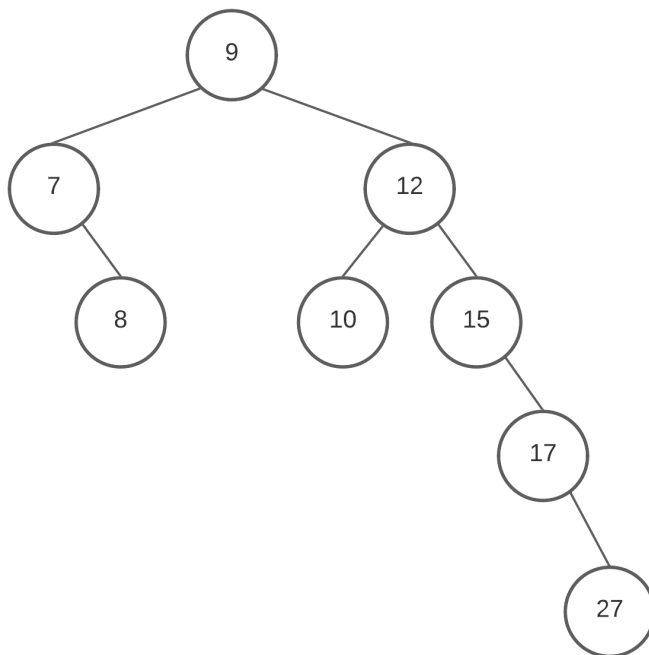


**Yeditepe University**  
**CSE 211 - Data Structures**  
**Assignment 4**

**Deadline: 22nd December 2021, 23:59**

In this assignment, you are going to implement some functions for a tree data structure. You must use the given tree implementation. Just submit the `tree.h` file but change its name into `NAME_SURNAME.h` before submitting (5 points). You do not need to submit the `main.cpp` file. You can use it to test your functions. **Do your own work to stay away from punishment.**



**Below are functions that you need to implement in this assignment, you can also see the expected outputs in the comments for the given tree above. Do not change the function names:**

```
void insertBST(int el); // 10 points
```

```
/* This function inserts el into binary search tree according to the  
rules of BST */
```

```
void insertWrong(int el); // 10 points
```

```
/* This function randomly inserts elements into a tree. It really  
does not matter the way you insert. Just make sure that it does not  
insert like insertBST */
```

```

bool isThisaBST(TreeNode<T>* treeNode); // 20 points

/* This function checks whether the given tree is a BST or not. You
can use the above functions to test this one. For example, it returns
true if the nodes are inserted using insertBST, if the nodes are
inserted using insertWrong it returns false. */

int findHeight(TreeNode<T>* treeNode); // 20 points

/* This function finds the height of a given tree. The height of
given tree in the figure is 5. */

vector<int> findOnlyChild(TreeNode<T>* treeNode); // 30 points

/* This function returns the nodes that do have not any siblings.
The only children in the given figure are 8, 17 and 27 */

void inOrder(TreeNode<T>* treeNode); // 5 points

/* This function prints the in-order traversal of a given tree.
Assume that elements are inserted using insertBST
The output: 7 8 9 10 12 15 17 */

```