

处理每个元素有几种状态 / 状态间有逻辑关系的集合问题

⇒ 每个元素拆成多个子节点，每种状态用一个编号，用并查集维护

例  $A \rightarrow B \rightarrow C$  食物链 给  $n$  只动物， $k$  条发言 ( $A_{k1}$  吃  $A_{k2}$  /  $A_{k3}$  与  $A_{k4}$  同类)  
判断几条假话

我们设  $[0, n)$  区间表示同类， $[n, 2*n)$  区间表示  $x$  吃的动物， $[2*n, 3*n)$  表示吃  $x$  的动物。

如果是关系1：

将  $y$  和  $x$  合并。将  $y$  吃的与  $x$  吃的合并。将吃  $y$  的和吃  $x$  的合并。

如果是关系2：

将  $y$  和  $x$  吃的合并。将吃  $y$  的与  $x$  合并。将  $y$  吃的与吃  $x$  的合并。

```
n, k = map(int, input().split())

p = [0] * (3 * n + 1)
for i in range(3 * n + 1): # 并查集初始化
    p[i] = i

ans = 0
for _ in range(k):
    a, x, y = map(int, input().split())
    if x > n or y > n:
        ans += 1; continue

    if a == 1:
        if find(x + n) == find(y) or find(y + n) == find(x):
            ans += 1; continue
```

No. 124 / 160

```
# 合并
p[find(x)] = find(y)
p[find(x + n)] = find(y + n)
p[find(x + 2 * n)] = find(y + 2 * n)
else:
    if find(x) == find(y) or find(y + n) == find(x):
        ans += 1; continue
    p[find(x + n)] = find(y)
    p[find(y + 2 * n)] = find(x)
    p[find(x + 2 * n)] = find(y + n)
```

```
print(ans)
```

