Binary Tree Traversal

Problem ID: tree

Problem Statement

You are given a **binary tree** with n nodes, rooted at node 1. The tree is specified using edges between nodes.

Each node has a unique integer label from 1 to n. Your task is to output the **inorder**, **preorder**, and **postorder** traversals of this tree.

Traversal Definitions

- **Preorder**: Visit the root, then recursively visit the left subtree, then the right subtree.
- **Inorder**: Recursively visit the left subtree, then the root, then the right subtree.
- **Postorder**: Recursively visit the left subtree, then the right subtree, then the root.

If a node has no left or right child, skip that direction during traversal. When a node has two children, the **smaller numbered child is the left child**.

Input

- The first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of nodes.
- Each of the next $n - 1$ lines contains two space-separated integers u and v ($1 \leq u, v \leq n$), representing an edge between nodes u and v.

The input is guaranteed to form a valid binary tree rooted at node 1. In particular: every node, when rooted at node 1, will have exactly 0 or 2 children.

Output

Print three lines:

- The **inorder traversal** of the tree (space-separated).
- The **preorder traversal** of the tree.
- The **postorder traversal** of the tree.

Each line should contain exactly n integers representing the node labels in the corresponding traversal order.

Subtasks

Subtask	Constraints	Points
1	$n = 1$	5
2	$n = 3$	15
3	No additional constraints	80

Sample Input 1

```
3
1 2
1 3
```

Sample Output 1

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

Explanation for Sample 1

The tree looks like:

```
    1
   / \
  2   3
```

- **Inorder**: Left → Root → Right → 2 1 3
- **Preorder**: Root → Left → Right → 1 2 3
- **Postorder**: Left → Right → Root → 2 3 1

End of Problem 3