San José State University
Department of Computer Engineering

CMPE 180-92
Data Structures and Algorithms in C++

Fall 2017
Instructor: Ron Mak

Assignment #11

Assigned: Saturday, November 4
Due: Thursday, November 9 at 5:30 PM
URL: http://codecheck.it/files/17110409209cpxpjyc2fr10hnjdfadd07au
Canvas: Assignment #11. STL Vector and Map
Points: 100

STL Vector and Map

This assignment will give you practice with the built-in Standard Template Library (STL) map (hash table) by comparing its performance with a sorted STL vector. Your program will read a text file and build two versions of a concordance table, one with a sorted vector and one with a map.

A concordance table is an alphabetical list of words from a document and their frequencies. Your input data will be a text file of the U.S. constitution and its amendments:
http://www.cs.sjsu.edu/~mak/CMPE180-92/assignments/11/USConstitution.txt

Your program should read each word of the text. If the word is not already in the concordance, enter the word with an initial count of 1 into both the vector and map versions. If the word already exists in the concordance, increment the word's count by one in the vector and in the map. The words in the concordance must be unique, and, of course, both versions must end up with the same words and counts. Word comparisons should not be case-sensitive. Do not include numbers or punctuation marks.

Timings

Your program should keep track of how much time it takes to enter all the words into the vector version of the concordance, and how much time it takes to enter all the words into the map version of the concordance. For each word, compute the elapsed time only of the operation of either entering a new word into the concordance or incrementing the count of an existing word. Do not include the time to read the word from the input text file. Compare the total time for the vector vs. the total time for the map. The timings should be in microseconds (μsec).

After building the two versions of the concordance, your program should compare the total time it takes to do 100,000 random word searches in each version. Since your vector-based concordance will be sorted, use a binary search.

Other operations

Your program should use a list of words, both in and not in the concordance, to make (untimed) spot checks of the completed vector and map versions of the concordance to make sure they agree on the frequency counts of those words.
Iterate over the completed vector and the map versions of the concordance in parallel to ensure that they contain the same data (words and counts) in the same order.

Sample output

Since there are timings, CodeCheck will not compare your output. Ignore the score 0. Your output should be similar to:

```
Timed insertions
----------------
      Lines:    865
 Characters: 43,976
      Words:  7,541

Vector size: 1,138
   Map size: 1,138

Vector total insertion time: 41,046 usec
   Map total insertion time:  6,550 usec

Spot checks of word counts
--------------------------
amendment: vector:35 map:35
article: vector:28 map:28
ballot: vector:5 map:5
citizens: vector:18 map:18
congress: vector:60 map:60
constitution: vector:25 map:25
democracy: vector:(not found) map:(not found)
electors: vector:16 map:16
government: vector:8 map:8
law: vector:39 map:39
legislature: vector:13 map:13
people: vector:9 map:9
president: vector:121 map:121
representatives: vector:29 map:29
right: vector:14 map:14
trust: vector:4 map:4
united: vector:85 map:85
vice: vector:36 map:36
vote: vector:16 map:16

Checking concordances
---------------------
Both match!

Timed searches (100,000 searches)
--------------
Vector total search time: 84,308 usec
   Map total search time: 61,287 usec

Done!
```

What to submit

Submit the signed zip file into Canvas: **Assignment #11. STL Vector and Map**. Also submit a text file containing your program's output.

You can submit as many times as necessary to get satisfactory results, and the number of submissions will not affect your score. When you're done with your program, click the "Download" link at the very bottom of the Report screen to download the signed zip file of your solution.

Rubrics

Criteria	Max points
Statistics (should be the same as the sample output) • Total words • Distinct words (vector and map sizes)	**20** • 10 • 10
Word insertions • Vector timing • Map timing	**30** • 15 • 15
Checks • Spot checks (same frequency counts as the sample output) • Matching vector and map concordances	**20** • 10 • 10
Word searches • Vector timing • Map timing	**30** • 15 • 15

Extra credit (10 points max)

Instead of scanning the sorted vector from the beginning to determine where to insert a new word, use a binary search to find the correct insertion position. How much does that change the vector insertion time?

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (http://theory.stanford.edu/~aiken/moss/). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.