

## 4 Системы непересекающихся множеств

### 4.1 Объединение таблиц

Ваша цель в данной задаче — реализовать симуляцию объединения таблиц в базе данных.

В базе данных есть  $n$  таблиц, пронумерованных от 1 до  $n$ , над одним и тем же множеством столбцов (атрибутов). Каждая таблица содержит либо реальные записи в таблице, либо **символьную ссылку** на другую таблицу. Изначально все таблицы содержат реальные записи, и  $i$ -я таблица содержит  $r_i$  записей. Ваша цель — обработать  $m$  запросов типа  $(destination_i, source_i)$ :

1. Рассмотрим таблицу с номером  $destination_i$ . Пройдясь по цепочке символьных ссылок, найдём номер реальной таблицы, на которую ссылается эта таблица:

пока таблица  $destination_i$  содержит символическую ссылку:  
 $destination_i \leftarrow \text{symlink}(destination_i)$

2. Сделаем то же самое с таблицей  $source_i$ .
3. Теперь таблицы  $destination_i$  и  $source_i$  содержат реальные записи. Если  $destination_i \neq source_i$ , скопируем все записи из таблицы  $source_i$  в таблицу  $destination_i$ , очистим таблицу  $source_i$  и пропишем в неё символическую ссылку на таблицу  $destination_i$ .
4. Выведем максимальный размер среди всех  $n$  таблиц. Размером таблицы называется число строк в ней. Если таблица содержит символическую ссылку, считаем её размер равным нулю.

**Формат входа.** Первая строка содержит числа  $n$  и  $m$  — число таблиц и число запросов, соответственно. Вторая строка содержит  $n$  целых чисел  $r_1, \dots, r_n$  — размеры таблиц. Каждая из последующих  $m$  строк содержит два номера таблиц  $destination_i$  и  $source_i$ , которые необходимо объединить.

**Формат выхода.** Для каждого из  $m$  запросов выведите максимальный размер таблицы после соответствующего объединения.

**Ограничения.**  $1 \leq n, m \leq 100\,000$ ;  $0 \leq r_i \leq 10\,000$ ;  
 $1 \leq destination_i, source_i \leq n$ .

**Пример.**

Вход:

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

Выход:

```
2
2
3
5
5
```

Изначально каждая таблица содержит ровно одну строку.

1. После первой операции объединения все записи из таблицы 5 копируются в таблицу 3. Теперь таблица 5 является ссылкой на таблицу 3, а таблица 3 содержит две записи.
2. Вторая операция аналогичным образом переносит все записи из таблицы 2 в таблицу 4.
3. Третья операция пытается объединить таблицы 1 и 4, но таблица 4 ссылается на таблицу 2, поэтому все записи из таблицы 2 копируются в таблицу 1. Таблица 1 теперь содержит три строки.
4. Чтобы произвести четвёртую операцию, проследим пути из ссылок:  $4 \rightarrow 2 \rightarrow 1$  и  $5 \rightarrow 3$ . Скопируем все записи из таблицы 1 в таблицу 3, после чего в таблице 3 будет пять записей.
5. После этого все таблицы ссылаются на таблицу 3, поэтому все оставшиеся запросы объединения ничего не меняют.

**Пример.**

Вход:

```
6 4
10 0 5 0 3 3
6 6
6 5
5 4
4 3
```

Выход:

```
10
10
10
11
```

1. Запрос объединения таблицы 6 с собой ничего не меняет, максимальным размером по-прежнему остаётся 10 (таблица 1).
2. Записи из таблицы 5 копируются в таблицу 6, размер таблицы 6 становится равным 6.
3. Записи из таблицы 4 копируются в таблицу 6, размер таблицы 6 становится равным 10.
4. Записи из таблицы 3 копируются в таблицу 6, размер таблицы 6 становится равным 11.