7_無趣的資料結構(Boring Data Structure)

(3 分/ 6 分/ 11 分)

時間限制: 1 second
記憶體限制: 256 MB

題目敘述

樂奈最近迷上了函數，特別是一個名為 mex（**m**inimum **ex**cluded value）的函數。mex 函數的定義如下：

- 對於一個集合 S，$\operatorname{mex}(S)$ 是最小的非負整數 x 滿足 S 中不存在 x。
    - 舉例來說：$\operatorname{mex}(\{0,1,3,5\})=2$、$\operatorname{mex}(\{1,3\})=0$、$\operatorname{mex}(\{0,1,2\})=3$。

樂奈在練新歌之餘，用這個函數構造了一道資料結構問題。給定一個數字 n 和 n 個集合 $A_1,A_2,\ldots,A_n$，其中第 i 個集合包含了 $0,1,2,\ldots,n-1$ 之中**除了** $e_{i,1},e_{i,2},\ldots,e_{i,k_i}$ **之外的所有整數**。樂奈想設計一個能支援 q 次以下兩種操作的資料結構：

1. `mex-add`：令 $A=_{\text{def}} A_1\cap A_2\cap\cdots\cap A_n$ 是這 n 個集合的交集且 $x=_{\text{def}}\operatorname{mex}(A)$。對 $i=1,2,\ldots,n$，如果 x 不在 A_i 之中，則將 x 加入 A_i。
2. `toggle` x：對 $i=1,2,\ldots,n$，如果 x 在 A_i 之中，則將其從 A_i 移除；反之，則將 x 加入 A_i。

請在每次操作完之後輸出所有集合內的數字之和。

樂奈在構造完這個問題之後，認為這個問題實在太無趣就跑去吃抹茶巴菲了。喵喵在練吉他的時候無意間看到了留在練習室的題目，於是就順手把它丟給了你。你能解開這個問題嗎？~~還是你要再把問題丟給你的隊友來做呢？~~

P.S. 交集的定義：x 存在於 $S_1\cap S_2\cap\cdots\cap S_k$ 之中若且唯若 x 同時存在於所有 $S_1,S_2,\ldots,S_k$ 之中。

輸入格式

- line 1: $n\ \ q$
- line $1+i$ $(1\le i\le n)$: $k_i\ \ e_{i,1}\ \ e_{i,2}\ \ \cdots\ \ e_{i,k_i}$
- line $n+1+j$ $(1\le j\le q)$: Query_j

其中，Query_j 的格式為下列兩者之一：

- `mex-add`
- `toggle` x_j

以上的變數所代表的含意都如同題目敘述所述。

輸出格式

- line 1: ans

資料範圍

- $1 \le n \le 300\,000$。
- $1 \le q \le 300\,000$。
- $0 \le k_i \le n$（$1 \le i \le n$）。
  - $\sum_{i=1}^{n} k_i \le 1\,000\,000$。
- $0 \le e_{i,1} < e_{i,2} < \cdots < e_{i,k_i} \le n-1$（$1 \le i \le n$）。
- $op_j \in \{\texttt{mex-add}, \texttt{toggle}\}$（$1 \le j \le q$）。
- $0 \le x_j \le n-1$（$1 \le j \le q$ 且 $op_j = \texttt{toggle}$）。
- 輸入的數字都是整數。

子任務

1. (3 points) $op_j = \texttt{toggle}$（$1 \le j \le q$）。
2. (6 points) $k_i = n$（$1 \le i \le n$）。
3. (11 points) 無額外限制。

測試範例

輸入範例 1

```
5 8
1 3
1 2
4 0 1 2 3
1 2
3 0 1 4
toggle 4
mex-add
mex-add
toggle 4
toggle 2
mex-add
toggle 3
mex-add
```

該範例輸入符合子任務 3 的限制。

輸出範例 1

```
20
20
22
34
36
40
37
46
```

- 最一開始：
  - $A_1 = \{0, 1, 2, 4\}$
  - $A_2 = \{0, 1, 3, 4\}$
  - $A_3 = \{4\}$
  - $A_4 = \{0, 1, 3, 4\}$
  - $A_5 = \{2, 3\}$
- 第一次操作 $\text{Query}_1 = \texttt{toggle}\ \ 4$ 之後：
  - $A_1 = \{0, 1, 2\}$（移除 4）
  - $A_2 = \{0, 1, 3\}$（移除 4）
  - $A_3 = \{\}$（移除 4）
  - $A_4 = \{0, 1, 3\}$（移除 4）
  - $A_5 = \{2, 3, 4\}$（加入 4）
  - 數字之和為 $(0 + 1 + 2) + (0 + 1 + 3) + (0 + 1 + 3) + (2 + 3 + 4) = 20$。
- 第二次操作 $\text{Query}_2 = \texttt{mex-add}$ 之後：
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0\}$（加入 0）
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 2, 3, 4\}$（加入 0）
  - 數字之和為 $(0 + 1 + 2) + (0 + 1 + 3) + (0) + (0 + 1 + 3) + (0 + 2 + 3 + 4) = 20$。
- 第三次操作 $\text{Query}_3 = \texttt{mex-add}$ 之後：
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0, 1\}$
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 1, 2, 3, 4\}$
  - 數字之和為 $(0 + 1 + 2) + (0 + 1 + 3) + (0 + 1) + (0 + 1 + 3) + (0 + 1 + 2 + 3 + 4) = 22$。

輸入範例 2

```
3 8
2 0 2
1 2
0
toggle 0
toggle 1
toggle 0
toggle 2
toggle 0
toggle 1
toggle 0
toggle 2
```

該範例輸入符合子任務 1, 3 的限制。

輸出範例 2

```
5
2
2
4
4
7
7
5
```

輸入範例 3

```
4 10
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
toggle 2
mex-add
toggle 1
mex-add
toggle 0
toggle 3
mex-add
toggle 2
mex-add
mex-add
```

該範例輸入符合子任務 2, 3 的限制。

輸出範例 3

```
8
8
12
24
24
12
12
4
12
24
```

7_Boring Data Structure

(3 points/6 points/11 points)

Time Limit: 1 second
Memory Limit: 256MB

Statement

Raana-chan has recently become obsessed with functions, especially a function called mex (**m**inimum **ex**cluded value). The mex function is defined as follows:

- For a set S, $\text{mex}(S)$ is the smallest non-negative integer x such that x is not in S.
  - For example: $\text{mex}(\{0,1,3,5\}) = 2$, $\text{mex}(\{1,3\}) = 0$, $\text{mex}(\{0,1,2\}) = 3$.

Raana-chan has constructed a data structure problem using this function in her spare time while practicing a new song. Given a number n and n sets $A_1, A_2, \ldots, A_n$, where the i^{th} set contains all integers from 0 to $n-1$ **except** $e_{i,1}, e_{i,2}, \ldots, e_{i,k_i}$. Raana-chan want to design a data structure which supports the following two operations q times:

1. `mex-add`: Let $A =_{\text{def}} A_1 \cap A_2 \cap \cdots \cap A_n$ be the intersection of these n sets and $x =_{\text{def}} \text{mex}(A)$. For $i = 1, 2, \ldots, n$, if x is not in A_i, then add x to A_i.
2. `toggle` x: For $i = 1, 2, \ldots, n$, if x is in A_i, then remove it from A_i; otherwise, add x to A_i.

Please output the sum of all numbers in the sets after each operation.

Raana-chan thought this problem was too boring and she went to eat matcha parfait. As the next user of the practice room, Neko-chan accidentally saw the problem left in the room while practicing guitar, and handed it to you. Can you solve this problem? ~~Or do you want to hand the problem to your teammates again?~~

P.S. Definition of intersection: x exists in $S_1 \cap S_2 \cap \cdots \cap S_k$ if and only if x exists in all $S_1, S_2, \ldots, S_k$.

Input Format

- line 1: $n \;\; q$
- line $1+i$ $(1 \le i \le n)$: $k_i \;\; e_{i,1} \;\; e_{i,2} \;\; \cdots \;\; e_{i,k_i}$
- line $n+1+j$ $(1 \le j \le q)$: Query_j

Where Query_j is in the following format:

- `mex-add`
- `toggle` x_j

All variable is as described above.

Output Format

- line 1: ans

Constraints

- $1 \leq n \leq 300\,000$.
- $1 \leq q \leq 300\,000$.
- $0 \leq k_i \leq n$ ($1 \leq i \leq n$).
    - $\sum_{i=1}^{n} k_i \leq 1\,000\,000$.
- $0 \leq e_{i,1} < e_{i,2} < \cdots < e_{i,k_i} \leq n-1$ ($1 \leq i \leq n$).
- $op_j \in \{\texttt{mex-add}, \texttt{toggle}\}$ ($1 \leq j \leq q$).
- $0 \leq x_j \leq n-1$ ($1 \leq j \leq q$ and $op_j = \texttt{toggle}$).
- All numbers in the input are integers.

Subtasks

1. (3 points) $op_j = \texttt{toggle}$ ($1 \leq j \leq q$).
2. (6 points) $k_i = n$ ($1 \leq i \leq n$).
3. (11 points) No additional constraints.

Test Cases

Input 1

```
5 8
1 3
1 2
4 0 1 2 3
1 2
3 0 1 4
toggle 4
mex-add
mex-add
toggle 4
toggle 2
mex-add
toggle 3
mex-add
```

This sample input satisfies the constraints of Subtask 3.

Output 1

```
20
20
22
34
36
40
37
46
```

- In the beginning:
  - $A_1 = \{0, 1, 2, 4\}$
  - $A_2 = \{0, 1, 3, 4\}$
  - $A_3 = \{4\}$
  - $A_4 = \{0, 1, 3, 4\}$
  - $A_5 = \{2, 3\}$
- After $\text{Query}_1 = \texttt{toggle}\ \ 4$:
  - $A_1 = \{0, 1, 2\}$ (remove 4)
  - $A_2 = \{0, 1, 3\}$ (remove 4)
  - $A_3 = \{\}$ (remove 4)
  - $A_4 = \{0, 1, 3\}$ (remove 4)
  - $A_5 = \{2, 3, 4\}$ (add 4)
  - The sum is $(0 + 1 + 2) + (0 + 1 + 3) + (0 + 1 + 3) + (2 + 3 + 4) = 20$.
- After $\text{Query}_2 = \texttt{mex-add}$:
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0\}$ (add 0)
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 2, 3, 4\}$ (add 0)
  - The sum is $(0 + 1 + 2) + (0 + 1 + 3) + (0) + (0 + 1 + 3) + (0 + 2 + 3 + 4) = 20$.
- After $\text{Query}_3 = \texttt{mex-add}$:
  - $A_1 = \{0, 1, 2\}$
  - $A_2 = \{0, 1, 3\}$
  - $A_3 = \{0, 1\}$
  - $A_4 = \{0, 1, 3\}$
  - $A_5 = \{0, 1, 2, 3, 4\}$
  - The sum is $(0 + 1 + 2) + (0 + 1 + 3) + (0 + 1) + (0 + 1 + 3) + (0 + 1 + 2 + 3 + 4) = 22$.

Input 2

```
3 8
2 0 2
1 2
0
toggle 0
toggle 1
toggle 0
toggle 2
toggle 0
toggle 1
toggle 0
toggle 2
```

This sample input satisfies the constraints of Subtasks 1, 3.

Output 2

```
5
2
2
4
4
7
7
5
```

Input 3

```
4 10
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
4 0 1 2 3
toggle 2
mex-add
toggle 1
mex-add
toggle 0
toggle 3
mex-add
toggle 2
mex-add
mex-add
```

This sample input satisfies the constraints of Subtasks 2, 3.

Output 3

```
8
8
12
24
24
12
12
4
12
24
```