Homework 6

Out: 11.7.23
Due: 11.15.23

1. [Binary Search Trees, 10 points]
Specify and explain what is the worst-case asymptotic time needed to insert a sequence of log(n) copies each of the numbers 1 through n to an initially empty binary search tree. For example, if $n = 10^{10}$, then we will insert ten 1's, followed by ten 2's, followed by ten 3's, etc. Note that we are referring to a standard binary search tree, which is not necessarily balanced.

2. [AVL Trees, 10 points]
Determine if the following claim is correct. It so, explain why, and if not, give a counter example: The order in which elements are inserted into an AVL tree does not matter, since the same AVL tree will be established following rotations.

3. [B-tree, 10 points]
Determine if the following claim is correct. It so, explain why, and if not, give a counter example: The order in which elements are inserted into a B-tree does not matter, since the same B-tree will be established.

4. [String matching, 20 Points]

a. Draw the FSM to find the pattern "abcc". Assume that $\Sigma=\{a,b,c\}$, and that any letter not in Σ returns to the initial state.

b. Draw a standard trie for the following set of strings:
{pea, call, pencil, can, pet, cat, pen}.

5. [Big Data, 50 points]
This problem involves finding information in a very large data file, *BigData.txt*, similar to what you might find from the dump of a hard drive. The file can be obtained from eng grid: /ad/eng/courses/ec/ec330/BigData.txt or https://drive.google.com/file/d/0B2H-ZCNPqkX2bzhxUlRVdHg4cVE/view?usp=sharing&resourcekey=0-feWTz4uAWol9e5qOz6_b7w
Let us know if you have trouble accessing the file.
We also provide, on Blackboard, a *TinyData.txt* file, which is useful for initial tests of your code.

Within this file, determine:

a. The number of BU-IDs in the file whose digits add to a number greater than 28. A BU ID is defined as anything starting with a U and followed by exactly eight digits and then a non-digit.

b. The number of English words (from the *dictionary.txt* file) that appear in the file that do not end with the same letter of the alphabet as your last name (for example, if your last name ends with the letter 't', then you should exclude dictionary words that end with ‘t’). The words do not need to be delimited by spaces (or other punctuation) in the file, and may overlap. The words must be contiguous e.g: door$knob or door knob will not find doorknob, but will still find door, and knob. Duplicates do count. For example, the word “a”, which is a dictionary word, appears in the data more than once, and should be counted accordingly.

c. The length of the longest palindrome (i.e. a string that equals its reverse) you can find within the text. When finding palindromes, consider ALL characters (spaces should count).

Your code must be submitted in a single file named *Problem5.cpp*. Your code should include three functions, *fiveA*, *fiveB*, and *fiveC*. These functions must have the same declarations as the function declarations in *Problem5.h*, and when run in the same directory as *BigData.txt* they should analyze this file. To receive full credit, your solution needs to be efficient. Your code should be compiled as follows:

```
> scl enable devtoolset-10 bash
> g++ -std=c++17 Problem5.cpp main.cpp
```

Note: You should test your functions with the provided *main.cpp* as well as with your own main function if you wish to do so, but in your submission of *Problem5.cpp* there should be no main function. You may also write other helper functions or classes, but the only requirement is that these 3 functions must exist in the submission file. You may assume that an int has enough space to store the correct values when running with *BigData.txt*.

In addition to your code, submit a file named *Problem5.txt*, containing the output of running the provided *main.cpp* on the *BigData.txt* file, along with a short explanation for how you solved each part. The format of this file should be as follows:

```
Name & Collaborator Name: Tali Moreshet & John Smith
First letter of last name for part b: h

main.cpp output:
The number of BU-IDs whose digits add to a number greater than 28: 100
Number of valid dictionary words that don't end with last letter of last name: 1000
Longest palindrome: 10

Explanation:
a) 1-2 sentences
b) 1-2 sentences
c) 1-2 sentences (or more if necessary)
```