Diameter of Binary Tree

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

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

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

int max(int a, int b)
{
	return a > b? a: b;
}

int DiameterOfBT(struct BTNode *root, int *height)
{
	int leftTreeHeight = 0, rightTreeHeight = 0, leftTreeDiameter = 0, rightTreeDiameter = 0;
	if(!root)
	{
		*height = 0;
		return 0;
	}
	leftTreeDiameter = DiameterOfBT(root->left, &leftTreeHeight);
	rightTreeDiameter = DiameterOfBT(root->right, &rightTreeHeight);
	*height = max(leftTreeHeight, rightTreeHeight) + 1;
	return max(leftTreeHeight+rightTreeHeight+1, max(leftTreeDiameter, rightTreeDiameter));
}

int main()
{
	struct BTNode *root = newNode(25);
	int height = 0;
	root->left = newNode(20);
	root->left = newNode(10);
	root->right = newNode(30);
	root->left->left = newNode(5);
	root->left->right = newNode(15);
	root->left->right->left = newNode(12);
	printf("Diameter of given binary tree is = %d", DiameterOfBT(root, &height));
}
```

Time complexity: O(n)

Space Complexity: O(n)