CSCI 2380 **Spring 2025** **Mr. Gustavo Dietrich**

Homework 12: Binary Search Trees **Due Date:** 5/7/25

In this iteration of autocomplete, you'll eliminate the weakness of the previous version: a long "load time" due to slow addition of words to the Autocompleter. Instead of add taking Θ(n) worst-case time, you should aim for Θ(log n) time. All other methods should retain their current speeds. To achieve this, a more advanced data structure is needed: a **binary search tree**.

The following files have been given to you:

1. A C++ header file (autocompleter.h) declaring the **Autocompleter** class[1].
2. A C++ source file (main.cpp) containing a main() function with tests.
3. A text file (words.txt) containing 10000 common words[2].

Create new C++ source file named **autocompleter.cpp** that implements the function declared in autocompleter.h so that autocompleter.cpp and the provided files compile into a program that runs with no failed tests.

Submit just the source code of **autocompleter.cpp**. You don't need to submit the main.cpp nor the other files because I will use my own autocompleter.h and main.cpp files to evaluate your autocompleter.cpp file.

Review the examples discussed in class and the textbook to get an idea of what you need to do. Analyze carefully the tests because that will help you understand how the functions that you need to create work.

Do not hesitate to use the corresponding topic in Discussions to post your questions/doubts about this assignment. I will reply as soon as I can.

IMPORTANT:

Make sure your program compiles and executes in full (it should pass all the tests included in main()).
Your solution must be a generic one, do not hard code it based on the tests just to pass them. You will receive no credit for hardcoded solutions.
You must submit ONLY ONE solution per team.

[1] The public methods of the Autocompleter class are identical to those provided in Homework 10. Only the private (helper) methods and instance variables differ.
[2] Source: http://norvig.com/ngrams/count_1w.txt

Your program must be well commented, use meaningful identifiers, and use indentation to improve its readability.

Your program must have the following comments at the top:

```
//*************************************************************************
// Team #                CSCI 2380          Spring 2025          Homework # 12
// First and Last Name
// First and Last Name
//
//*************************************************************************
```

When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.
Paste the link to your final solution along with your source code in the textbox opened when you click on Create Submission before you click on Submit.
The following is the basic criteria to be used to grade your submission:

You start with 100 points and then lose points as you don't do something that is required.

-12 : Incorrect implementation of size_recurse()
-12 : Incorrect implementation of completion_count_recurse()
-12 : Incorrect implementation of completions_recurse()
-12 : Incorrect implementation of Autocompleter()
-12 : Incorrect implementation of insert()
-12 : Incorrect implementation of size()
-12 : Incorrect implementation of completion_count()
-12 : Incorrect implementation of completions()
-20 : Program crashes when executed
-5 : Unnecessary statements in your code
-40 : Program does not compile
-10 : Missing/too few comments
-20 : Incorrect/missing cpp file or Incorrect/missing link to your Repl.it solution
--100: The code submitted is not your creation
-10 : Late
-100 : No submission.
-100 : No team contribution

Important: more points may be lost for other reasons not specified here.