Assignment #D: 图 & 散列表
Updated 2042 GMT+8 May 20, 2025
2025 spring, Complied by 李振硕、信息管理系

1. 题目
M17975: 用二次探查法建立散列表
http://cs101.openjudge.cn/practice/17975/
<mark>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。</mark>
```python
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：
```python

```
代码运行截图 <mark>（至少包含有"Accepted"）</mark>
M01258: Agri-Net
MST, http://cs101.openjudge.cn/practice/01258/
思路：
代码：

状态: Accepted

源代码

```python
import sys
import heapq

def read_case():
    lines = []
    while True:
        line = sys.stdin.readline()
        if not line:
            break
        if line.strip().isdigit():
            if lines:
                yield lines
                lines = []
        lines.append(line.strip())
    if lines:
        yield lines

def parse_matrix(lines):
    N = int(lines[0])
    data = ' '.join(lines[1:]).split()
    matrix = []
    idx = 0
    for i in range(N):
        row = list(map(int, data[idx:idx+N]))
        matrix.append(row)
        idx += N
    return N, matrix

def prim(N, matrix):
    visited = [False] * N
    min_cost = [float('inf')] * N
    min_cost[0] = 0
    heap = [(0, 0)]  # (cost, node)
    total = 0

    while heap:
        cost, u = heapq.heappop(heap)
        if visited[u]:
            continue
        visited[u] = True
        total += cost
        for v in range(N):
            if not visited[v] and matrix[u][v] < min_cost[v]:
                min_cost[v] = matrix[u][v]
                heapq.heappush(heap, (matrix[u][v], v))
    return total

# 主处理逻辑
for case_lines in read_case():
    N, matrix = parse_matrix(case_lines)
    print(prim(N, matrix))
```

基本信息

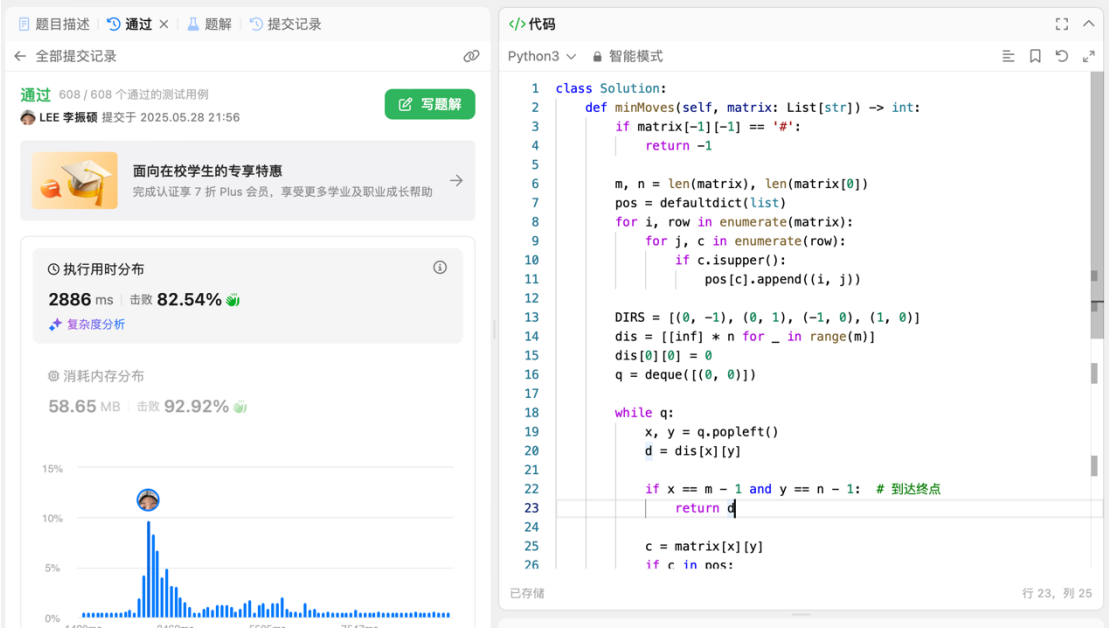
#:	49290119
题目:	01258
提交人:	24n2300093007
内存:	4832kB
时间:	28ms
语言:	Python3
提交时间:	2025-05-28 21:31:35

M3552.网络传送门旅游

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

思路:

代码:

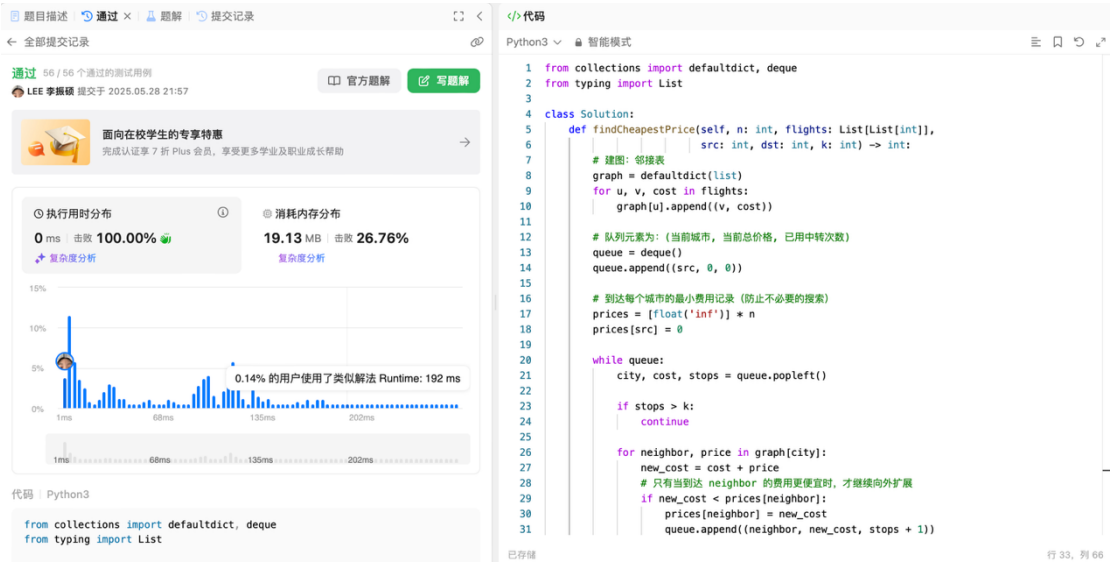


M787.K 站中转内最便宜的航班

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

思路：

代码：



M03424: Candies

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

思路：

代码：

#49296646提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```python
import heapq
from collections import defaultdict

def solve_with_dijkstra(n, m, constraints):
    graph = defaultdict(list)
    for a, b, c in constraints:
        graph[a].append((b, c))  # candies[b] <= candies[a] + c → edge

    dist = [float('inf')] * (n + 1)
    dist[1] = 0  # snoopy (node 1) gets 0 candies

    pq = [(0, 1)]  # (distance, node)
    while pq:
        d, u = heapq.heappop(pq)
        if d > dist[u]:
            continue
        for v, w in graph[u]:
            if dist[v] > dist[u] + w:
                dist[v] = dist[u] + w
                heapq.heappush(pq, (dist[v], v))

    return dist[n] - dist[1]

# 主程序输入处理
if __name__ == "__main__":
    import sys
    input = sys.stdin.read
    data = input().split()

    n, m = int(data[0]), int(data[1])
    constraints = []
    idx = 2
    for _ in range(m):
        a, b, c = int(data[idx]), int(data[idx+1]), int(data[idx+2])
        constraints.append((a, b, c))
        idx += 3

    print(solve_with_dijkstra(n, m, constraints))
```

基本信息

#:	49296646
题目:	03424
提交人:	24n2300093007
内存:	61760kB
时间:	311ms
语言:	Python3
提交时间:	2025-05-29 17:49:48

English 帮助 关于

M22508:最小奖金方案

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

思路:

代码:

#49296670提交状态

查看　提交　统计　提问

状态: Accepted

源代码

```python
from collections import defaultdict, deque

def min_total_award(n, m, matches):
    # 构建反向图（败方 → 胜方）
    graph = defaultdict(list)
    indegree = [0] * n
    for a, b in matches:
        graph[b].append(a)  # b → a 表示 a 的奖金必须比 b 高
        indegree[a] += 1

    # 初始化拓扑排序队列
    queue = deque()
    dist = [0] * n  # 奖金增量部分（最终是 100 + dist[i]）
    for i in range(n):
        if indegree[i] == 0:
            queue.append(i)

    # 拓扑排序 + DP 计算每个节点的最长路径长度
    while queue:
        u = queue.popleft()
        for v in graph[u]:
            if dist[v] < dist[u] + 1:
                dist[v] = dist[u] + 1
            indegree[v] -= 1
            if indegree[v] == 0:
                queue.append(v)

    total_award = sum(100 + d for d in dist)
    return total_award

# 主函数处理输入
if __name__ == "__main__":
    import sys
    input = sys.stdin.read
    data = input().split()

    n = int(data[0])
    m = int(data[1])
    matches = []
    idx = 2
    for _ in range(m):
        a = int(data[idx])
        b = int(data[idx + 1])
        matches.append((a, b))
        idx += 2

    print(min_total_award(n, m, matches))
```

基本信息

#:	49296670
题目:	22508
提交人:	24n2300093007
内存:	4240kB
时间:	24ms
语言:	Python3
提交时间:	2025-05-29 17:51:20

English　帮助　关于

2. 学习总结和收获

<mark> “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 、

这次作业对我来说是最难的，用二次探查法建立散列表这一道题想了很久，但现在还没 AC，想以后再补。Dijkstra, topological order 还没学好。机考前一定要复习完。