Tree _order

Description

A *tree* is a connected acyclic graph.

A *binary tree* is a tree for which each node has a left child, a right child, both, or neither, e.g.

```
   1
  / \
 2   3
/ \   \
4   5   6
```

There are three common ways to recursively traverse such a tree.

1. Pre-order: parent, left subtree, right subtree
2. Post-order: left subtree, right subtree, parent
3. In-order: left subtree, parent, right subtree

Given pre-order, post-order, and in-order traversals, determine if they can be of the same binary tree.

For example,

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

are the pre-order, post-order, and in-order traversals of the tree above.

But

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

cannot be the pre-order, post-order, and in-order traversals of the same binary tree.

Input

The first line is the number of nodes in each traversal, 0 < N <= 8000.
The second line is the N space separated nodes of the pre-order traversal.
The third line is the N space separated nodes of the post-order traversal.
The fourth line is the N space separated nodes of the in-order traversal.

Each traversal is a sequence of the nodes, numbered 1 to N, without repetition.

Output

Print "yes" if all three traversals can be of the same tree, and "no" otherwise.

Input

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

Input

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

Output

```
yes
```

Output

```
no
```