CSE 20212 Fundamentals of Computing II
Spring 2016

Lab handout for weeks of April 5 and 12

Objectives

1. Practice using advanced STL as an alternative for a prior algorithm and a new we based data mining problem

2. Have fun!

In-lab activities

1. (1 point) Report to lab on time. Attendance will be taken at the scheduled lab time.

2. Start writing a simple C++ program that will be used to detect/process similar documents. You program may or may not use classes, but should satisfy the following requirements. For now you can assume each word is unique, however, to start #3 with the help of TAs.

    a. Your program should as the user for the names of two plain text files
    b. Using STL string processing, ensure all punctuation is ignored, all formatting text is ignored, and each word is converted to all lower case. For example, the string "HoWeveR," should be converted to "however."

3. Display the number of unique words in the first file, and the words and their counts found in the second file to the user via standard out.

4. If you still have time, please start the post-lab activities.

Post-lab assignment

1. Given a text file, and two lists of words in languages of your choice (dictionaries), write a simple program to guess which language the text file is written (aka a simple "Google translate" preprocessor). Use your imagination although it is simplest to probably use English and another language. For example, a dictionary of Chinese (English) in Pinyin for a child could have "mao" (cat), "gou" (dog), "ping guo" (apple), "yu" (fish), "ma" (horse), etc.

2. The MinHash approximation of Jaccard similarity was originally used by AltaVista to detect similar web pages. Please read more about this in advance of class next week (http://en.wikipedia.org/wiki/MinHash)

3. If we ignoring computational requirements, maps can be used to simplify calculating Jaccard similarity, which can be used in "big data" to detect

plagiarism, similar documents (including the same web page content on different HTML pagers), and in large-scale clustering problems. Compute the Jaccard similarity of the two target documents, which is the ratio of the number of items in the intersection of two sets to their union (see page in #1 below)

4. Sets are another way to solve Suduku. Specifically, instead of having a vector to represent possibilities (and scanning said vector, with O(n) operations) we can use a set that allows finding a possible value in O(log n) time. Modify your previous submission to use sets. Because some of the operations are simpler using c++11, use at least one feature of C++11 in this solution. We will test your Sudoku solver on the "Easy" puzzle unless instructed otherwise (e.g., in the really rare case when your program can solve medium but not easy)

5. Please add the following additional sections to your report, also to be outlined in the rubric

    a. Speculate how many words (or how to seed) your dictionaries should ideally have. Can you use any Unix knowledge here? Files available on the web?

    b. Based on your opinion, is the possibilities vector more natural/easier to code or the set? Is it easier in C++11 or not? Given our limited size of the board there should be no noticeable performances differences but there are larger sudokus you can find online (e.g., 20x20 puzzles) where a faster search could be helpful.

Handing it in

Please read and follow the general lab guidelines available on the course website:

http://www.cse.nd.edu/courses/cse20212/www/labs.html

If you have any questions about these or the grading rubric, please use Piazza.

Coder challenge!

Given our project deadlines we again encourage you to use your extra creative cycles to make your projects more awesome.