

CONSTRUCT A BINARY TREE FROM INORDER AND PREORDER TRAVERSAL

Write an efficient algorithm to construct a binary tree from the given inorder and preorder sequence.

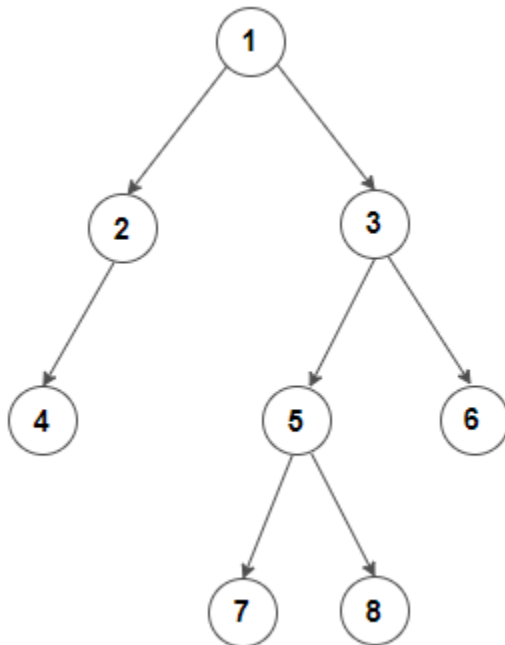
For example,

Input:

Inorder Traversal : { 4, 2, 1, 7, 5, 8, 3, 6 }

Preorder Traversal: { 1, 2, 4, 3, 5, 7, 8, 6 }

Output: Below binary tree



The idea is to start with the root node, which would be the first item in the preorder sequence, and find the boundary of its left and right subtree in the inorder sequence. To find the boundary, search for the index of the root node in the inorder sequence. All keys before the root node in the inorder sequence become part of the left subtree, and all keys after the root node become part of the right subtree. Repeat this recursively for all nodes in the tree and construct the tree in the process.

To illustrate, consider the following inorder and preorder sequence:

Inorder : { 4, 2, 1, 7, 5, 8, 3, 6 }

Preorder : { 1, 2, 4, 3, 5, 7, 8, 6 }

The root will be the first element in the preorder sequence, i.e., 1. Next, locate the index of the root node in the inorder sequence. Since 1 is the root node, all nodes before 1 in the inorder sequence must be included in the left subtree, i.e., {4, 2} and all the nodes after 1 must be included in the right subtree, i.e., {7, 5, 8, 3, 6}. Now the problem is reduced to building the left and right subtrees and linking them to the root node.

Left subtree:

Inorder : {4, 2}

Preorder : {2, 4}

Right subtree:

Inorder : {7, 5, 8, 3, 6}

Preorder : {3, 5, 7, 8, 6}