

CIS*3490 The Analysis and Design of Algorithms

Winter 2017

Instructor: Fangju Wang

Assignment 3 (10% + 5% = 15%)

Question 1 (6%)

An *anagram* is a string formed from another 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. (2%)

1.2 A more efficient program based on the presorting technique. (4%)

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, displays the anagrams, and reports the total number of anagrams found. A program is required to report the running time for each search.

Question 2 (4% + 5%)

Write the following programs for string search:

2.1 A program to implement a brute force algorithm. (1%)

2.1 A program to implement the Horspool's algorithm. (3%)

2.3 A program to implement the Boyer-Moore algorithm. (bonus question, 4%)

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. A program is required to report the number of pattern shifts and running time for each search.

2.4 Analyze performance of programs. (bonus question 1%)

The performance parameters are the number of times the patterns shift and running time. To compare two programs, choose ten search patterns of different lengths, and search them 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: 2:30pm, Monday March 13, 2017