Left view 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;
}

void LeftViewOfBT(struct BTNode *root, int level, int *maxLevel)
{
	if(root)
	{
		if(*maxLevel < level)
		{
			printf("%d\t", root->data);
			*maxLevel = level;
		}
		LeftViewOfBT(root->left, level+1, maxLevel);
		LeftViewOfBT(root->right, level+1, maxLevel);
	}
}

int main()
{
	struct BTNode *root = newNode(25);
	int maxLevel = 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);
	LeftViewOfBT(root, 1, &maxLevel);
	return 0;
}
```

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