

Assignment #D: 图 & 散列表

Updated 2042 GMT+8 May 20, 2025

2025 spring, Compiled by 王梓航、物理学院

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。

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：要求太奇怪了，最后只能顺从了。

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
a = [int(i) for i in data[index:index+n]]
d = [0.5]*m
c = []
for index in a:
    temp = index%m
    if d[temp]==0.5 or d[temp]==index:
        d[temp]=index
        c.append(temp)
        continue
    j = 0
    while True:
        j+=1
        x,y = (temp+j**2)%m,(temp-j**2)%m
        if d[x]==0.5 or d[x]==index:
            d[x]=index
            c.append(x)
            break
        if d[y]==0.5 or d[y]==index:
            d[y]=index
            c.append(y)
            break
print(*c)

```

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

```

基本信息

#: 49238336
 题目: 17975
 提交人: 24n2400011481
 内存: 3672kB
 时间: 22ms
 语言: Python3
 提交时间: 2025-05-22 23:48:53

M01258: Agri-Net

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

思路:

代码:

```
import sys
input = sys.stdin.read
data = input().split()
m = len(data)
i = 0
import heapq
while i < m:
    n = int(data[i])
    i += 1
    a = [list(map(int, data[index:index + n])) for index in range(i, i + n * n, n)]
    i += n * n
    ans = 0
    key = [(a[0][j], j) for j in range(1, n)]
    b = set(range(1, n))
    heapq.heapify(key)
    while b:
        temp, index = heapq.heappop(key)
        if index not in b:
            continue
        ans += temp
        b.remove(index)
        for j in b:
            heapq.heappush(key, (a[index][j], j))
    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()
m = len(data)
i = 0
import heapq
while i < m:
```

基本信息

#: 49236349
题目: 01258
提交人: 24n2400011481
内存: 5624kB
时间: 33ms
语言: Python3
提交时间: 2025-05-22 20:16:21

M3552.网络传送门旅游

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

思路: 正常bfs, 因为用过一次之后就都经过了, 且已经保证了最小路径, 所以就不需要额外判断传送过程会不会重复在一个字母上发生

代码:

```
class Solution:
```

```

def minMoves(self, matrix: List[str]) -> int:
    a = {}
    dir = {(-1,0),(1,0),(0,1),(0,-1)}
    for i,index in enumerate(matrix):
        for j,s in enumerate(index):
            if s not in {'.', '#'}:
                a.setdefault(s, set())
                a[s].add((i,j))

    from collections import deque
    b = deque([(0,0,0)])
    n = len(matrix)
    m = len(matrix[0])
    visited = [[False]*m for _ in range(n)]
    visited[0][0]=True
    if matrix[0][0] not in {'.', '#'}:
        for i,j in a[matrix[0][0]]:
            visited[i][j]=True
            b.append((0,i,j))

    count = 0
    while b and not visited[n-1][m-1]:
        count,x,y =b.popleft()
        count+=1
        for dx,dy in dir:
            px,py = x+dx,y+dy
            if 0<=px<n and 0<=py<m:
                if not visited[px][py] and matrix[px][py]!='#':
                    visited[px][py]=True
                    b.append((count,px,py))
                    if matrix[px][py] not in {'.', '#'}:
                        for i,j in a[matrix[px][py]]:
                            if not visited[i][j]:
                                visited[i][j]=True
                                b.append((count,i,j))

    if visited[n-1][m-1]:
        return count
    else:
        return -1

```

代码运行截图 (至少包含有"Accepted")

8

通过

2025.05.19

Python3

🕒 2163 ms

💾 72.4 MB

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：因为最小堆，后面的距离上没有优势，如果次数上也没有优势就优化掉，这样也比较快。

代码：

```
class Solution:
    def findCheapestPrice(self, n: int, flights: List[List[int]], src: int, dst: int, k:
int) -> int:
        b = [[inf]*n for _ in range(n)]
        d = [set() for _ in range(n)]
        for i,j,s in flights:
            b[i][j] = s
            d[i].add(j)
        import heapq
        c = []
        heapq.heappush(c, [0,0,src])
        ans = inf
        time = [inf]*n
        while c:
            price,count,place = heapq.heappop(c)
            if price>=ans or time[place]<=count:
                continue
            time[place]=count
            if place == dst:
                ans = min(ans,price)
            if count==k+1:
                continue
            for index in d[place]:
                temp = price+b[place][index]
                heapq.heappush(c, [temp,count+1,index])
        if ans==inf:
            return -1
        else:
            return ans
```

代码运行截图 (至少包含有"Accepted")

通过

2025.05.13

Python3

🕒 3 ms

💾 19.4 MB

M03424: Candies

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

思路：最短的路径就是所有限制的下限

代码：

```
n,m=map(int,input().split())
a = [[] for i in range(n)]
for _ in range(m):
    i,j,k = map(int,input().split())
    a[i-1].append((j-1,k))
import heapq
b=[(0,0)]
num = [float('inf')]*n
num[0]=0
while b:
    count,i = heapq.heappop(b)
    if num[i]<count:
        continue
    for p,q in a[i]:
        temp = count+q
        if num[p]>temp:
            num[p]=temp
            heapq.heappush(b,(temp,p))
print(num[-1])
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n,m=map(int,input().split())
a = [[] for i in range(n)]
for _ in range(m):
    i,j,k = map(int,input().split())
    a[i-1].append((j-1,k))
import heapq
b=[(0,0)]
```

基本信息

#: 49235808
题目: 03424
提交人: 24n2400011481
内存: 24504kB
时间: 376ms
语言: Python3
提交时间: 2025-05-22 19:44:18

M22508:最小奖金方案

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

思路：利用拓扑排序处理

代码：

```

n,m = map(int,input().split())
win = [0]*n
a = set()
f = [set() for _ in range(n)]
for _ in range(m):
    x,y = map(int,input().split())
    if x not in f[y]:
        win[x]+=1
        a.add(x)
        f[y].add(x)
from collections import deque
b = deque(set(range(n))-a)
price = [0]*n
ans = 0
while b:
    index = b.popleft()
    temp = price[index]
    ans+=temp
    for i in f[index]:
        win[i]-=1
        price[i]=temp+1
        if win[i]==0:
            b.append(i)
print(ans+100*n)

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

n,m = map(int,input().split())
win = [0]*n
a = set()
f = [set() for _ in range(n)]
for _ in range(m):
    x,y = map(int,input().split())
    if x not in f[y]:
        win[x]+=1
        a.add(x)
        f[y].add(x)

```

基本信息

#: 49158955
 题目: 22508
 提交人: 24n2400011481
 内存: 3888kB
 时间: 26ms
 语言: Python3
 提交时间: 2025-05-14 14:36:14

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。

现在在看讲义后面的算法和题目, 感觉还是很多, 这周得多做题了。