息

#: 49280633
题目: 01258
提交人: 2400011037
内存: 4020kB
时间: 34ms
语言: Python3
提交时间: 2025-05-27 17:23:09

M3552.网络传送门旅游

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

思路：需要另外写一个处理传送门的，这个地方处理起来感觉还是有难度的。

代码：

```
import heapq
from collections import defaultdict
class Solution(object):
    def minMoves(self, matrix):
        m = len(matrix)
        n = len(matrix[0])
        directions = [[0, 1], [1, 0], [-1, 0], [0, -1]]
        portal = defaultdict(list)
        for i in range(m):
            for j in range(n):
                if matrix[i][j].isalpha():
                    portal[matrix[i][j]].append((i, j))
        q = []
        distance = {(0,0):0}
        heapq.heappush(q, (0, 0, 0))
        while q:
            step, x, y = heapq.heappop(q)
            if distance[(x,y)] < step:
                continue
            if x == m - 1 and y == n - 1:
                return step
            if matrix[x][y].isalpha():
                while portal[matrix[x][y]]:
                    nx, ny = portal[matrix[x][y]].pop()
                    distance[(nx, ny)] = step
                    heapq.heappush(q, (step, nx, ny))
            for i in range(len(directions)):
                nx = x + directions[i][0]
                ny = y + directions[i][1]
                if 0 <= nx < m and 0 <= ny < n:
                    if matrix[nx][ny] != "#":
                        if (nx, ny) not in distance or distance[(nx, ny)] > step + 1:
                            distance[(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：二维 dp 显式记录不同中转次数的状态。Bellman-Ford 算法中的“松弛”确实很巧。

代码：

```
class Solution:
    def findCheapestPrice(self, n: int, flights: List[List[int]], src: int, dst: int, k: int) -> int:
        max_step = k + 1
        dp = [[float('inf')] * n for _ in range(max_step + 1)]
        dp[0][src] = 0
        for s in range(1, max_step + 1):
            dp[s] = [float('inf')] * n
            for u, v, w in flights:
                if dp[s - 1][u] != float('inf'):
                    if dp[s][v] > dp[s - 1][u] + w:
                        dp[s][v] = dp[s - 1][u] + w
        res = min(dp[s][dst] for s in range(1, max_step + 1))
        return res if res != 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：

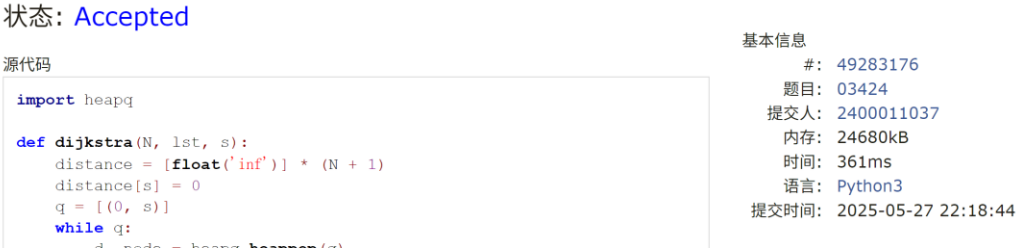
代码：

```
import heapq

def dijkstra(N, lst, s):
    distance = [float('inf')] * (N + 1)
    distance[s] = 0
    q = [(0, s)]
    while q:
        d, node = heapq.heappop(q)
        if d > distance[node]:
            continue
        for x,y in lst[node]:
            n_distance = distance[node] + y
            if n_distance < distance[x]:
                distance[x] = n_distance
                heapq.heappush(q, (n_distance, x))
    return distance

N, M = map(int, input().split())
lst = [[] for _ in range(N + 1)]
for _ in range(M):
    A, B, c = map(int, input().split())
    lst[A].append((B, c))
ans = dijkstra(N, lst, 1)
print(ans[-1])
```

代码运行截图 （至少包含有"Accepted"）



M22508:最小奖金方案

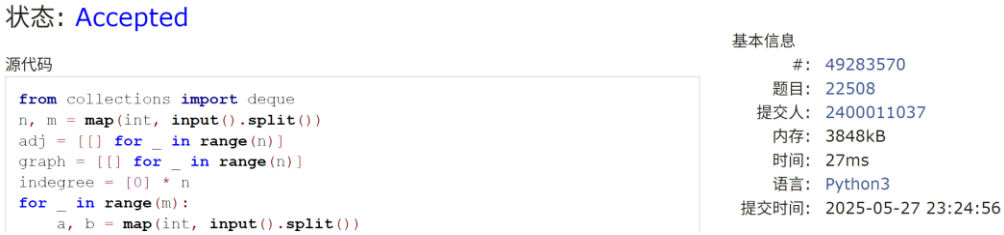
topological order, http://cs101.openjudge.cn/practice/22508/

思路：拓扑排序掌握得还是不理想，又问了 d 老师，看了好一会题解和讲义。

代码：

```
from collections import deque
n, m = map(int, input().split())
adj = [[] for _ in range(n)]
graph = [[] for _ in range(n)]
indegree = [0] * n
for _ in range(m):
    a, b = map(int, input().split())
    adj[a].append(b)
    graph[b].append(a)
    indegree[a] += 1
q = deque()
for i in range(n):
    if indegree[i] == 0:
        q.append(i)
top_order = []
while q:
    u = q.popleft()
    top_order.append(u)
    for v in graph[u]:
        indegree[v] -= 1
        if indegree[v] == 0:
            q.append(v)
bonus = [100] * n
for u in top_order:
    max_val = 0
    for v in adj[u]:
        if bonus[v] > max_val:
            max_val = bonus[v]
    if adj[u]:
        bonus[u] = max_val + 1
print(sum(bonus))
```

代码运行截图 （至少包含有"Accepted"）



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。

感觉题目没有特别困难的，思路和代码都没有完全卡手的地方，但是综合性也是非常强的，也有较多跟模板不一样的东西，所以还是有些难以战胜。下周就要机考了，希望最后一周能再巩固一下，把 cheatsheet 整理好，争取前面掌握的也别有生疏。