Problem D. Monkey

Problem Description

At "Animal World Sanctuary" zoo, there are n monkeys live and work for the zookeeper.

These monkeys are numbered from 1 to n.

They follow certain rules:

1. Each monkey can oversee up to two other monkeys.
2. Each monkey should be overseen by another monkey. However, there's a monkey king who doesn't listen to anyone else.
3. The monkeys with no one to oversee are called "little brother monkeys."

From these certain rules we can find that the monkey relationship can be represented as a binary tree.

To avoid spending too much time on paying bananas as the monkeys' salary, you decide to give all the bananas and the salary list to the monkey king. However, there are two questions bothering you:

- Who is the monkey king?
- To prevent the monkey king from stealing away others' salary, you want to know how many bananas should the "little brother monkeys" have eaten in total.

Can you write a program to find out?

Input Format

Given the monkey relationship with post-order and in-order.

- line 1: n as the number of monkey
- line 2: b_1 b_2 $\ldots$ b_n as the banana given to i^{th} monkey.
- line 3: in_1 in_2 $\ldots$ in_n as the in-order representation of the monkey relationship.
- line 4: $post_1$ $post_2$ $\ldots$ $post_n$ as the post-order representation of the monkey relationship.

Output Format

- line 1: index of the king monkey.
- line 2: the sum of bananas eaten by all of the "little brother monkey."

Constraints

- $1 \leq n \leq 2\,000\,000$.
- $1 \leq b_i \leq 10^9$ for $i = 1, 2, \ldots, n$.
- $\{in_1, in_2, \ldots, in_n\}$ is a permutation of $\{1, 2, \ldots, n\}$.
- $\{post_1, post_2, \ldots, post_n\}$ is a permutation of $\{1, 2, \ldots, n\}$.
- The relationship could be constructed from the in-order and post-order representations.
- All input values are integers.

Subtasks

1. (60 points) Monkey only oversee those whose index is bigger them itself.
2. (40 points) No additional constraints.

No.	Testdata Range	Time Limit (ms)	Memory Limit (KiB)
Samples	1-2	1000	262144
1	3-15	1000	262144
2	1-28	1000	262144

Samples

Sample Input 1

```
7
10 3 7 4 5 6 8
4 2 5 1 6 3 7
4 5 2 6 7 3 1
```

This sample input satisfies the constraints of all the subtasks.

Sample Output 1

```
1
23
```

Sample Input 2

```
5
50 20 40 10 30
5 4 3 2 1
1 2 3 4 5
```

This sample input satisfies the constraints of Subtask 2.

Sample Output 2

```
5
50
```