

Problem 3 - Decks of Cards (50 pts)

Problem Description

Given N decks of cards, all initially empty, we want to perform four types of operations on them:

1. Add a card numbered u to the top of deck i .
2. Remove a card from the top of deck i .
3. Move all cards from deck i onto the top of deck j . If deck i has cards 3 4 from top to bottom, and deck j has cards 2 5 6 9 from top to bottom. The resulting deck j will have 3 4 2 5 6 9 from top to bottom.
4. Merge deck i into deck j in a shuffling manner. If deck i has cards 3 4 from top to bottom, and deck j has cards 2 5 6 7 from top to bottom. The resulting deck j will have 3 2 4 5 6 7 from top to bottom.

The last operation merges two decks of cards according to the following procedure:

1. Move the top card from deck i to the bottom of the temporary deck if deck i is not empty.
2. Move the top card from deck j to the bottom of the temporary deck if deck j is not empty.
3. If one of the two decks is empty, directly move all the cards in the other deck, from top to bottom, to the bottom of the temporary deck. Otherwise, return to step 1.
4. Move the whole temporary deck to deck j .

After all operations, please output the cards in all the decks.

Input

The first line includes two integers, N and M , representing the number of decks and operations, respectively. The next M lines include 2 or 3 integers. The first integer, t , represents the operation type:

- Type 1 is for the “Add” operation, followed by two integers, i and u , representing the target deck and the card number, respectively.
- Type 2 is for the “Remove” operation, followed by only one integer, i , representing the target deck.

- Type 3 is for the “Move” operation, followed by two integers, i and j , representing the source and destination decks, respectively.
- Type 4 is for the “Merge” operation, followed by two integers, i and j , representing the source and destination decks, respectively.

Output

The output should consist of N lines. For the i -th line, first output c_i , the number of cards in the i -th deck. Then, output c_i numbers representing each card in deck i from top to bottom. All numbers should be separated by a space.

Constraints

- $1 \leq N, M \leq 10^5$
- $1 \leq i, j \leq N$
- $1 \leq t \leq 4$
- $1 \leq u \leq 10^9$
- When executing the “Remove” operation, deck i will not be empty.
- When executing the “Move” or “Merge” operation, $i \neq j$.

Subtasks

Subtask 1 (10 pts)

- $1 \leq N, M \leq 10^3$

Subtask 2 (5 pts)

- only “Add” and “Remove” operations

Subtask 3 (10 pts)

- only “Add”, “Remove”, and “Move” operations

Subtask 4 (25 pts)

- no other constraints

Sample Testcases

Sample Input 1

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

Sample Output 1

```
1 1
2 4 3
```

Sample Input 2

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

Sample Output 2

```
0
4 2 1 4 3
```

Sample Input 3

```
2 7
1 1 1
1 1 2
1 2 3
1 2 4
1 2 5
1 2 6
4 1 2
```

Sample Output 3

```
0
6 2 6 1 5 4 3
```

Hints

- You can use *linked lists* to effectively solve this problem. If you are not familiar with what a linked list is, you can review it with this problem on the Judge Girl system:

<http://140.113.150.142/problem/0/11258>

- When one of the decks is empty, it would be more efficient to directly concatenate the remaining deck to the bottom of the temporary deck.