

Problem 1(100pt): Template Binary Search Tree

Consider a binary search tree class `BinarySearchTree` without duplicates, where the class is templated by element type `T` and has at least the following public member functions with prototype below:

- `void insert(T data)`: if data already exists, then make no change
- `void erase(T data)`: if data does not exist, then make no change
- `void ascending_print() const`: prints all elements in the ascending order
- `void descending_print() const`: prints all elements in the descending order
- `T largest() const`: returns the largest element; throws a `std::logic_error` if the tree is empty
- `T second_largest() const`: returns the second largest element; throws a `std::logic_error` if the tree is empty or only has one element

In addition, `BinarySearchTree` also has

- a private nested class `TreeNode`

A sample `treemain.cpp` is given. Your output should look exactly the same as follows.

```
*****
*****
Tried to call largest() for an empty tree
*****
Tried to call second_largest() for an empty tree
*****
Tried to call second_largest() for an one-element tree
*****
3 2
*****
3
2
1
0
*****
1
2
3
```

Instructions:

- Your code will be graded based on correctness, efficiency, clearness, and practices. For this homework, the only library header files you are allowed to use are:
 - `iostream`, `iomanip`
 - `string`
 - `stdexcept`
- A binary search tree of strings is given in `stringtree.cpp`. You can modify the code as needed.
- (10pt) Define all of your classes and functions within namespace `UCLA_Spring2021_PIC10C_HW5`
- (5pt) Put all of your code in one file, named `tree.h` and submitted it to CCLE. Do not include `int main` function in your code. Add description of this file in the beginning to show your ownership. A sample description may look like:

```
/*
    PIC 10C Homework 1, Heap.h
    Purpose: Define a template heap class
    Author: John Doe
    Date: 01/01/2021
*/
```
- (15pt) Good coding practice includes commenting your code, using descriptive variable/function names, using efficient algorithms, etc. Coding practice part will be graded by three levels: 5, 10, 15.
- (70pt) Implement `BinarySearchTree` class as instructed above.
- The official grading compiler is Visual Studio 2019 and you may lose majority of points if your code does not compile. If you don't have VS2019 installed in your computer, you are welcome to check your homework using virtual machines before submission. Please only check your homework after it has satisfying results on your local computer. Manually log out your account after using the machine.