

Assignment #D: 图 & 散列表

Updated 2042 GMT+8 May 20, 2025

2025 spring, Compiled by 颜鼎堃 工学院

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。

- 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。
- 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。

```
1 import sys
2 input = sys.stdin.read
3 data = input().split()
4 index = 0
5 n = int(data[index])
6 index += 1
7 m = int(data[index])
8 index += 1
9 num_list = [int(i) for i in data[index:index+n]]
```

Python

思路:

- 被重复数据卡了，没想到有重复的

代码:

```
1 from sys import stdin
2 readin = map(int, stdin.read().split())
3 N = next(readin)
4 M = next(readin)
5 hashtable = [0] * M
6 pos = [0] * N
7 nums = []
8 for i in range(N):
9     nums.append(next(readin))
10 for i in range(N):
11     j = 0
12     h = nums[i] % M
13     p = h
14     while hashtable[p] and hashtable[p] != nums[i]:
15         p = (h + (-1 if j % 2 else 1) * (j // 2 + 1)**2) % M
16         j += 1
```

Python

```
17     pos[i] = p
18     hashtable[p] = nums[i]
19     print(*pos)
```

代码运行截图 (至少包含有"Accepted")

OpenJudge 题目ID, 标题, 描述 24n2400011125 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#49241694提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
from sys import stdin
readin = map(int, stdin.read().split())
N = next(readin)
M = next(readin)
hashtable = [0] * M
pos = [0] * N
nums = []
for i in range(N):
    nums.append(next(readin))
for i in range(N):
    j = 0
    h = nums[i] % M
    p = h
    while hashtable[p] and hashtable[p] != nums[i]:
        p = (h + (-1 if j % 2 else 1) * (j // 2 + 1)**2) % M
        j += 1
    pos[i] = p
    hashtable[p] = nums[i]
print(*pos)
```

基本信息

#: 49241694

题目: 17975

提交人: 颜鼎堃(24n2400011125)

内存: 3648kB

时间: 21ms

语言: Python3

提交时间: 2025-05-23 16:20:42

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

M01258: Agri-Net

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

思路:

- 最小生成树
- 之前我想了好久最小生成树的权值是不是等于全源最短路中的最小一行的和, 应该是不等于

代码:

```
1 from sys import stdin
2 from heapq import heappush, heappop, heapify
3 readin = map(int, stdin.read().split())
4 while True:
5     try:
6         N = next(readin)
7     except StopIteration:
8         break
9     adj_mat = []
10    for i in range(N):
11        adj_mat.append([])
12        for j in range(N):
13            adj_mat[-1].append(next(readin))
14    heap = [(adj_mat[0][i], i) for i in range(1, N)]
15    heapify(heap)
16    vis = {0}
17    ans = 0
18    while True:
19        d, i = heappop(heap)
20        if i in vis:
21            continue
22        ans += d
```

Python

```
23     vis.add(i)
24     if len(vis) == N:
25         break
26     for n in range(N):
27         if n not in vis:
28             heappush(heap, (adj_mat[i][n], n))
29 print(ans)
30
```

代码运行截图 (至少包含有"Accepted")

OpenJudge

题目ID, 标题, 描述

24n2400011125 信箱 账号

 **CS101 / 题库 (包括计概、数算题目)**

题目 排名 状态 提问

#49244370提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from sys import stdin
from heapq import heappush, heappop, heapify
readin = map(int, stdin.read().split())
while True:
    try:
        N = next(readin)
    except StopIteration:
        break
    adj_mat = []
    for i in range(N):
        adj_mat.append([])
        for j in range(N):
            adj_mat[-1].append(next(readin))
    heap = [(adj_mat[0][i], i) for i in range(1, N)]
    heapify(heap)
    vis = {0}
    ans = 0
    while True:
        d, i = heappop(heap)
        if i in vis:
            continue
        ans += d
        vis.add(i)
        if len(vis) == N:
            break
        for n in range(N):
            if n not in vis:
                heappush(heap, (adj_mat[i][n], n))
    print(ans)
```

基本信息

#: 49244370
题目: 01258
提交人: 颜鼎堃(24n2400011125)
内存: 5564kB
时间: 39ms
语言: Python3
提交时间: 2025-05-23 20:12:05

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M3552.网络传送门旅游

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

思路:

- 你要干啥
- 你卡你那常数你想干啥
- 改一晚上也改不出来, 出题人你想干啥
- 抄的答案

代码:

```
1  from typing import *
2  from collections import deque, defaultdict
3  class Solution:
4      def minMoves(self, matrix: List[str]) -> int:
5          if matrix[-1][-1] == '#':
6              return -1
7          inf = float("inf")
8          m, n = len(matrix), len(matrix[0])
9          pos = defaultdict(list)
10         for i, row in enumerate(matrix):
11             for j, c in enumerate(row):
12                 if c.isupper():
13                     pos[c].append((i, j))
14
```

Python

[illegible]

代码运行截图 (至少包含有"Accepted")

	所有状态 ▾	所有语言 ▾	执行用时	消耗内存	备注	⚙
11	通过 2 小时前	Python3	🕒 2983 ms	💾 58.7 MB		
10	超出时间限制 2 小时前	Python3	🕒 N/A	💾 N/A		
9	超出时间限制 2 小时前	Python3	🕒 N/A	💾 N/A		
8	超出时间限制 1 天前	Python3	🕒 N/A	💾 N/A		
7	超出时间限制 1 天前	Python3	🕒 N/A	💾 N/A		
6	超出时间限制 1 天前	Python3	🕒 N/A	💾 N/A		
5	超出时间限制 1 天前	Python3	🕒 N/A	💾 N/A		
4	超出时间限制 1 天前	Python3	🕒 N/A	💾 N/A		
3	解答错误 1 天前	Python3	🕒 N/A	💾 N/A		
2	执行出错 1 天前	Python3	🕒 N/A	💾 N/A		
1	解答错误 1 天前	Python3	🕒 N/A	💾 N/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:

- 和月考题很像

代码:

```
1 from typing import *
2 from heapq import heappush, heappop
3 class Solution:
4     def findCheapestPrice(self, n: int, flights: List[List[int]], src: int, dst: int, k: int) -> int:
5         adj_mat = [[100001 for i in range(n)] for j in range(n)]
6         for s, c, p in flights:
7             adj_mat[s][c] = p
8         dist = [[100001 for i in range(k + 2)] for j in range(n)]
9         heap = [(0, 0, src)]
10        ans = -1
11        while heap:
12            c, ret, node = heappop(heap)
13            if dist[node][ret] < c:
14                continue
15            if node == dst:
16                ans = c
17                break
18            for i in range(n):
19                if adj_mat[node][i] < 10001 and ret <= k and c + adj_mat[node][i] < dist[i][ret + 1]:
20                    dist[i][ret + 1] = c + adj_mat[node][i]
21                    heappush(heap, (dist[i][ret + 1], ret + 1, i))
```

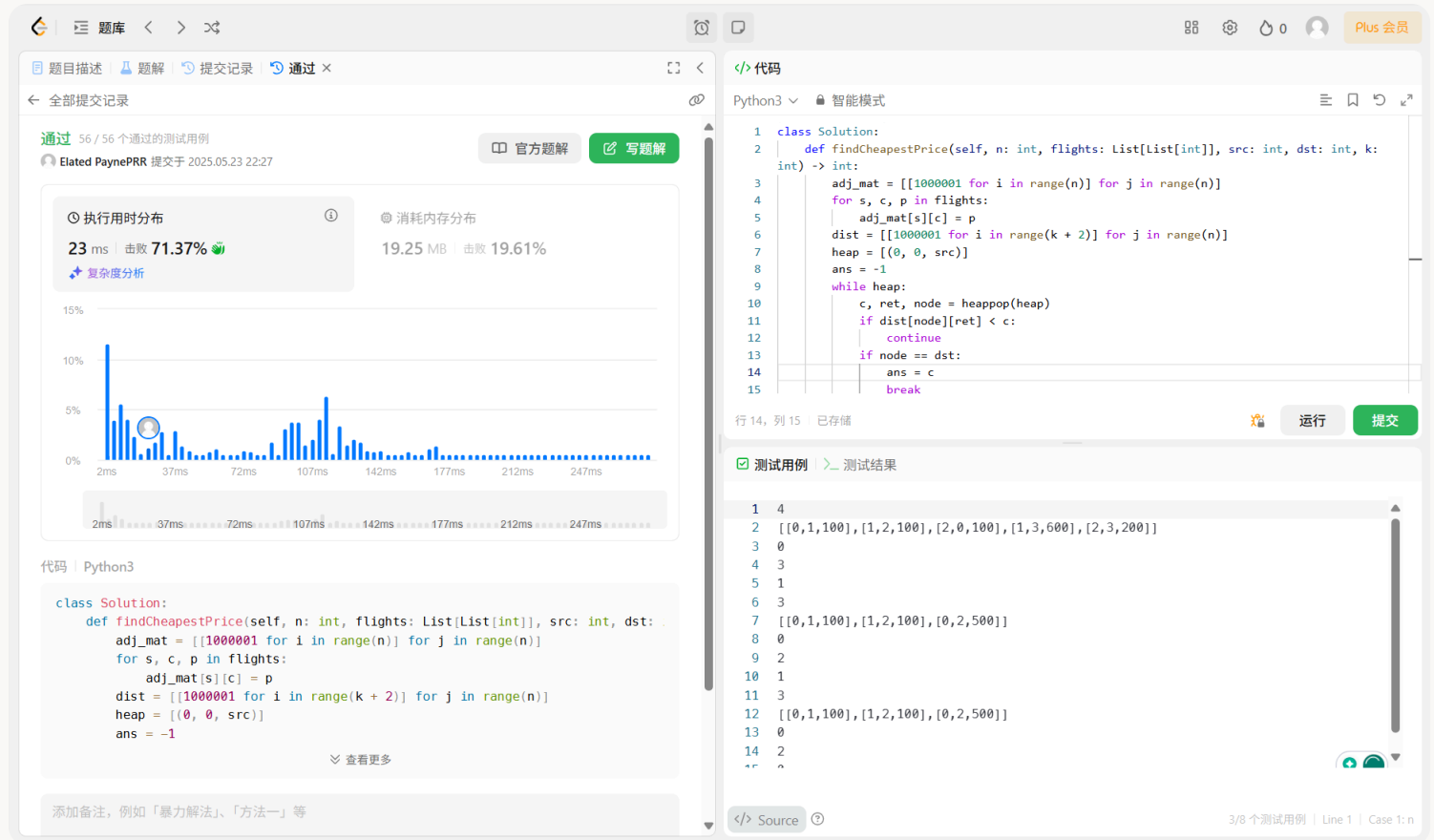
Python

```

22         return ans
23
24
25
26 if __name__ == "__main__":
27     sol = Solution()
28     print(sol.findCheapestPrice(n = 4, flights = [[0,1,100],[1,2,100],[2,0,100],[1,3,600],[2,3,200]],
29         src = 0, dst = 3, k = 1))

```

代码运行截图 (至少包含有"Accepted")



M03424: Candies

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

思路:

- 先省略100字对作者的情感输出
- 我写的代码全部WA, AI写的代码 (包括手上拿着答案的小北explore在内) 全部TLE

代码:

- Dijkstra算法, WA

```

1 from heapq import heappush, heappop
2
3 N, M = map(int, input().split())
4 adj_mat = [[float("inf")] * N for _ in range(N)]
5 for _ in range(M):
6     a, b, c = map(int, input().split())
7     adj_mat[a-1][b-1] = c
8
9 dist = [float("inf")] * N
10 dist[0] = 0
11 heap = [(0, 0)]
12 visited = [False] * N
13
14 while heap:

```

Python

```

15     d, node = heappop(heap)
16     if node == N - 1:
17         break
18     if visited[node]:
19         continue
20     visited[node] = True
21     for i in range(N):
22         if adj_mat[node][i] != float("inf") and d + adj_mat[node][i] < dist[i]:
23             dist[i] = d + adj_mat[node][i]
24             heappush(heap, (dist[i], i))
25
26     print(dist[-1])

```

- Bellman-Ford算法, WA

```

1  from collections import deque
2  N, M = map(int, input().split())
3  edge = [[1e5 for i in range(N)] for j in range(N)]
4  for i in range(M):
5      a, b, c = map(int, input().split())
6      edge[a-1][b-1] = c
7  queue = deque()
8  vis = [False] * (N + 1)
9  queue.append(0)
10 while queue:
11     node = queue.popleft()
12     vis[node] = False
13     for i in range(N):
14         if edge[node][i] < 1e5 and edge[0][i] > edge[0][node] + edge[node][i]:
15             edge[0][i] = edge[0][node] + edge[node][i]
16             if not vis[i]:
17                 queue.append(i)
18                 vis[i] = True
19 print(edge[0][-1])

```

Python

- Floyd-Warshall算法, WA

```

1  N, M = map(int, input().split())
2  edge = [[1e11 for i in range(N)] for j in range(N)]
3  for i in range(M):
4      a, b, c = map(int, input().split())
5      edge[a-1][b-1] = c
6  for k in range(N):
7      for i in range(N):
8          for j in range(N):
9              edge[i][j] = min(edge[i][j], edge[i][k] + edge[k][j])
10
11 print(edge[0][-1])

```

Python

- 好，就在我编辑这份即将上交的作业的时候，我又交了一次，ac了，我不过是把第一次提交的代码中邻接矩阵改成了邻接表
- 那我不禁要问了，是作者分糖的时候左脑攻击右脑尖尖开始思考了吗？刚说完右边的哥们只准比我多一个，过了三秒钟说，右边的哥们只准比我多两个，那我听哪个？用邻接矩阵就是，听你最新版的指令，用邻接表那就是我管你这的那的我听最小的

```

1  from heapq import heappush, heappop
2  N, M = map(int, input().split())
3  adj_mat = [[] for j in range(N)]
4  heap = []
5  for i in range(M):
6      a, b, c = map(int, input().split())
7      adj_mat[a-1].append((b-1, c))

```

Python

```
8 dist = [1e11 for i in range(N)]
9 heap = [(0, 0)]
10 while heap:
11     d, node = heappop(heap)
12     if node == N - 1:
13         dist[-1] = d
14         break
15     for i, w in adj_mat[node]:
16         if d + w < dist[i]:
17             dist[i] = d + w
18             heappush(heap, (dist[i], i))
19
20 print(dist[-1])
```

- 更难听的话还是不骂了
- 耽误我一下午

代码运行截图 (至少包含有"Accepted")

你的提交记录

#	结果	时间
25	Accepted	2025-05-24
24	Time Limit	2025-05-24
	Exceeded	
23	Time Limit	2025-05-24
	Exceeded	
22	Accepted	2025-05-24
21	Wrong Answer	2025-05-24
20	Time Limit	2025-05-24
	Exceeded	
19	Wrong Answer	2025-05-24
18	Time Limit	2025-05-24
	Exceeded	
17	Time Limit	2025-05-24
	Exceeded	
16	Wrong Answer	2025-05-24
15	Wrong Answer	2025-05-24
14	Time Limit	2025-05-24
	Exceeded	
13	Wrong Answer	2025-05-24
12	Wrong Answer	2025-05-24
11	Time Limit	2025-05-24
	Exceeded	
10	Compile Error	2025-05-24
9	Time Limit	2025-05-24
	Exceeded	
8	Wrong Answer	2025-05-24
7	Wrong Answer	2025-05-24
6	Time Limit	2025-05-24
	Exceeded	
5	Wrong Answer	2025-05-24
4	Wrong Answer	2025-05-24
3	Wrong Answer	2025-05-24
2	Wrong Answer	2025-05-24
1	Wrong Answer	2025-05-24

- 第22次提交的是网上搜到的能通过的C++代码

M22508:最小奖金方案

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

思路:

- 拓扑排序, 和月考题有一定的相似之处

代码:

```
1 from collections import deque
2 n, m = map(int, input().split())
```

Python

```

3  d = [[] for i in range(n)]
4  deg = [0 for i in range(n)]
5  prize = [0 for i in range(n)]
6  queue = set(range(n))
7  for i in range(m):
8      a, b = map(int, input().split())
9      d[b].append(a)
10     deg[a] += 1
11     queue -= {a}
12 queue = deque(queue)
13 while queue:
14     node = queue.popleft()
15     for i in d[node]:
16         deg[i] -= 1
17         if not deg[i]:
18             prize[i] = prize[node] + 1
19             queue.append(i)
20
21 print(100 * n + sum(prize))

```

OpenJudge

题目ID, 标题, 描述

24n2400011125

信箱

账号



CS101 / 题库 (包括计概、数算题目)

题目

排名

状态

提问

#49245876提交状态

查看

提交

统计

提问

状态: Accepted

基本信息

源代码

#: 49245876
题目: 22508
提交人: 颜鼎堃(24n2400011125)
内存: 4008kB
时间: 27ms
语言: Python3
提交时间: 2025-05-23 23:07:26

```
from collections import deque
n, m = map(int, input().split())
d = [[] for i in range(n)]
deg = [0 for i in range(n)]
prize = [0 for i in range(n)]
queue = set(range(n))
for i in range(m):
    a, b = map(int, input().split())
    d[b].append(a)
    deg[a] += 1
    queue -= {a}
queue = deque(queue)
while queue:
    node = queue.popleft()
    for i in d[node]:
        deg[i] -= 1
        if not deg[i]:
            prize[i] = prize[node] + 1
            queue.append(i)

print(100 * n + sum(prize))
```

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

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。