Advanced Algorithms, Homework Assignment 1
Due Wednesday, Jan. 20

This is a warm-up assignment. You get to show off your programming and analysis skills by implementing a solution to a simple problem. This is an opportunity for you demonstrate your ability to use common data structures and think about their computational overhead.

Common Substring Counting

Consider the two strings below. The first string has 28 substrings, all of them different. For example “abcd” and “cdef” are both four-character substrings of the first string. The second string has 36 substrings, but some of them are duplicates.

abcdefg
cdefabab

Your job is to develop a program (common.cpp or Common.java) that counts the number of different strings that are substrings of both of two input strings. For example, “f”, “ab” and “cde” are all substrings of the two strings above. The “ab” substring occurs twice in the second string, but you should only count it once.

Your program will read its input from standard in, one string per line. Input string may contain spaces. As output, your program should simply print the number of different strings that are substrings of the two input strings.

Sample Input

abcdefg
cdefabab

Sample Output

13

Analysis

In addition to your program, you are to write up an asymptotic analysis of its worst case running time. This will probably take you a little less than a page. You should include enough detail to permit someone else to understand your approach even if they couldn’t see a copy of your source code. I expect you to cover how you measure the input size, and how the running time of each stage of your algorithm and your associated data structures are dependent on input size. Combine all of the contributions to running time into a single formula and then simplify the formula where possible. Be sure to explain these steps appropriately to the reader.

You will need to prepare your analysis in word processor of your choice. You may use Microsoft Word, HTML, LaTeX or anything else you want, but you need to be able to produce appropriate mathematical notation and formatting in whatever system you use. This is a skill that will come in handy repeatedly as you make progress on your research, so we may as well start getting used to it now.
Submitting Your Work

On the due date, you will need to turn in a paper copy of your source code and analysis at the start of class. Don’t forget to staple your pages together. You will also need to submit an electronic copy. Put a copy of your source code on our department Unix systems and submit it with my submission script, submit.aalg. You should just be able to type the following at the shell prompt.

shell> submit.aalg common.cpp