

MERGE TWO BINARY TREES

LEETCODE PROGRAM BATCH-3

```
if (root1 == NULL)
```

```
    return root2;
```

```
if (root2 == NULL)
```

```
    return root1;
```

```
struct TreeNode* merged = (struct TreeNode*)malloc(sizeof(struct TreeNode));
```

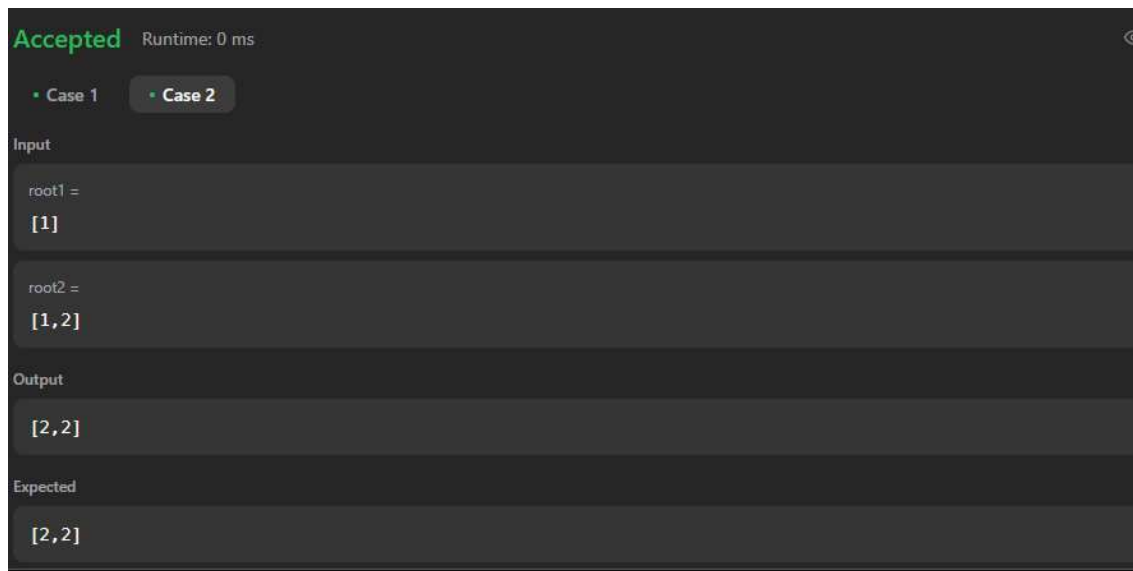
```
merged->val = root1->val + root2->val;
```

```
merged->left = mergeTrees(root1->left, root2->left);
```

```
merged->right = mergeTrees(root1->right, root2->right);
```

```
return merged;
```

The screenshot displays a LeetCode submission interface with a dark theme. At the top, it shows 'Accepted' in green and 'Runtime: 0 ms'. Below this, there are two tabs: 'Case 1' (selected) and 'Case 2'. The 'Input' section contains two text boxes: 'root1 =' with the value '[1,3,2,5]' and 'root2 =' with the value '[2,1,3,null,4,null,7]'. The 'Output' section shows a text box with the value '[3,4,5,5,4,null,7]'. The 'Expected' section shows a text box with the value '[3,4,5,5,4,null,7]'. A small eye icon is visible in the top right corner of the interface.



Merge TWO SORTED LINKED LISTS CODE

HACKERRANK PROGRAM BATCH-3

```
if (head1 == NULL) return head2;
    if (head2 == NULL) return head1;

    struct SinglyLinkedListNode* mergedHead = NULL;

    if (head1->data <= head2->data) {
        mergedHead = head1;
        head1 = head1->next;
    } else {
        mergedHead = head2;
        head2 = head2->next;
    }
```

```
struct SinglyLinkedListNode* current = mergedHead;
```

```
while (head1 && head2) {
```

```
    if (head1->data <= head2->data) {
```

```
        current->next = head1;
```

```
        head1 = head1->next;
```

```
    } else {
```

```
        current->next = head2;
```

```
        head2 = head2->next;
```

```
    }
```

```
    current = current->next;
```

```
}
```

```
if (head1) {
```

```
    current->next = head1;
```

```
} else {
```

```
    current->next = head2;
```

```
}
```

```
return mergedHead;
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✔ Sample Test case 0

✔ Sample Test case 1

Input (stdin)[Download](#)

| | |
|---|----|
| 1 | 1 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |
| 6 | 3 |
| 7 | 1 |
| 8 | 2 |
| 9 | 10 |

Your Output (stdout)

| | |
|---|--------------|
| 1 | 1 2 4 5 6 10 |
|---|--------------|

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✔ Sample Test case 0

✔ Sample Test case 1

Input (stdin)[Download](#)

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

Your Output (stdout)

| | |
|---|-----------|
| 1 | 1 2 3 3 4 |
|---|-----------|