Assignment #B: 图论和树算

Updated 0221 GMT+8 May 6, 2024

2024 spring, Complied by 钟明衡 物理学院

说明：

1）请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora https://typoraio.cn ，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

2）提交时候先提交pdf文件，再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、"作业评论"区有上传的md或者doc附件。

3）如果不能在截止前提交作业，请写明原因。

编程环境

操作系统：Windows_NT x64 10.0.19045

Python编程环境：Visual Studio Code 1.76.1

C/C++编程环境：Visual Studio Code 1.76.1

1. 题目

28170: 算鹰

dfs, http://cs101.openjudge.cn/practice/28170/

思路：

用dfs标记一整块，计算整块的数量

代码

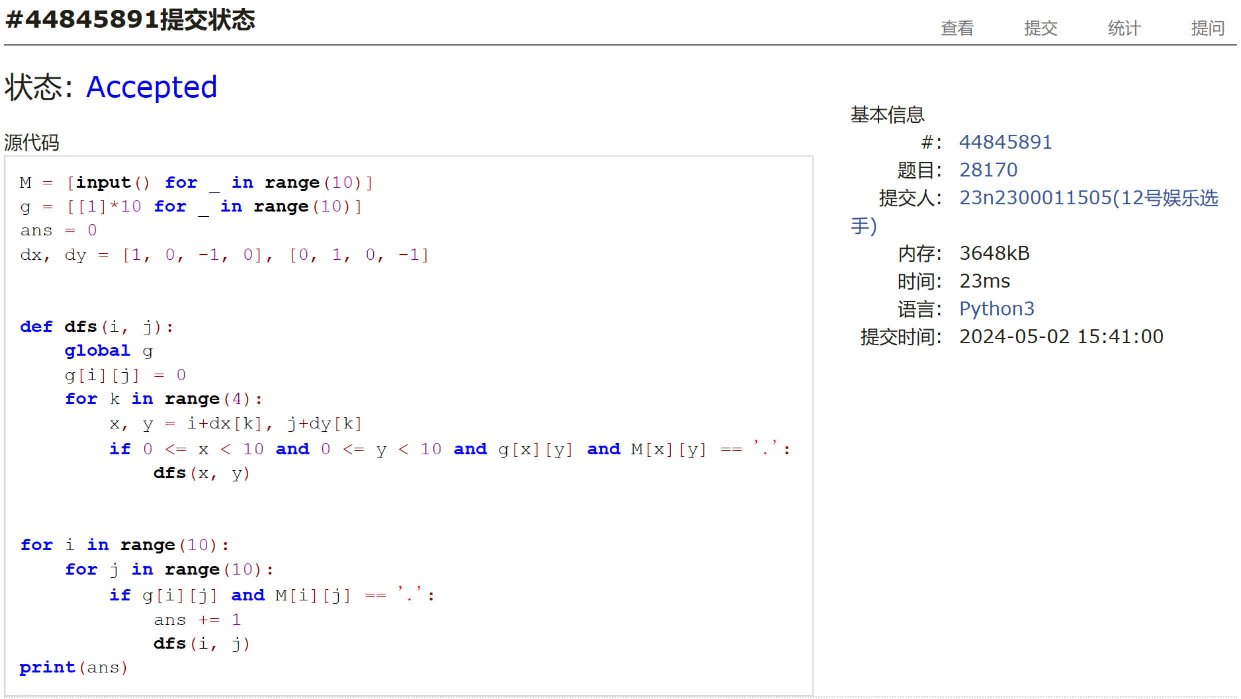
```python
M = [input() for _ in range(10)]
g = [[1]*10 for _ in range(10)]
ans = 0
dx, dy = [1, 0, -1, 0], [0, 1, 0, -1]


def dfs(i, j):
    global g
    g[i][j] = 0
    for k in range(4):
        x, y = i+dx[k], j+dy[k]
        if 0 <= x < 10 and 0 <= y < 10 and g[x][y] and M[x][y] == '.':
            dfs(x, y)


for i in range(10):
    for j in range(10):
        if g[i][j] and M[i][j] == '.':
            ans += 1
            dfs(i, j)
print(ans)

```

代码运行截图



#44845891提交状态

状态: Accepted

源代码

```
M = [input() for _ in range(10)]
g = [[1]*10 for _ in range(10)]
ans = 0
dx, dy = [1, 0, -1, 0], [0, 1, 0, -1]


def dfs(i, j):
    global g
    g[i][j] = 0
    for k in range(4):
        x, y = i+dx[k], j+dy[k]
        if 0 <= x < 10 and 0 <= y < 10 and g[x][y] and M[x][y] == '.':
            dfs(x, y)


for i in range(10):
    for j in range(10):
        if g[i][j] and M[i][j] == '.':
            ans += 1
            dfs(i, j)
print(ans)
```

基本信息

#: 44845891
题目: 28170
提交人: 23n2300011505(12号娱乐选手)
内存: 3648kB
时间: 23ms
语言: Python3
提交时间: 2024-05-02 15:41:00

02754: 八皇后

dfs, http://cs101.openjudge.cn/practice/02754/

思路:

直接dfs暴力搜索，用$i+j$和$i-j$来标记已经占用的斜列

代码

```
count = 0
ans = []

```

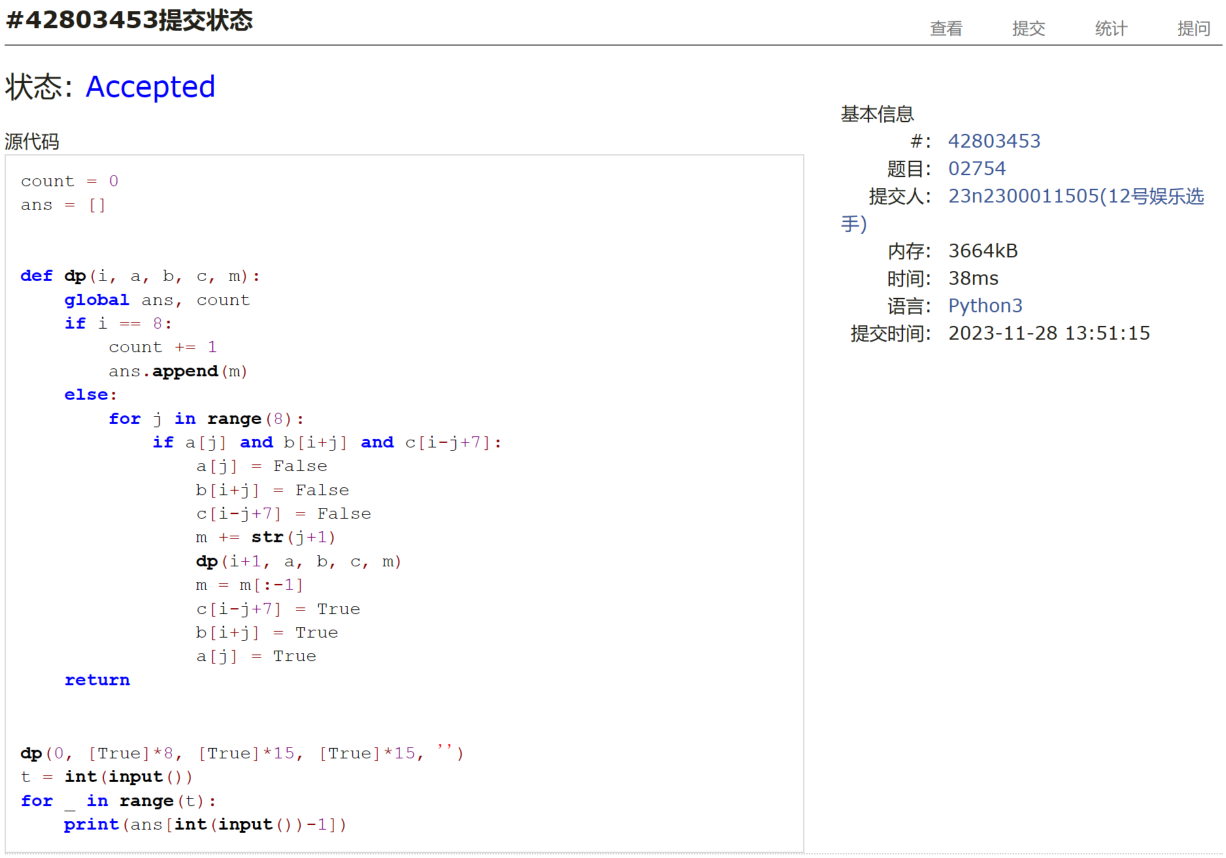
```python

def dp(i, a, b, c, m):
    global ans, count
    if i == 8:
        count += 1
        ans.append(m)
    else:
        for j in range(8):
            if a[j] and b[i+j] and c[i-j+7]:
                a[j] = False
                b[i+j] = False
                c[i-j+7] = False
                m += str(j+1)
                dp(i+1, a, b, c, m)
                m = m[:-1]
                c[i-j+7] = True
                b[i+j] = True
                a[j] = True
    return


dp(0, [True]*8, [True]*15, [True]*15, '')
t = int(input())
for _ in range(t):
    print(ans[int(input())-1])

```

代码运行截图


#42803453提交状态

状态: Accepted

源代码

```
count = 0
ans = []


def dp(i, a, b, c, m):
    global ans, count
    if i == 8:
        count += 1
        ans.append(m)
    else:
        for j in range(8):
            if a[j] and b[i+j] and c[i-j+7]:
                a[j] = False
                b[i+j] = False
                c[i-j+7] = False
                m += str(j+1)
                dp(i+1, a, b, c, m)
                m = m[:-1]
                c[i-j+7] = True
                b[i+j] = True
                a[j] = True
    return


dp(0, [True]*8, [True]*15, [True]*15, '')
t = int(input())
for _ in range(t):
    print(ans[int(input())-1])
```

基本信息

#: 42803453
题目: 02754
提交人: 23n2300011505(12号娱乐选手)
内存: 3664kB
时间: 38ms
语言: Python3
提交时间: 2023-11-28 13:51:15

03151: Pots

bfs, http://cs101.openjudge.cn/practice/03151/

思路：

很直接的bfs，不断尝试各种可能即可。注意要记录路径，我采用的方法是用1~6代表各种操作，路径则为一个整数，比如样例中的那个操作就可以表示为263626

代码

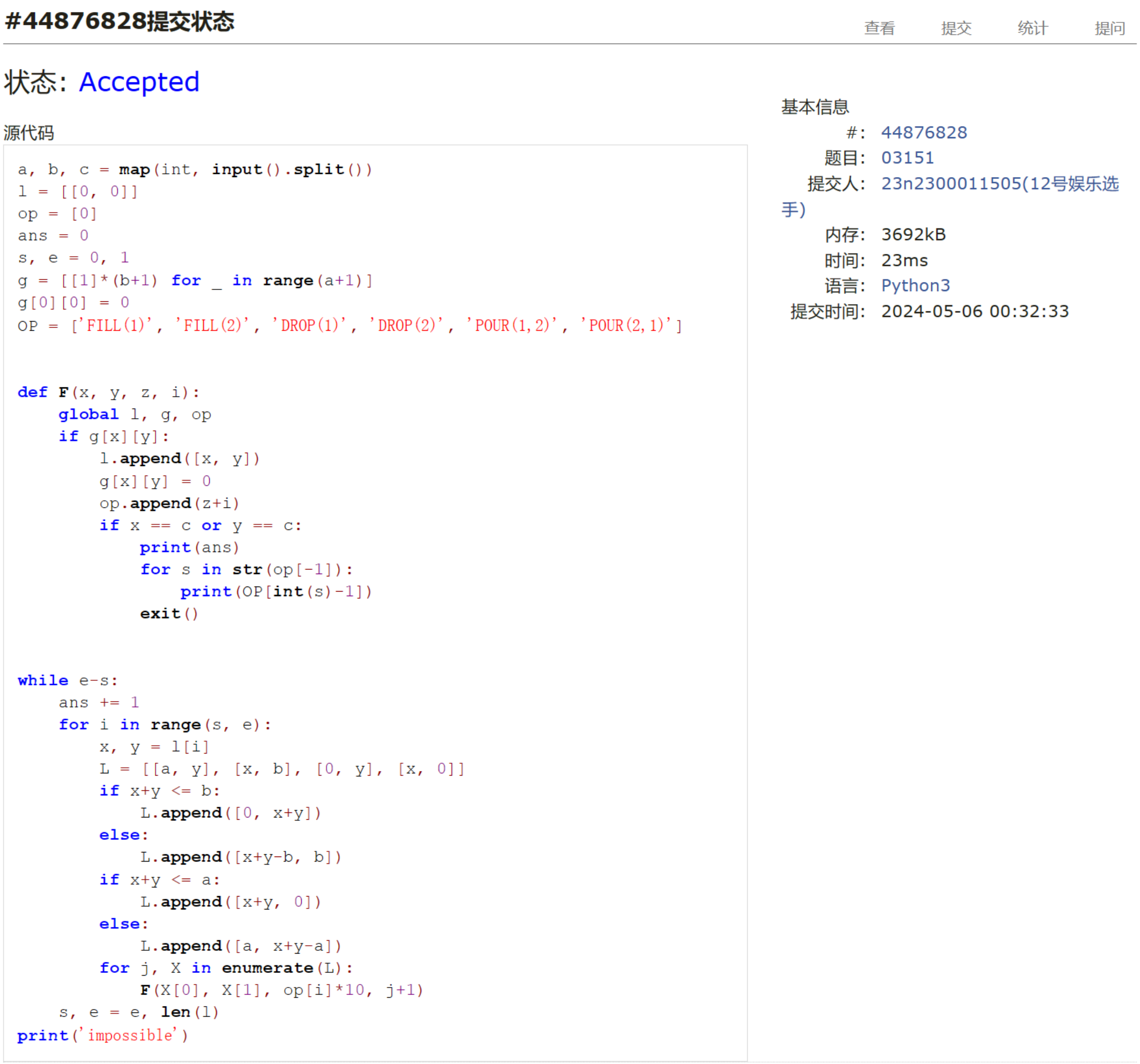
```python
a, b, c = map(int, input().split())
l = [[0, 0]]
op = [0]
ans = 0
s, e = 0, 1
g = [[1]*(b+1) for _ in range(a+1)]
g[0][0] = 0
OP = ['FILL(1)', 'FILL(2)', 'DROP(1)', 'DROP(2)', 'POUR(1,2)', 'POUR(2,1)']


def F(x, y, z, i):
    global l, g, op
    if g[x][y]:
        l.append([x, y])
        g[x][y] = 0
        op.append(z+i)
        if x == c or y == c:
            print(ans)
            for s in str(op[-1]):
                print(OP[int(s)-1])
            exit()


while e-s:
    ans += 1
    for i in range(s, e):
        x, y = l[i]
        L = [[a, y], [x, b], [0, y], [x, 0]]
        if x+y <= b:
            L.append([0, x+y])
        else:
            L.append([x+y-b, b])
        if x+y <= a:
            L.append([x+y, 0])
        else:
            L.append([a, x+y-a])
        for j, X in enumerate(L):
            F(X[0], X[1], op[i]*10, j+1)
    s, e = e, len(l)
print('impossible')

```

代码运行截图



状态: Accepted

源代码

```
a, b, c = map(int, input().split())
l = [[0, 0]]
op = [0]
ans = 0
s, e = 0, 1
g = [[1]*(b+1) for _ in range(a+1)]
g[0][0] = 0
OP = ['FILL(1)', 'FILL(2)', 'DROP(1)', 'DROP(2)', 'POUR(1,2)', 'POUR(2,1)']


def F(x, y, z, i):
    global l, g, op
    if g[x][y]:
        l.append([x, y])
        g[x][y] = 0
        op.append(z+i)
        if x == c or y == c:
            print(ans)
            for s in str(op[-1]):
                print(OP[int(s)-1])
            exit()


while e-s:
    ans += 1
    for i in range(s, e):
        x, y = l[i]
        L = [[a, y], [x, b], [0, y], [x, 0]]
        if x+y <= b:
            L.append([0, x+y])
        else:
            L.append([x+y-b, b])
        if x+y <= a:
            L.append([x+y, 0])
        else:
            L.append([a, x+y-a])
        for j, X in enumerate(L):
            F(X[0], X[1], op[i]*10, j+1)
    s, e = e, len(l)
print('impossible')
```

基本信息

#: 44876828
题目: 03151
提交人: 23n2300011505(12号娱乐选手)
内存: 3692kB
时间: 23ms
语言: Python3
提交时间: 2024-05-06 00:32:33

05907: 二叉树的操作

http://cs101.openjudge.cn/practice/05907/

思路:

直接按字面意思写就可以，建树时记得记录parent，方便后序查找交换

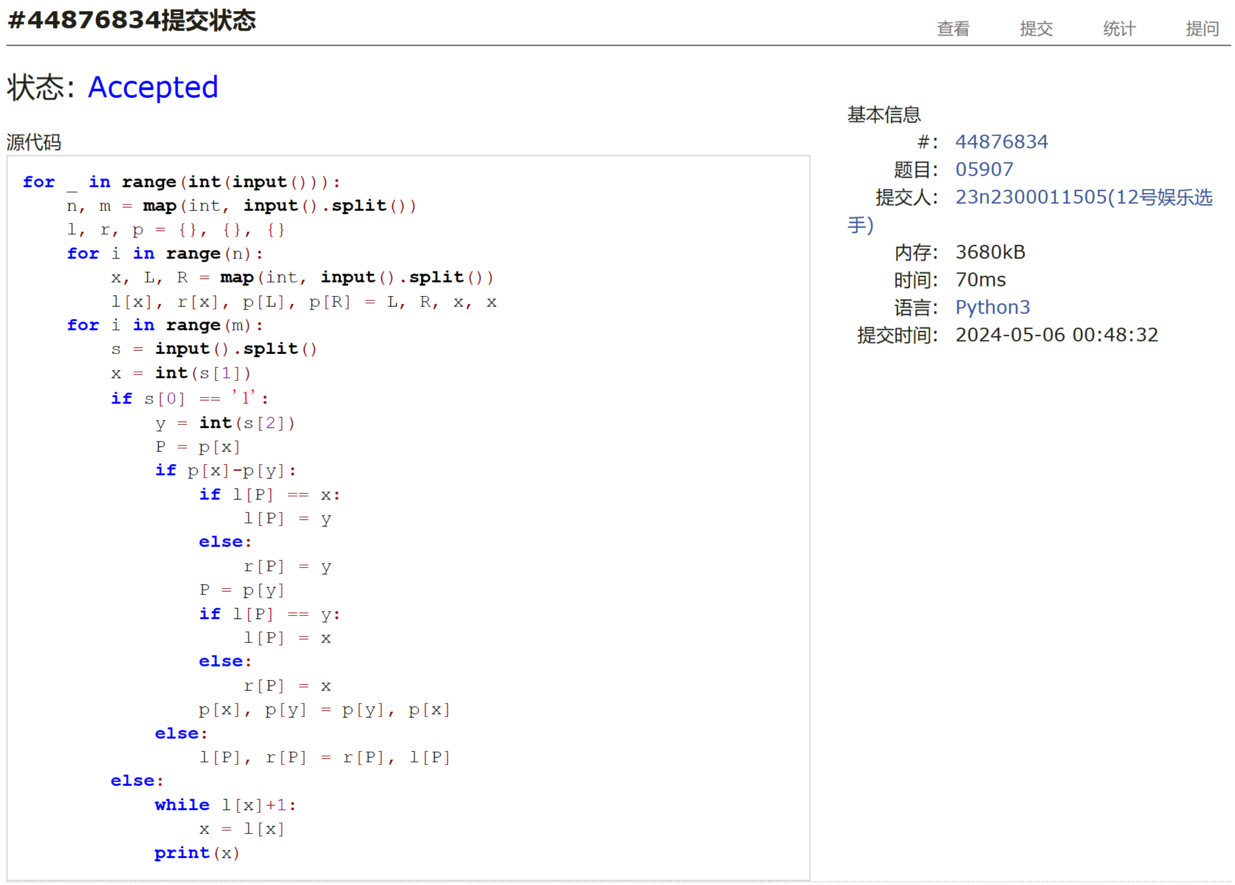
代码

```
for _ in range(int(input())):
    n, m = map(int, input().split())
    l, r, p = {}, {}, {}
    for i in range(n):
        x, L, R = map(int, input().split())
        l[x], r[x], p[L], p[R] = L, R, x, x
    for i in range(m):
        s = input().split()
```

```python
        x = int(s[1])
        if s[0] == '1':
            y = int(s[2])
            P = p[x]
            if p[x]-p[y]:
                if l[P] == x:
                    l[P] = y
                else:
                    r[P] = y
                P = p[y]
                if l[P] == y:
                    l[P] = x
                else:
                    r[P] = x
                p[x], p[y] = p[y], p[x]
            else:
                l[P], r[P] = r[P], l[P]
        else:
            while l[x]+1:
                x = l[x]
            print(x)

```

代码运行截图



#44876834提交状态

状态: Accepted

源代码

```python
for _ in range(int(input())):
    n, m = map(int, input().split())
    l, r, p = {}, {}, {}
    for i in range(n):
        x, L, R = map(int, input().split())
        l[x], r[x], p[L], p[R] = L, R, x, x
    for i in range(m):
        s = input().split()
        x = int(s[1])
        if s[0] == ’1’:
            y = int(s[2])
            P = p[x]
            if p[x]-p[y]:
                if l[P] == x:
                    l[P] = y
                else:
                    r[P] = y
                P = p[y]
                if l[P] == y:
                    l[P] = x
                else:
                    r[P] = x
                p[x], p[y] = p[y], p[x]
            else:
                l[P], r[P] = r[P], l[P]
        else:
            while l[x]+1:
                x = l[x]
            print(x)
```

基本信息

#:	44876834
题目:	05907
提交人:	23n2300011505(12号娱乐选手)
内存:	3680kB
时间:	70ms
语言:	Python3
提交时间:	2024-05-06 00:48:32

18250: 冰阔落 I

Disjoint set, http://cs101.openjudge.cn/practice/18250/

思路：

非常标准的并查集，“祖宗”为当前可乐处于的杯子

代码

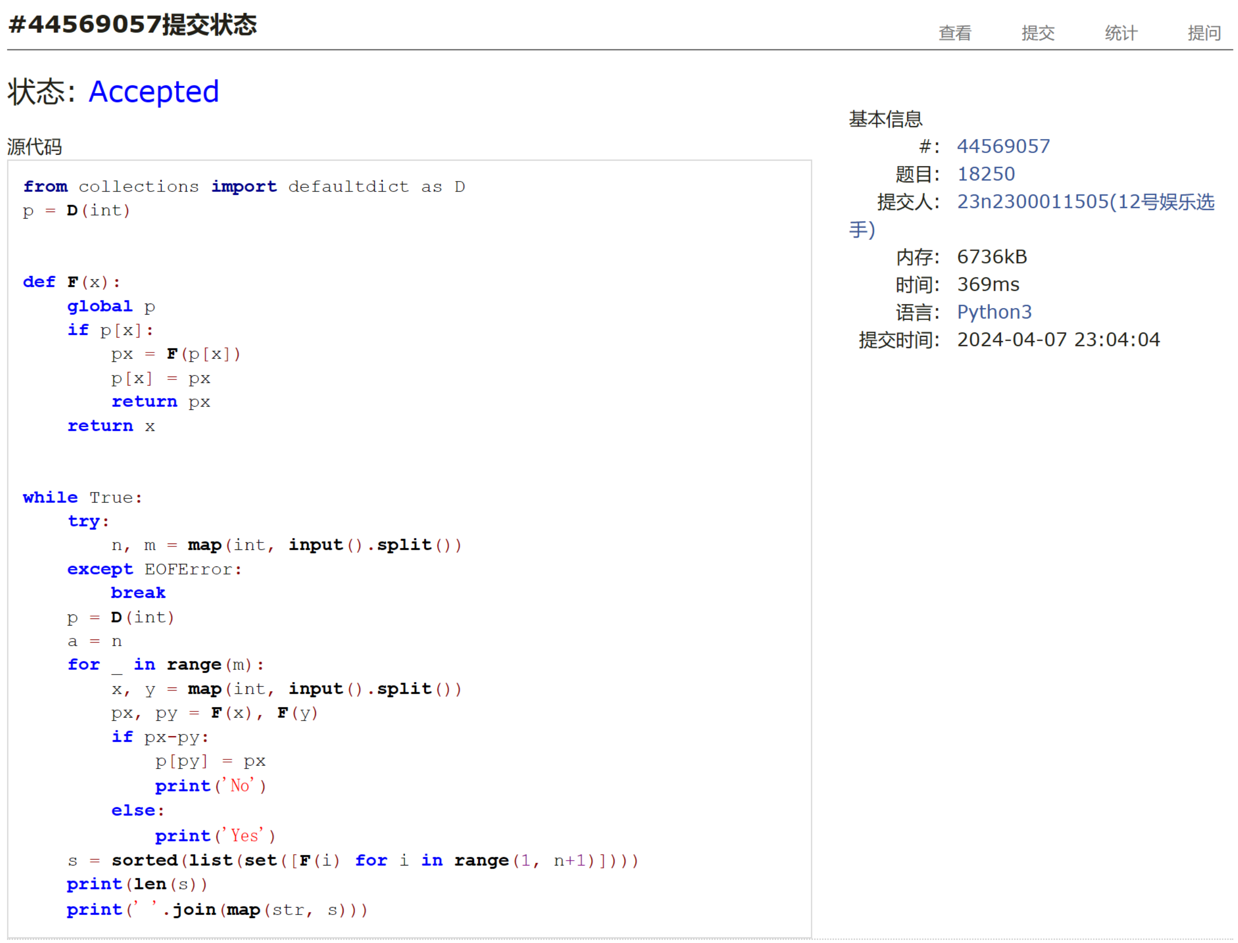
```python
from collections import defaultdict as D
p = D(int)


def F(x):
    global p
    if p[x]:
        px = F(p[x])
        p[x] = px
        return px
    return x


while True:
    try:
        n, m = map(int, input().split())
    except EOFError:
        break
    p = D(int)
    a = n
    for _ in range(m):
        x, y = map(int, input().split())
        px, py = F(x), F(y)
        if px-py:
            p[py] = px
            print('No')
        else:
            print('Yes')
    s = sorted(list(set([F(i) for i in range(1, n+1)])))
    print(len(s))
    print(' '.join(map(str, s)))

```

代码运行截图



#44569057提交状态

状态: Accepted

源代码

```python
from collections import defaultdict as D
p = D(int)


def F(x):
    global p
    if p[x]:
        px = F(p[x])
        p[x] = px
        return px
    return x


while True:
    try:
        n, m = map(int, input().split())
    except EOFError:
        break
    p = D(int)
    a = n
    for _ in range(m):
        x, y = map(int, input().split())
        px, py = F(x), F(y)
        if px-py:
            p[py] = px
            print('No')
        else:
            print('Yes')
    s = sorted(list(set([F(i) for i in range(1, n+1)])))
    print(len(s))
    print(' '.join(map(str, s)))
```

基本信息

#: 44569057
题目: 18250
提交人: 23n2300011505(12号娱乐选手)
内存: 6736kB
时间: 369ms
语言: Python3
提交时间: 2024-04-07 23:04:04

05443: 兔子与樱花

http://cs101.openjudge.cn/practice/05443/

思路：

非常标准的Dijkstra，为了输出路径，要保存走到每个节点的上一个节点

代码

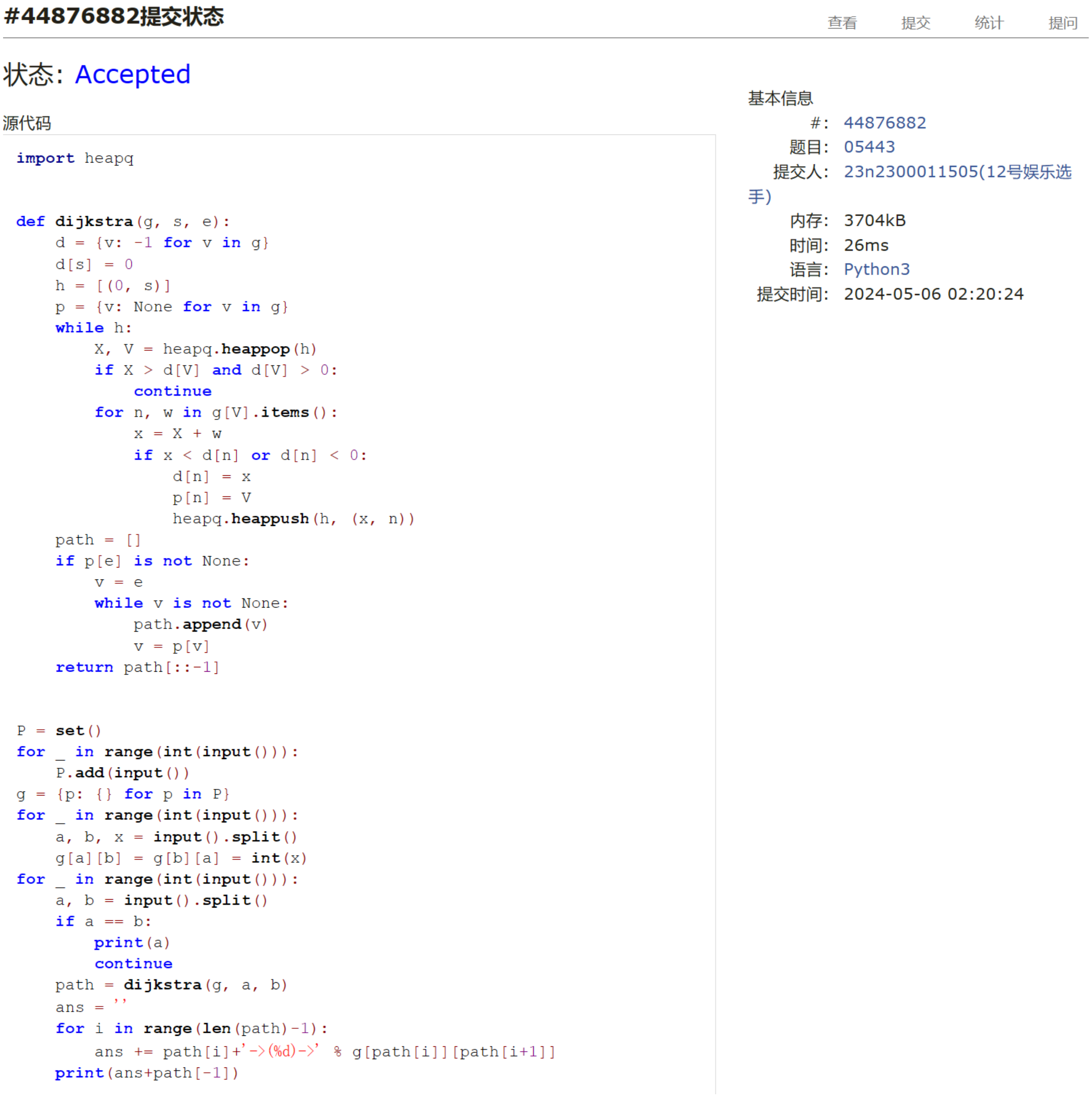
```python
import heapq


def dijkstra(g, s, e):
    d = {v: -1 for v in g}
    d[s] = 0
    h = [(0, s)]
    p = {v: None for v in g}
    while h:
        x, v = heapq.heappop(h)
        if x > d[v] and d[v] > 0:
            continue
        for n, w in g[v].items():
            x = X + w
            if x < d[n] or d[n] < 0:
                d[n] = x
                p[n] = V
                heapq.heappush(h, (x, n))
    path = []
    if p[e] is not None:
        v = e
        while v is not None:
            path.append(v)
            v = p[v]
    return path[::-1]


P = set()
for _ in range(int(input())):
    P.add(input())
g = {p: {} for p in P}
for _ in range(int(input())):
    a, b, x = input().split()
    g[a][b] = g[b][a] = int(x)
for _ in range(int(input())):
    a, b = input().split()
    if a == b:
        print(a)
        continue
    path = dijkstra(g, a, b)
    ans = ''
    for i in range(len(path)-1):
        ans += path[i]+'->(%d)->' % g[path[i]][path[i+1]]
    print(ans+path[-1])

```

代码运行截图



#44876882提交状态

状态: Accepted

源代码

```
import heapq


def dijkstra(g, s, e):
    d = {v: -1 for v in g}
    d[s] = 0
    h = [(0, s)]
    p = {v: None for v in g}
    while h:
        X, V = heapq.heappop(h)
        if X > d[V] and d[V] > 0:
            continue
        for n, w in g[V].items():
            x = X + w
            if x < d[n] or d[n] < 0:
                d[n] = x
                p[n] = V
                heapq.heappush(h, (x, n))
    path = []
    if p[e] is not None:
        v = e
        while v is not None:
            path.append(v)
            v = p[v]
    return path[::-1]


P = set()
for _ in range(int(input())):
    P.add(input())
g = {p: {} for p in P}
for _ in range(int(input())):
    a, b, x = input().split()
    g[a][b] = g[b][a] = int(x)
for _ in range(int(input())):
    a, b = input().split()
    if a == b:
        print(a)
        continue
    path = dijkstra(g, a, b)
    ans = ''
    for i in range(len(path)-1):
        ans += path[i]+'->(%d)->' % g[path[i]][path[i+1]]
    print(ans+path[-1])
```

基本信息

#:	44876882
题目:	05443
提交人:	23n2300011505(12号娱乐选手)
内存:	3704kB
时间:	26ms
语言:	Python3
提交时间:	2024-05-06 02:20:24

2. 学习总结和收获

五一出去玩了，几天没写代码，该捡回来了