Size of Binary Tree

```
#include <stdio.h>
#include <stdlib.h>

struct node
{
	int data;
	struct node *left;
	struct node *right;
};

struct node *newNode(int data)
{
	struct node *temp = (struct node *)malloc(sizeof(struct node));
	temp->data = data;
	temp->left = temp->right = NULL;
	return temp;
}

int sizeOfTree(struct node *root)
{
	return root? (1+sizeOfTree(root->left)+sizeOfTree(root->right)): 0;
}

int main()
{
	struct node *root=NULL;
	root = newNode(10);
	root->left = newNode(20);
	root->right = newNode(30);
	root->left->left = newNode(40);
	root->right->left = newNode(50);
	printf("%d\n", sizeOfTree(root));
	return 0;
}
```

Time complexity: O(n)
Space complexity: O(n)