

Assignment #D: 图 & 散列表

Updated 2042 GMT+8 May 20, 2025

2025 spring, Compiled by 同学的姓名、院系

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:

看了群聊天记录才知道有重复数据

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]]
lst=[None]*m
ans=[]
visited=set()
for i in num_list:
    if i in visited:
        ans.append(ans[num_list.index(i)])
        continue
    visited.add(i)
    record=i%m
    if lst[record]==None:
        lst[record]=i
        ans.append(record)
    else:
        t=1
        cnt=0
        while True:
            if cnt%2==0:
                if lst[(record+t**2)%m]==None:
                    lst[(record+t**2)%m]=i
                    ans.append((record+t**2)%m)
                    break
            else:
                if lst[(record-t**2)%m]==None:
                    lst[(record-t**2)%m]=i
                    ans.append((record-t**2)%m)
                    break
            if cnt%2!=0:
                t+=1
            cnt+=1
print(*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])
index += 1
num_list = [int(i) for i in data[index:index+n]]
lst=[None]*m
ans=[]
visited=set()
for i in num_list:
    if i in visited:
```

M01258: Agri-Net

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

思路:

代码:

```
import heapq
from collections import defaultdict
def prim(d):
    visited={0}
    pq=d[0]
    heapq.heapify(pq)
    total_cost=0
    edge_used=1
    while pq and edge_used<len(d):
        distance,node=heapq.heappop(pq)
        if node not in visited:
            visited.add(node)
            total_cost+=distance
            edge_used+=1
            for nei_distance,neighbor in d[node]:
                if neighbor not in visited:
                    heapq.heappush(pq,[nei_distance,neighbor])
    return total_cost

while True:
```

```

try:
    n=int(input())
    d=defaultdict(list)
    for _ in range(n):
        s=[int(x) for x in input().split()]
        for i in range(n):
            d[_].append([s[i],i]) #distance,end
    print(prim(d))
except EOFError:
    break

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

import heapq
from collections import defaultdict
def prim(d):
    visited={0}
    pq=d[0]
    heapq.heapify(pq)
    total_cost=0
    edge_used=1
    while pq and edge_used<len(d):
        distance,node=heapq.heappop(pq)

```

M3552.网络传送门旅游

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

思路：差点忘了deque中还有appendleft

代码：

```

from collections import defaultdict,deque
class Solution(object):
    def minMoves(self, matrix):
        """
        :type matrix: List[str]
        :rtype: int
        """
        n=len(matrix)
        m=len(matrix[0])

```

```

door=defaultdict(list)
for i in range(n):
    for j in range(m):
        if matrix[i][j] not in '.#':
            door[matrix[i][j]].append((i,j))
#print(door)
d=[(0,1),(0,-1),(1,0),(-1,0)]
def bfs(x,y):
    visited={(x,y)}
    pq=deque([(0,x,y)]) #step,x,y
    while pq:
        step,x,y=pq.popleft()
        if x==n-1 and y==m-1:
            return step
        if matrix[x][y] in door and matrix[x][y] not in visited:
            for nx,ny in door[matrix[x][y]]:
                if (nx,ny) != (x,y):
                    pq.appendleft((step,nx,ny))
            visited.add(matrix[x][y])
        for a,b in d:
            nx,ny=x+a,y+b
            if 0<=nx<n and 0<=ny<m:
                if matrix[nx][ny]!='#' and (nx,ny) not in visited:
                    visited.add((nx,ny))
                    pq.append((step+1,nx,ny))
    return -1
return bfs(0,0)

```

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：

代码：

```
class Solution(object):
    def findCheapestPrice(self, n, flights, src, dst, k):
        """
        :type n: int
        :type flights: List[List[int]]
        :type src: int
        :type dst: int
        :type k: int
        :rtype: int
        """
        distance=[float('inf')]*n
        distance[src]=0
        for _ in range(k+1):
            cur=distance[:]
```

```

for st,ed,cost in flights:
    if distance[st]+cost<distance[ed]:
        cur[ed]=min(cur[ed],distance[st]+cost)
        #min确保在本轮中多个航班可能更新同一个节点的问题。
        #保护原来的cur值,如果有多个边可以到达ed,应该选最小的那个
    distance=cur
if distance[dst]==float('inf'):
    return -1
else:
    return distance[dst]

```

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
from collections import defaultdict
def dijkstra(d):
    pq=[(0,0)] #weigh,st
    visited=set()
    dist={i:float('inf') for i in range(n)}
    dist[0]=0

```

```

while pq:
    weigh,st=heapq.heappop(pq)
    if st==n-1:
        return weigh
    if st in visited:
        continue
    visited.add(st)
    for i in d[st]:
        if i[0] not in visited and dist[i[0]]>weigh+i[1]:
            heapq.heappush(pq,(weigh+i[1],i[0]))
            dist[i[0]]=weigh+i[1]
    return dist[n-1]

n,m=[int(x) for x in input().split()]
d=defaultdict(list)
for _ in range(m):
    a,b,c=[int(x) for x in input().split()]
    d[a-1].append((b-1,c))
print(dijkstra(d))

```

代码运行截图 (至少包含有"Accepted")

#49238944提交状态

状态: Accepted

源代码

```

import heapq
from collections import defaultdict
def dijkstra(d):
    pq=[(0,0)] #weigh,st
    visited=set()
    dist={i:float('inf') for i in range(n)}
    dist[0]=0
    while pq:
        weigh,st=heapq.heappop(pq)
        if st==n-1:
            return weigh
        if st in visited:

```


M22508:最小奖金方案

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

思路:

代码:

```
from collections import deque,defaultdict
n,m=[int(x) for x in input().split()]
graph=defaultdict(list)
for _ in range(m):
    a,b=[int(x) for x in input().split()]
    graph[b].append(a)
indegree=defaultdict(int)
q=deque()
result=[]
for u in graph:
    for v in graph[u]:
        indegree[v]+=1
for u in graph:
    if indegree[u]==0:
        q.append((u,0))
while q:
    u,cost=q.popleft()
    result.append((u,cost))
    for v in graph[u]:
        indegree[v]-=1
        if indegree[v]==0:
            q.append((v,cost+1))
ans=100*n
for i in result:
    ans+=i[1]
print(ans)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque, defaultdict
n, m = [int(x) for x in input().split()]
graph = defaultdict(list)
for _ in range(m):
    a, b = [int(x) for x in input().split()]
    graph[b].append(a)
```

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。

感觉图部分掌握得一点也不熟啊啊啊，机考怎么办，只剩一个多星期了，感觉学不会