

# Almost Union-Find

---

## 题面翻译

---

有  $n$  个集合， $m$  次操作。规定第  $i$  个集合里初始只有  $i$ 。有三种操作：

1. 输入两个元素  $p$  和  $q$ ，若  $p$  和  $q$  不在一个集合中，合并两个元素的集合。
2. 输入两个元素  $p$  和  $q$ ，若  $p$  和  $q$  不在一个集合中，把  $p$  添加到  $q$  所在的集合。
3. 输入一个元素  $p$ ，查询  $p$  所在集合的元素个数和所有元素之和。

### 【输入格式】

有几组数据。

每组数据第一行输入  $n$  和  $m$  两个整数。

每组数据以下  $m$  行，每行第一个数  $k$  代表选择哪一个命令，若  $k$  是 1 或 2 命令，则再输入两个整数  $p$  和  $q$ 。若  $k$  是 3，则输入一个整数  $p$ 。

输入文件结束符（EOF）结束输入。

### 【输出格式】

输出行数为每组数据 3 号命令的总数。

每一行输出两个整数  $a$  和  $b$ ，即元素个数和元素和。

### 【数据范围】

$$1 \leq n, m \leq 10^5, 1 \leq p, q \leq n。$$

## 题目描述

---

[PDF](#)

I hope you know the beautiful Union-Find structure. In this problem, you're to implement something similar, but not identical.

The data structure you need to write is also a collection of disjoint sets, supporting 3 operations:

1 $p$ $q$	Union the sets containing $p$ and $q$ . If $p$ and $q$ are already in the same set, ignore this command.
2 $p$ $q$	Move $p$ to the set containing $q$ . If $p$ and $q$ are already in the same set, ignore this command.
3 $p$	Return the number of elements and the sum of elements in the set containing $p$ .

Initially, the collection contains  $n$  sets:  $\{1\}, \{2\}, \{3\}, \dots, \{n\}$ .

## 输入格式

There are several test cases. Each test case begins with a line containing two integers  $n$  and  $m$  ( $1 \leq n, m \leq 100,000$ ), the number of integers, and the number of commands. Each of the next  $m$  lines contains a command. For every operation,  $1 \leq p, q \leq n$ . The input is terminated by end-of-file (EOF).

## 输出格式

For each type-3 command, output 2 integers: the number of elements and the sum of elements.

### Explanation

Initially:  $\{1\}, \{2\}, \{3\}, \{4\}, \{5\}$

Collection after operation 1 1 2:  $\{1,2\}, \{3\}, \{4\}, \{5\}$

Collection after operation 2 3 4:  $\{1,2\}, \{3,4\}, \{5\}$  (we omit the empty set that is produced when taking out 3 from  $\{3\}$ )

Collection after operation 1 3 5:  $\{1,2\}, \{3,4,5\}$

Collection after operation 2 4 1:  $\{1,2,4\}, \{3,5\}$

## 样例 #1

### 样例输入 #1

```
5 7
1 1 2
2 3 4
1 3 5
3 4
2 4 1
3 4
3 3
```

## 样例输出 #1

3 12

3 7

2 8