

## </> Code

C Auto

```
1 struct TreeNode* invertTree(struct TreeNode* root){
2     // Base case...
3     if(root == NULL){
4         return root;
5     }
6
7     invertTree(root->left);
8
9     invertTree(root->right);
10
11     struct TreeNode* curr = root->left;
12     root->left = root->right;
13     root->right = curr;
14     return root;
15 }
```

Accepted

Sarvesh Rastogi submitted at Feb 19, 2024 12:10

Editorial

Solution

Runtime

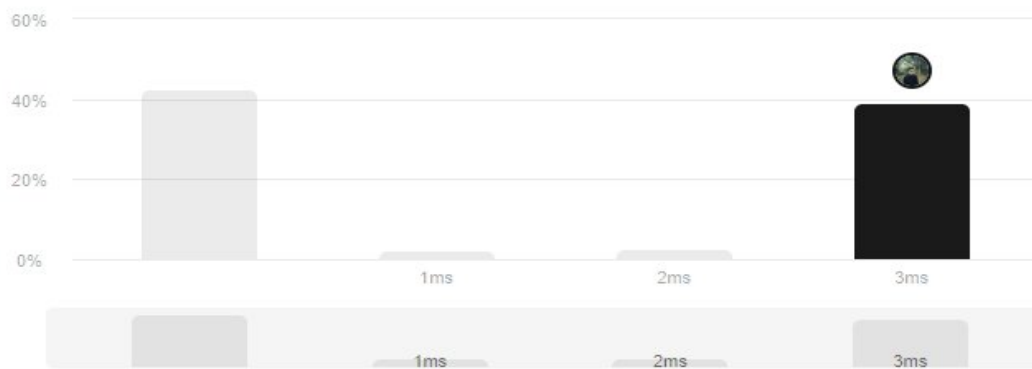
3 ms

Beats 53.08% of users with C

Memory

6.05 MB

Beats 55.13% of users with C



Code | C

```
struct TreeNode* invertTree(struct TreeNode* root){
    // Base case...
    if(root == NULL){
        return root;
    }

    invertTree(root->left);
```

View more

```

77  *
78  */
79  int findMergeNode(SinglyLinkedListNode* head1, SinglyLinkedListNode* head2) {
80      if(head1 && !head1->next)
81          return head1->data;
82      else if(head2 && !head2->next)
83          return head2->data;
84
85      struct SinglyLinkedListNode* iter = head1;
86      int arr1[10000], arr2[10000], arr11[10000], arr21[10000];
87      int count1, count2;
88      count1 = count2 = 0;
89      while(iter)
90      {
91          arr1[count1] = iter->data + (int)iter;
92          arr11[count1++] = iter->data;
93          iter = iter->next;
94      }
95      iter = head2;
96      while(iter)
97      {
98          arr2[count2] = iter->data + (int)iter;
99          arr21[count2++] = iter->data;
100         iter = iter->next;
101     }
102     arr1[count1] = '\0';
103     arr2[count2] = '\0';
104
105     for(int i = 1; i < (count1 <= count2 ? count1 : count2) + 1; i++)
106     {
107         if(arr1[count1 - i] != arr2[count2 - i])
108             return arr11[count1 - i + 1];
109     }
110     return arr21[0];
111 }

```

Line: 111 Col: 1

## Congratulations

You solved this challenge. Would you like to challenge your friends? [f](#) [t](#) [in](#)

Next Challenge

### Test case 0

### Test case 1

### Test case 2

### Test case 3

### Test case 4

### Test case 5

### Test case 6

Input (stdin)

Download

```

1 1
2 1
3 3
4 1
5 2
6 3
7 1
8 1

```

Expected Output

Download

```

1 2

```