Code:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */

void findLeaves(struct TreeNode* node, int** leafValues, int* size, int*
capacity) {
    if (node == NULL) {
        return;
    }

    if (node->left == NULL && node->right == NULL) {
        if (*size >= *capacity) {
            *capacity *= 2;
            *leafValues = (int*) realloc(*leafValues, *capacity * sizeof(int));
        }
        (*leafValues)[(*size)++] = node->val;
    }

    findLeaves(node->left, leafValues, size, capacity);
    findLeaves(node->right, leafValues, size, capacity);
}

bool leafSimilar(struct TreeNode* root1, struct TreeNode* root2) {
    int *leaves1 = (int*) malloc(sizeof(int) * 10);
    int size1 = 0, capacity1 = 10;

    int *leaves2 = (int*) malloc(sizeof(int) * 10);
    int size2 = 0, capacity2 = 10;

    findLeaves(root1, &leaves1, &size1, &capacity1);
    findLeaves(root2, &leaves2, &size2, &capacity2);

    if (size1 != size2) {
        free(leaves1);
        free(leaves2);
        return false;
```

```
    }

    for (int i = 0; i < size1; i++) {
        if (leaves1[i] != leaves2[i]) {
            free(leaves1);
            free(leaves2);
            return false;
        }
    }

    free(leaves1);
    free(leaves2);
    return true;
}
```

Output: