Problem 4 - Dragon's Song: Awakening (100 pts + 20 bonus pts)

Time Limit: 3 s
Memory Limit: 524288 KB

Problem Description

N classmates of Little Shimeming are playing Dragon's Song: Awakening (DSA), a mobile game notorious for its inflationary powers. The classmates are labeled from 1 to N based on their initial ranks in the game. Each of them initially possesses a power denoted as $p_1, p_2, \cdots, p_N$, where $p_1 \geq p_2 \geq \cdots \geq p_N$.

The game proceeds with T incidents happening sequentially. The i-th incident is one of the following:

1. The classmate labeled a_i at rank j attacks! That is, ze increases zir power to be the same as the classmate that is right in front of zir (i.e. the classmate at rank $j-1$). Then the two classmates swap their ranks.
    - If the classmate labeled a_i is already at the first place, nothing happens and **the attack does not count**.
    - If the two classmates are of the same power, there is no power increase but the swapping should still happen. That is, **the attack still counts**.
2. The game rewards each classmate according to zir rank. The classmate at rank 1 increases zir power by $(N-1)$ points, the classmate at rank 2 increases zir power by $(N-2)$ points, and so on.
3. Little Shimeming can query the last rank that is of power $\geq q_i$, and report the label of the classmate at that rank.
4. Little Shimeming can query the total power increase of classmate labeled b_i from the last m_i attacks, where m_i is no more than some known constant M.

Input

The first line contains three space-separated integers N, T, and M. The second line contains N integers p_1, p_2, $\cdots$, p_N. The i-th line of the following T lines contains one, two, or three integers depending on the incident:

- 1 a_i, indicating incident 1 happens.
- 2, indicating incident 2 happens.
- 3 q_i, indicating incident 3 happens.
- 4 b_i m_i, indicating incident 4 happens.

Output

The output consists of two parts separated by an empty line. The first part consists of the answers to incidents 3 and 4.

- For each incident 3, output 2 integers in a line, representing the last rank that is of power $\geq q_i$, and report the label of the classmate at that rank. If there is no such classmate, please output two numbers `0 0` as a single line.
- For each incident 4, output 1 integer in a line, the total power increase of classmate labeled b_i from the last m_i attacks. If the number of attacks from b_i is less than m_i, output the total power increase of classmate labeled b_i from all attacks.

The second part consists of the game record. It contains N lines, and the j-th line is the record for the classmate labeled j. The first integer k_j of the line denotes the number of attacks that the classmate has executed. Then, the next k_j numbers denote the power gained from (i.e. the difference) the first attack, the second attack, etc. Please separate the integers by space.

Constraints

- $1 \leq N \leq 10^6$
- $1 \leq T \leq 5 \cdot 10^5$
- $1 \leq M \leq 5 \cdot 10^5$
- $0 \leq p_N \leq p_{N-1} \leq \cdots \leq p_1 \leq 10^9$
- $1 \leq a_i, b_i \leq N$
- $0 \leq q_i \leq 10^{18}$
- $1 \leq m_i \leq M$

Subtasks

Subtask 1 (10 pt)

- $1 \leq N \leq 10^3$
- $1 \leq T \leq 10^3$
- $m_i = M$

Subtask 2 (10 pts)

- Only incidents 1 and 3 occur.

Subtask 3 (30 pts)

- $m_i = M = 1$

Subtask 4 (50 pts)

- $m_i = M$

Subtask 5 (20 bonus pts)

- No other constraints, i.e., it is possible that $m_i < M$.

Sample Test Cases

Sample Input 1 (basic)

```
5 6 10
20 15 10 10 0
1 1
1 4
1 4
2
3 16
4 4 10
```

Sample Output 1

```
3 2
5

0
0
0
2 0 5
0
```

Explanation for Sample 1

The initial list of number(power), ordered by ranks, is: 1(20) 2(15) 3(10) 4(10) 5(0). After each incident, the list becomes:

1. No changes occur because the classmate labeled 1 is already at the first place.
2. 1(20) 2(15) 4(10) 3(10) 5(0)
3. 1(20) 4(15) 2(15) 3(10) 5(0)
4. 1(24) 4(18) 2(17) 3(11) 5(0)
5. No changes occur.
6. No changes occur.

Sample Input 2 (basic)

```
8 11 2
52 43 41 41 26 20 20 18
3 41
2
3 100
1 5
1 6
2
1 6
2
4 6 2
1 6
4 6 2
```

Sample Output 2

```
4 4
0 0
24
4

0
0
0
0
1 16
3 23 1 3
0
0
```

Sample Input 3 (bonus)

```
3 7 20
487 6 3
4 3 19
1 3
4 3 1
1 3
4 3 1
1 3
4 3 8
```

Sample Output 3

```
0
3
481
484

0
0
2 3 481
```

Hints

Please be aware that the bonus subtask is intended as an optional extra, meaning that solving it is not obligatory. However, if you are keen on tackling it, the data structure introduced in Problem 3 could serve as one potential tool, alongside others that you will learn later in this semester.