Problem Link:

<http://leetcode.com/problems/find-elements-in-a-contaminated-binary-tree/submissions/1550255784/?envType=daily-question&envId=2025-02-21>

Solution:

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode() : val(0), left(nullptr), right(nullptr) {}

\* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}

\* TreeNode(int x, TreeNode \*left, TreeNode \*right) : val(x), left(left), right(right) {}

\* };

\*/

class FindElements {

public:

unordered\_set<int> v;

FindElements(TreeNode\* root) {

recover(root, 0);

}

void recover(TreeNode\* node, int value) {

if(node == nullptr)

{

return;

}

node->val = value;

v.insert(value);

if(node->left != nullptr)

{

recover(node->left, 2 \* value + 1);

}

if(node->right != nullptr)

{

recover(node->right, 2 \* value + 2);

}

}

bool find(int target) {

return v.find(target) != v.end();

}

};

/\*\*

\* Your FindElements object will be instantiated and called as such:

\* FindElements\* obj = new FindElements(root);

\* bool param\_1 = obj->find(target);

\*/