School of Computer Science
University of Guelph

CIS*3490 The Analysis and Design of Algorithms

Winter 2021
Instructor: Fangju Wang

Assignment 3 (100%)

Question 1 (30%)

An *anagram* is a string formed from another string by rearranging the latter's symbols. For example, *1243* and *1423* are anagrams of *1234*, *eat* and *ate* are anagrams of *tea*, and *AABC* is an anagram of *ABCA*. Write the following programs for detecting anagrams.

1.1 A program to implement a brute force algorithm. (10%)

1.2 A more efficient program based on the presorting technique. (20%)

When a program is executed, it reads in the 30000 strings in file `data_4.txt`, prompts for a string, finds all the anagrams of the string in the file (**not** including the string), displays the anagrams, and reports the total number of anagrams found. A program is required to report the running time for each search (for 1.1 search time, and for 1.2 sorting and search time separately). The running time should not include the time for reading the file.

The file `data_4.txt` will be used to test your programs. You may hard-code the file name in your programs.

Question 2 (70%)

Write the following programs for string search:

2.1 A program to implement a brute force algorithm. (10%)

2.2 A program to implement the Horspool's algorithm. (20%)

2.3 A program to implement the Boyer-Moore algorithm. (30%)

The text is in file `data_5.txt`, which has 44049 lines of strings. A search patterns includes the 52 upper and lower case letters only. Search is case-sensitive. When a program is executed, it reads in the text, prompts for a pattern, finds **all** the occurrences of the pattern in the text, and reports the total number of occurrences found. **Don't** remove any symbols (characters) from the text. A program is required to report the number of pattern shifts and running time for each search.

The file `data_5.txt` will be used to test your programs. You may hard-code the file name in your programs.

2.4 Analyze performance of your brute force and Horspool programs. (10%)

The performance parameters are the number of pattern shifts and running time. To compare two programs, choose ten search patterns of different lengths, and search them by using the programs separately. For each pattern, calculate the ratios of the performance parameters of the two programs. Then, for all the ten patterns, calculate average ratios, and compare and analyze the performance of the two programs in terms of the ratios. Very briefly write your comparison and analysis in the `readme` file submitted with your programs.

Due time: 23:59, Monday March 15, 2021. Submit your work as a tar file to Moodle.