Problem C
Cards

Time limit: 2 seconds

Memory limit: 2048 megabytes

Problem Description

Diana is a student who likes to play various types of board games. Today, she receives a deck of cards from her teacher as her birthday gift!

The deck of cards is special: there are n cards in the deck, and each card has a number on its front and another number on its back. Each number on the front or the back is an integer from 1 to n. Furthermore, all n numbers on the front are unique, and so are the n numbers on the back. In other words, numbers on the front and the back are two different permutations of numbers from 1 to n.

Apart from board games, Diana is also interested in mathematics and computer science. While she is playing with those cards, the concept of inversions in a permutation comes to her mind. An inversion is defined as a pair of indices (i, j) such that $i < j$ and the element at position i is greater than the element at position j. In other words, an inversion represents a situation where two elements are "out of order" relative to their positions. A permutation has inversion count c if there are c inversions can be found within it.

Diana wonders if she could rearrange the cards in some order so that the permutation on the front has the same inversion count as the permutation on the back (she cannot flip or throw away some cards). She cannot solve the problem in a while, so she wants to hear your solution.

In formal, you are given two permutations of integers from 1 to n: $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$. You have to find another permutation of the first n positive integers $p_1, p_2, \ldots, p_n$, such that $a' = [a_{p_1}, a_{p_2}, \ldots, a_{p_n}]$ and $b' = [b_{p_1}, b_{p_2}, \ldots, b_{p_n}]$ have the same inversion count. Output the sequences a' and b'.

Input Format

The first line of the input contains an integer n, denoting the number cards in the deck. The second line of the input contains n integers $a_1, a_2, \ldots, a_n$, where a_i is the number on the front of the i-th card. The third line of the input contains n integers $b_1, b_2, \ldots, b_n$, where b_i is the number on the back of the i-th card.

Output Format

If it is impossible to rearrange the cards so that the aforementioned condition is satisfied, print `No`. Otherwise, print `Yes` in the first line of the output. Then in the second line of the

output, print n integers $a'_1, a'_2, \ldots, a'_n$, denoting the numbers on the front of the cards after the rearrangement. In the third line of the output, print n integers $b'_1, b'_2, \cdots, b'_n$, denoting the numbers on the back of the cards after the rearrangement.

If there are multiple possible solutions, print any of them.

Technical Specification

- $1 \leq n \leq 5 \times 10^5$
- $1 \leq a_i \leq n$ for $i \in \{1, 2, \ldots, n\}$
- $1 \leq b_i \leq n$ for $i \in \{1, 2, \ldots, n\}$
- It is guaranteed that $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ are both permutations of integers $1, 2, \ldots, n$.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

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

Sample Output 2

```
No
```

Sample Input 3

```
10
7 4 3 1 6 10 5 2 9 8
8 6 2 9 5 10 7 1 4 3
```

Sample Output 3

```
Yes
2 3 8 1 4 5 9 6 7 10
1 2 3 9 6 7 4 5 8 10
```

Sample Input 4

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

Sample Output 4

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