CS 1410 Introduction to Computer Science - CS2
Assignment #1
Given: January 12
Due: January 19 before class
Total Points: 50 points

Problem description

Assume that you are working for Google and you want to write a program to analyze the trends of users' queries, e.g. what are the most frequent search terms. This program accepts input as a text file named input.txt containing search terms, each term is on a single line. Its output is a text file named output.txt which lists all distinct terms in the input file with their corresponding frequencies sorted decreasingly. Each term and its frequency are written on one line. To make the processing more efficient, in the input and output files, a search term is encoded as an unique, integral ID rather than its character sequence.

Example:

```
input.txt
----------
0
1
2
0
3
2
0
```

```
output.txt
------------
0 3
2 2
1 1
3 1
```

Since Google receives billions of queries each day, your program should run as fast and use as little memory as possible. In this assignment, your program is expected to be able to process the provided input file with a million (1,000,000) lines and ten thousand (10,000) distinct search terms within 10 minutes.

The following solution is potentially able to achieve that requirement. First, you will store all the distinct search terms and their frequencies in an array sorted by those terms. When you read a search term from the input file, you will use the binary search algorithm to look for that term in that array. If you find the term, just increase its frequency. If it is not there, you will insert it into the array at its proper location to keep the array sorted. After reading the input file, you sort the array, in descending order, by frequency and write its elements to the output file.

Programming tasks

Now, design and implement the aforementioned ideas with the following instructions.

Task 1 (10 points). Declare a class TermTable to store all distinct search terms and their frequencies in two arrays of 10,000 elements: one for the terms (i.e., term array) and one for their corresponding frequencies (i.e., frequency array). This class also has a field (e.g., a data member) to store the current number of terms stored in those arrays. The default constructor assigns 0 to that field and to all elements of the two arrays.

Task 2 (10 points). Implement a “binarySearch” method in class TermTable. This method uses the binary search algorithm to look for a search term in the term array.

Task 3 (10 points). Implement an “insert” method in class TermTable to insert a new term to its two arrays (suggest to change it to the following: its term array and insert its count (e.g., 1) to the corresponding frequency array). In other words, this method increases the current number of terms, assigns the given term to the corresponding location in the term array, assigns 1 to the corresponding location in the frequency array, and shifts the elements after the inserted locations to their proper locations (e.g., the next locations) to keep all terms sorted.

Task 4 (10 points). Implement a “sort” method in class TermTable to sort the term and frequency arrays descendingly by the frequencies.

Task 5 (10 points). Implement the main function to read all terms in the input file, process them (i.e. search each term in the TermTable, update its frequency if found or insert it if not found), and write the distinct terms and their frequencies sorted in the descending order to the output.

To read and write files, you could use the following code snippets:

Reading file:

```
   ifstream fin("input.txt");
   while (fin >> term) {
         // do something with term here
   }
   fin.close();
```

Writing file:

```
   ofstream fout("output.txt");
```

```
// to write a value x to fout, e.g. you could use fout << x;
fout.close();
```

You should submit your solution on Canvas as a cpp file named as Firstname_Lastname_HW1.cpp. For example, if your name is Adam Smith, the submitted file will be Adam_Smith_HW1.cpp