Assignment #D: 图 & 散列表

Updated 2042 GMT+8 May 20, 2025

2025 spring, Complied by **袁奕 2400010766 数院**

> **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。

```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：注意输入的关键词可能有重复!!! 例如

> ```
> 2 7
> 12 12
> ```

输出的结果应该是 `5 5`

代码：

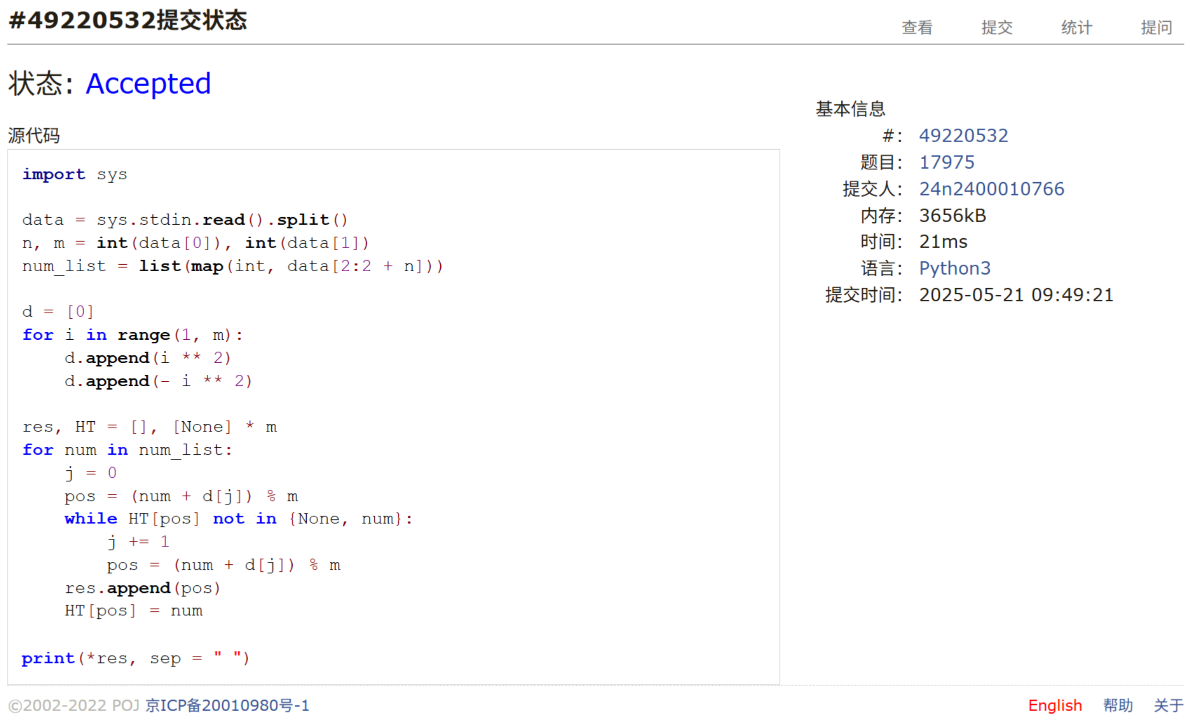
```python
import sys

data = sys.stdin.read().split()
n, m = int(data[0]), int(data[1])
num_list = list(map(int, data[2:2 + n]))

d = [0]
for i in range(1, m):
    d.append(i ** 2)
    d.append(- i ** 2)

res, HT = [], [None] * m
for num in num_list:
    j = 0
    pos = (num + d[j]) % m
    while HT[pos] not in {None, num}:
        j += 1
        pos = (num + d[j]) % m
    res.append(pos)
    HT[pos] = num

print(*res, sep = " ")
```



#49220532提交状态

状态: Accepted

源代码

```python
import sys

data = sys.stdin.read().split()
n, m = int(data[0]), int(data[1])
num_list = list(map(int, data[2:2 + n]))

d = [0]
for i in range(1, m):
    d.append(i ** 2)
    d.append(- i ** 2)

res, HT = [], [None] * m
for num in num_list:
    j = 0
    pos = (num + d[j]) % m
    while HT[pos] not in {None, num}:
        j += 1
        pos = (num + d[j]) % m
    res.append(pos)
    HT[pos] = num

print(*res, sep = " ")
```

基本信息

#:	49220532
题目:	17975
提交人:	24n2400010766
内存:	3656kB
时间:	21ms
语言:	Python3
提交时间:	2025-05-21 09:49:21

©2002-2022 POJ 京ICP备20010980号-1

M01258: Agri-Net

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

思路：不断往MST中添加Vertex (greedy思想, 选距离 *现有MST* 权值最小的Vertex)

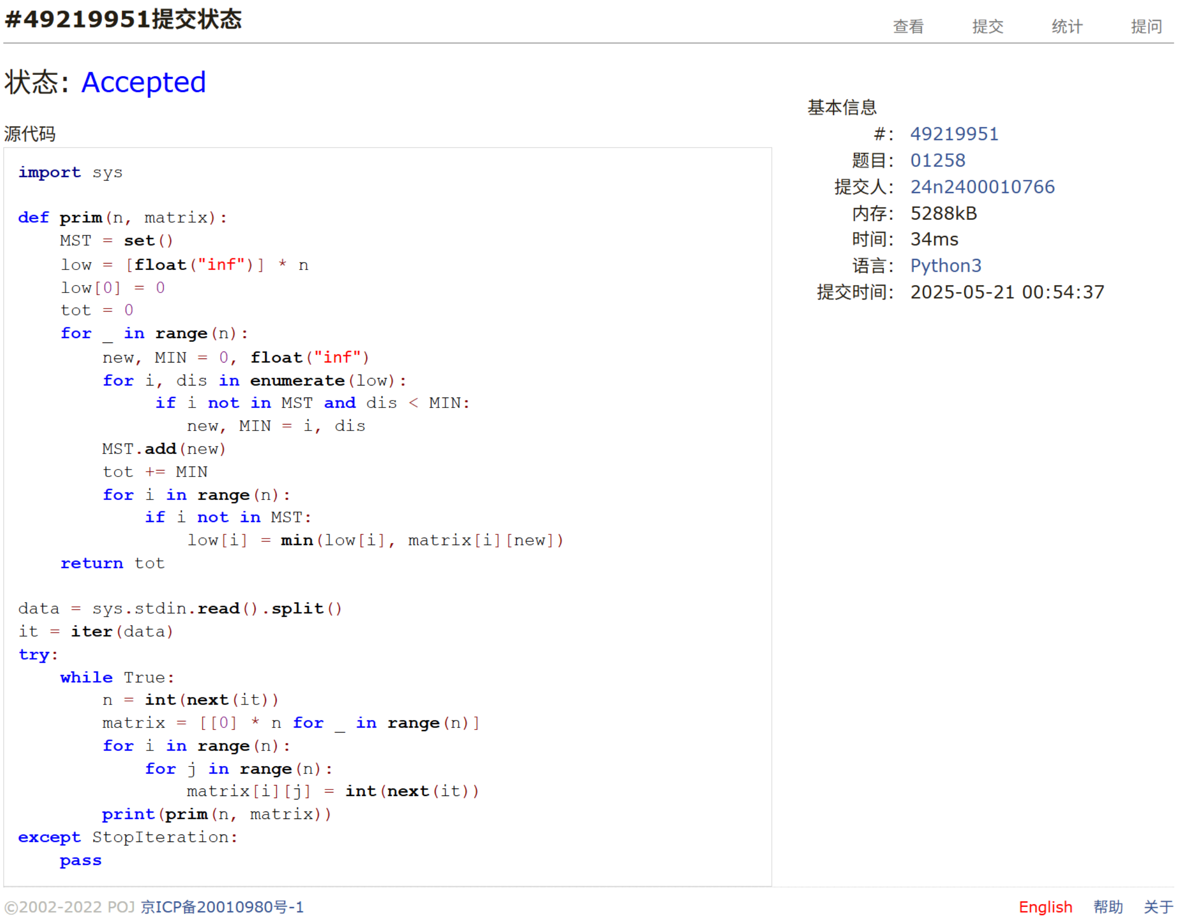
`low[k]` 表示当前 MST 距离 `k` 点的最小权值. 每次更新新版 MST 距离 `k` 点的最小权值.

代码:

```python
import sys

def prim(n, matrix):
    MST = set()
    low = [float("inf")] * n
    low[0] = 0
    tot = 0
    for _ in range(n):
        new, MIN = 0, float("inf")
        for i, dis in enumerate(low):
             if i not in MST and dis < MIN:
                new, MIN = i, dis
        MST.add(new)
        tot += MIN
        for i in range(n):
            if i not in MST:
                low[i] = min(low[i], matrix[i][new])
    return tot

data = sys.stdin.read().split()
it = iter(data)
try:
    while True:
        n = int(next(it))
        matrix = [[0] * n for _ in range(n)]
        for i in range(n):
            for j in range(n):
                matrix[i][j] = int(next(it))
        print(prim(n, matrix))
except StopIteration:
    pass
```



#49219951提交状态

状态: Accepted

源代码

```
import sys

def prim(n, matrix):
    MST = set()
    low = [float("inf")] * n
    low[0] = 0
    tot = 0
    for _ in range(n):
        new, MIN = 0, float("inf")
        for i, dis in enumerate(low):
            if i not in MST and dis < MIN:
                new, MIN = i, dis
        MST.add(new)
        tot += MIN
        for i in range(n):
            if i not in MST:
                low[i] = min(low[i], matrix[i][new])
    return tot

data = sys.stdin.read().split()
it = iter(data)
try:
    while True:
        n = int(next(it))
        matrix = [[0] * n for _ in range(n)]
        for i in range(n):
            for j in range(n):
                matrix[i][j] = int(next(it))
        print(prim(n, matrix))
except StopIteration:
    pass
```

基本信息

#:	49219951
题目:	01258
提交人:	24n2400010766
内存:	5288kB
时间:	34ms
语言:	Python3
提交时间:	2025-05-21 00:54:37

©2002-2022 POJ 京ICP备20010980号-1

M3552.网络传送门旅游

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

思路：

代码：

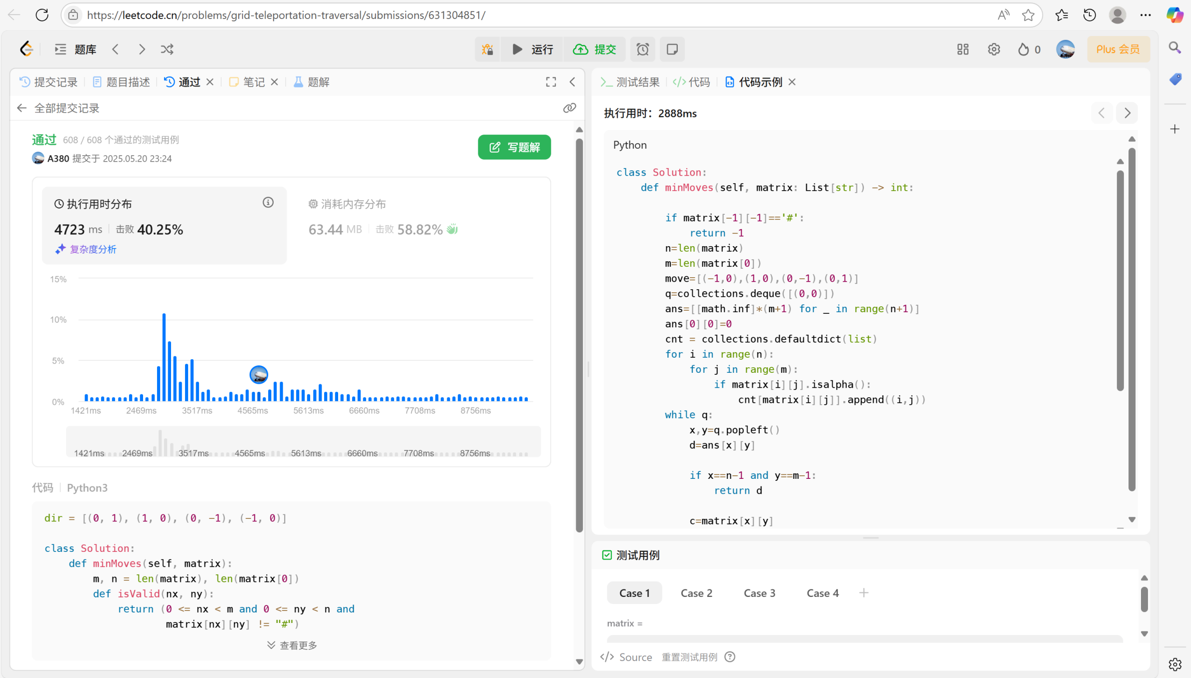
```
dir = [(0, 1), (1, 0), (0, -1), (-1, 0)]

class Solution:
    def minMoves(self, matrix):
        m, n = len(matrix), len(matrix[0])
        def isValid(nx, ny):
            return 0 <= nx < m and 0 <= ny < n and matrix[nx][ny] != "#"

        door = defaultdict(set)
        for i in range(m):
            for j in range(n):
                if matrix[i][j] not in "#.":
                    door[matrix[i][j]].add((i,j))

        def visit_door(lx, ly):
            if door[matrix[lx][ly]] != set():
                for nx, ny in door[matrix[lx][ly]]:
                    que.append((nx, ny))
                    dis[nx][ny] = dis[lx][ly]
                door[matrix[lx][ly]] = set()

        dis = [[float("inf")] * n for _ in range(m)]
        dis[0][0] = 0
        que = deque([(0,0)])
        visit_door(0,0)
        while que:
            lx, ly = que.popleft()
            if lx == m - 1 and ly == n - 1:
                return dis[m - 1][n - 1]
            for dx, dy in dir:
                nx, ny = lx + dx, ly + dy
                ns = dis[lx][ly] + 1
                if isValid(nx, ny) and ns < dis[nx][ny]:
                    que.append((nx, ny))
                    dis[nx][ny] = ns
                    visit_door(nx, ny)
        return -1
```



通过 608 / 608 个通过的测试用例

A380 提交于 2025.05.20 23:24

执行用时分布

4723 ms | 击败 40.25%

复杂度分析

消耗内存分布

63.44 MB | 击败 58.82%

代码 | Python3

```
dir = [(0, 1), (1, 0), (0, -1), (-1, 0)]

class Solution:
    def minMoves(self, matrix):
        m, n = len(matrix), len(matrix[0])
        def isValid(nx, ny):
            return (0 <= nx < m and 0 <= ny < n and
                    matrix[nx][ny] != "#")
```

执行用时：2888ms

Python

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

        if matrix[-1][-1]=='#':
            return -1
        n=len(matrix)
        m=len(matrix[0])
        move=[(-1,0),(1,0),(0,-1),(0,1)]
        q=collections.deque([(0,0)])
        ans=[[math.inf]*(m+1) for _ in range(n+1)]
        ans[0][0]=0
        cnt = collections.defaultdict(list)
        for i in range(n):
            for j in range(m):
                if matrix[i][j].isalpha():
                    cnt[matrix[i][j]].append((i,j))
        while q:
            x,y=q.popleft()
            d=ans[x][y]

            if x==n-1 and y==m-1:
                return d

            c=matrix[x][y]
```

matrix =

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：

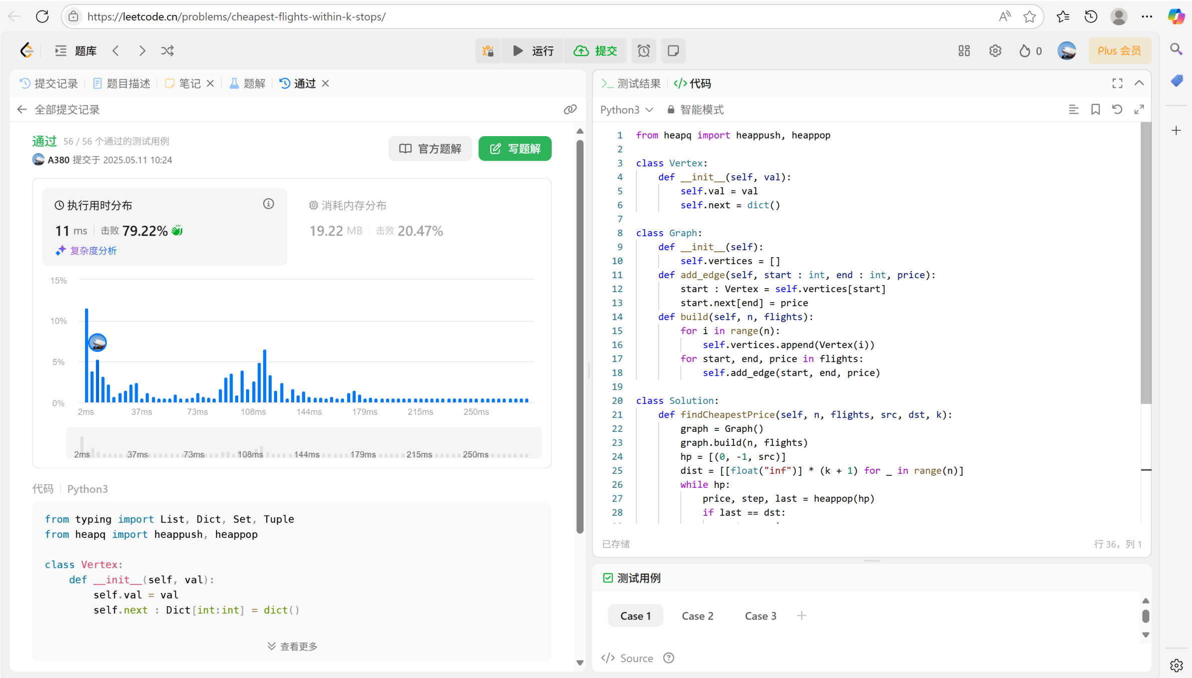
```
from heapq import heappush, heappop

class Vertex:
    def __init__(self, val):
        self.val = val
```

```python
        self.next = dict()

class Graph:
    def __init__(self):
        self.vertices = []
    def add_edge(self, start : int, end : int, price):
        start : Vertex = self.vertices[start]
        start.next[end] = price
    def build(self, n, flights):
        for i in range(n):
            self.vertices.append(Vertex(i))
        for start, end, price in flights:
            self.add_edge(start, end, price)

class Solution:
    def findCheapestPrice(self, n, flights, src, dst, k):
        graph = Graph()
        graph.build(n, flights)
        hp = [(0, -1, src)]
        dist = [[float("inf")] * (k + 1) for _ in range(n)]
        while hp:
            price, step, last = heappop(hp)
            if last == dst:
                return price
            for new, price_between in graph.vertices[last].next.items():
                new_price = price + price_between
                if step < k and dist[new][step + 1] > new_price:
                    heappush(hp, (new_price, step + 1, new))
                    dist[new][step + 1] = new_price
        return -1
```



通过 56 / 56 个通过的测试用例

A380 提交于 2025.05.11 10:24

执行用时分布

11 ms | 击败 79.22%

复杂度分析

消耗内存分布

19.22 MB | 击败 20.47%

代码 | Python3

```python
from typing import List, Dict, Set, Tuple
from heapq import heappush, heappop

class Vertex:
    def __init__(self, val):
        self.val = val
        self.next : Dict[int:int] = dict()
```

查看更多

```python
from heapq import heappush, heappop

class Vertex:
    def __init__(self, val):
        self.val = val
        self.next = dict()

class Graph:
    def __init__(self):
        self.vertices = []
    def add_edge(self, start : int, end : int, price):
        start : Vertex = self.vertices[start]
        start.next[end] = price
    def build(self, n, flights):
        for i in range(n):
            self.vertices.append(Vertex(i))
        for start, end, price in flights:
            self.add_edge(start, end, price)

class Solution:
    def findCheapestPrice(self, n, flights, src, dst, k):
        graph = Graph()
        graph.build(n, flights)
        hp = [(0, -1, src)]
        dist = [[float("inf")] * (k + 1) for _ in range(n)]
        while hp:
            price, step, last = heappop(hp)
            if last == dst:
```

M03424: Candies

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

思路：

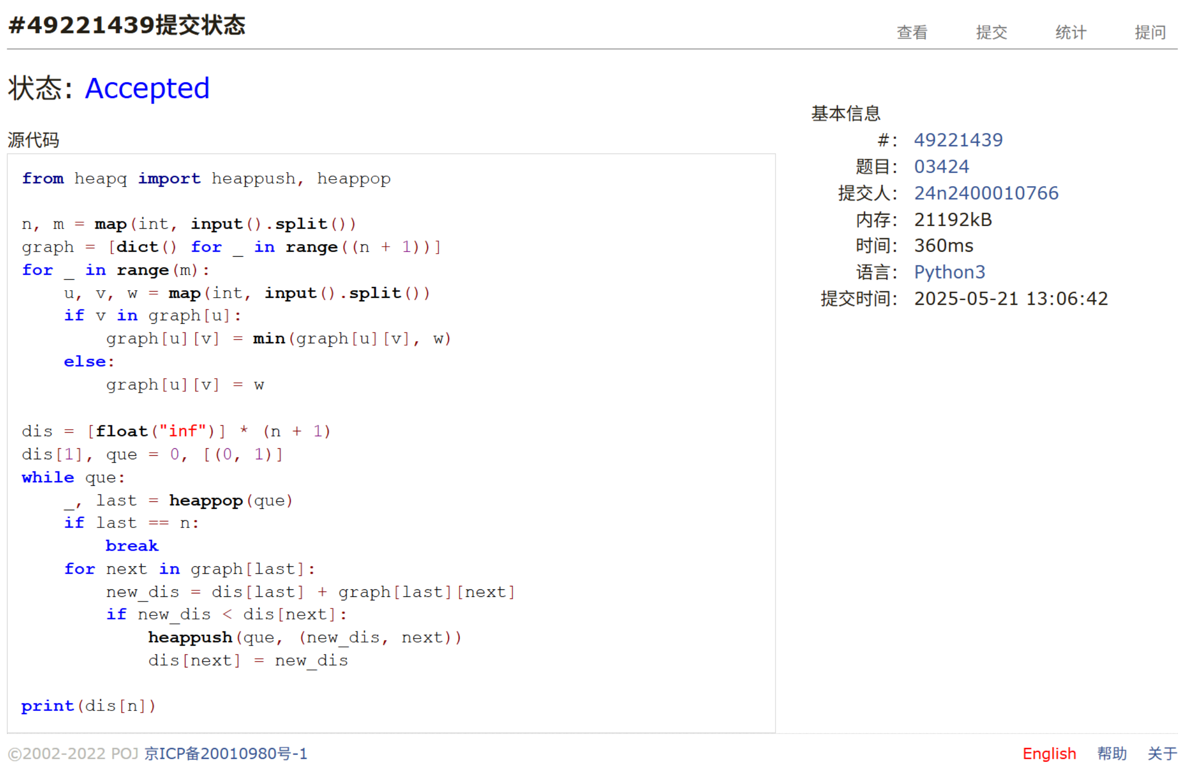
代码：

```python
from heapq import heappush, heappop

n, m = map(int, input().split())
graph = [dict() for _ in range((n + 1))]
for _ in range(m):
    u, v, w = map(int, input().split())
    if v in graph[u]:
        graph[u][v] = min(graph[u][v], w)
    else:
        graph[u][v] = w

dis = [float("inf")] * (n + 1)
dis[1], que = 0, [(0, 1)]
while que:
    _, last = heappop(que)
    if last == n:
        break
    for next in graph[last]:
        new_dis = dis[last] + graph[last][next]
        if new_dis < dis[next]:
            heappush(que, (new_dis, next))
            dis[next] = new_dis

print(dis[n])
```



#49221439提交状态

状态: Accepted

源代码

```python
from heapq import heappush, heappop

n, m = map(int, input().split())
graph = [dict() for _ in range((n + 1))]
for _ in range(m):
    u, v, w = map(int, input().split())
    if v in graph[u]:
        graph[u][v] = min(graph[u][v], w)
    else:
        graph[u][v] = w

dis = [float("inf")] * (n + 1)
dis[1], que = 0, [(0, 1)]
while que:
    _, last = heappop(que)
    if last == n:
        break
    for next in graph[last]:
        new_dis = dis[last] + graph[last][next]
        if new_dis < dis[next]:
            heappush(que, (new_dis, next))
            dis[next] = new_dis

print(dis[n])
```

基本信息

#: 49221439
题目: 03424
提交人: 24n2400010766
内存: 21192kB
时间: 360ms
语言: Python3
提交时间: 2025-05-21 13:06:42

©2002-2022 POJ 京ICP备20010980号-1

M22508:最小奖金方案

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

思路:

代码:

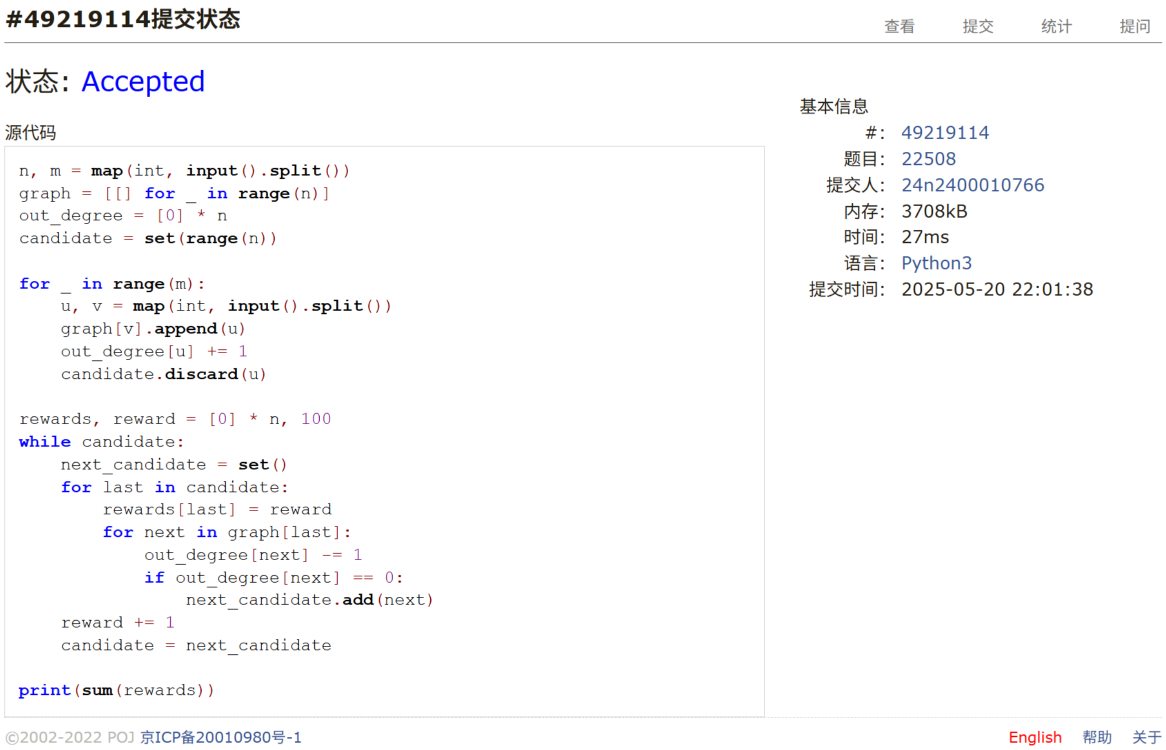
```python
n, m = map(int, input().split())
graph = [[] for _ in range(n)]
out_degree = [0] * n
candidate = set(range(n))

for _ in range(m):
    u, v = map(int, input().split())
    graph[v].append(u)
    out_degree[u] += 1
    candidate.discard(u)

rewards, reward = [0] * n, 100
while candidate:
    next_candidate = set()
    for last in candidate:
        rewards[last] = reward
        for next in graph[last]:
            out_degree[next] -= 1
            if out_degree[next] == 0:
                next_candidate.add(next)
    reward += 1
    candidate = next_candidate
```

```python

print(sum(rewards))
```



#49219114提交状态

状态: Accepted

源代码

```python
n, m = map(int, input().split())
graph = [[] for _ in range(n)]
out_degree = [0] * n
candidate = set(range(n))

for _ in range(m):
    u, v = map(int, input().split())
    graph[v].append(u)
    out_degree[u] += 1
    candidate.discard(u)

rewards, reward = [0] * n, 100
while candidate:
    next_candidate = set()
    for last in candidate:
        rewards[last] = reward
        for next in graph[last]:
            out_degree[next] -= 1
            if out_degree[next] == 0:
                next_candidate.add(next)
    reward += 1
    candidate = next_candidate

print(sum(rewards))
```

基本信息

#:	49219114
题目:	22508
提交人:	24n2400010766
内存:	3708kB
时间:	27ms
语言:	Python3
提交时间:	2025-05-20 22:01:38

©2002-2022 POJ 京ICP备20010980号-1

2. 学习总结和收获

1. 需要特别小心"重复"的问题, 例如输入数据中, 图可能会有重边, 关键词可能会有重复
2. 希望抓紧时间总结最短路径相关算法. (最近DDL太多了)