

2. Invert Binary Tree

Shashank Patel CJ

1BM22CS255

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */
void invert(struct TreeNode ** root){
    if(*root !=NULL){
        struct TreeNode * temp = (*root)->left;
        (*root)->left= (*root)->right;
        (*root)->right = temp;;

        invert(&(*root)->left);
        invert(&(*root)->right);
    }
}
struct TreeNode* invertTree(struct TreeNode* root) {
    invert(&root);
    return root;
}
```

Output:

Accepted Runtime: 5 ms

• Case 1 • Case 2 • Case 3

Input

```
root =  
[4,2,7,1,3,6,9]
```

Output

```
[4,7,2,9,6,3,1]
```

Expected

```
[4,7,2,9,6,3,1]
```

Description Editorial Solutions Submissions

← All Submissions

Accepted

ShashankPatelCJ submitted at Feb 21, 2024 22:10

Editorial Solution

Runtime

3 ms

Beats 53.63% of users with C

Memory

5.78 MB

Beats 73.52% of users with C

60%

40%

20%

0%

1ms 2ms 3ms

1ms 2ms 3ms

Code

```
1 /**  
2  * Definition for a bin  
3  * struct TreeNode {  
4  *     int val;  
5  *     struct TreeNode  
6  *     struct TreeNode  
7  * };  
8  */  
9 /**  
10 * Definition for a bin  
11 * struct TreeNode {  
12 *     int val;  
13 *     struct TreeNode  
14 *     struct TreeNode  
15 * };  
16 */  
17 void invert(struct Tre  
18     if(*root !=NULL){  
19         struct TreeNo  
20         (*root)->left  
21         (*root)->right  
22  
23         invert(&(*roo  
24         invert(&(*roo  
25     }
```

ShashankPatelCJ

Access all features with our Premium subscription!

My Lists Notebook Submissions

Progress Points Session

Try New Features

Orders

My Playgrounds

Revert to old version

Appearance

Sign Out

Saved to local

Ln 30, Col 2

Testcase Test Result