CSE2/CSE5ALG– Algorithms and Data Structures – 2019 Assignment – Part 2

Assessment: This part 2 of the assignment is worth 20% of the final mark for this subject.

Due Date: Monday 27 May 2019 at 10:00 AM

Delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime. Penalties are applied to late assignments, accepted up to 5 days after the due date only. Please make the submission after Friday 10 May 2019 at 10:00 AM.

Copying, Plagiarism: Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. The Department of CS and IT treats academic misconduct seriously. When it is detected, penalties are strictly imposed.

Submission Details: Submit all the Java files that are required for the tasks described in this handout. The code has to run under Unix on the latcs8 machine. You submit your files from your latcs8 account. Make sure you are in the same directory as the files you are submitting. Submit each file separately using the submit command. For example, for the file called (say) WordMatch.java, use command:

```
submit ALG WordMatch.java
```

After submitting the files, you can run the following command that lists the files submitted from your account:

```
verify
```

You can submit the same filename as many times as you like before the assignment deadline; the previously submitted copy will be replaced by the latest one.

Testing Platform: While you are free to develop the code for this assignment on any operating system, your solution must run on the latcs8 system. We should be able to compile your classes with the simple command `javac *.java`, and execute your programs with a simple command, e.g. `java WordMatch in1.txt out1.txt in2.txt out2.txt`.

Return of Assignment: Assignments will be returned within three weeks of the due date. Students will be notified by email and via the CSE2/CSE5ALG website when marking sheets are available for collection.

Assignment Objectives

- To understand various data structures and searching and sorting techniques;
- To analyse these techniques in relation to a particular problem;
- To implement these techniques within a Java program.

Background

As described in the handout for Part 1, the overall aim of the assignment is to develop a program to build a lexicon and to find the words that match certain patterns.

Whereas for Part 1 we are only concerned with the correctness of the program, for Part 2 we are concerned with the efficiency. More specifically, you are required to do the tasks described below.

Besides the information given in the tasks below, please refer to Part 1 of the Assignment for any other information you need.

Task 1 (80 marks)

Write a Java program called `WordMatch.java`. This program takes four command-line arguments. For example:

```
java WordMatch in1.txt out1.txt in2.txt out2.txt
```

1. The first is the name of a text file that contains the names of the text files from which the words are to be read to build the lexicon (The aim of this argument is to specify the input files)
2. The second is the name of the text file to which the words in the lexicon are to be written (The argument specifies the file that contains the words and their neighbors in the lexicon)
3. The third is the name of a text file that contains a number of matching patterns, one per line (The aim of this argument is to provide the matching patterns)
4. The fourth is the name of the text file that contains the result of the matching for the given patterns (The argument specifies the file that contains the result)

For this version, the efficiency with which the program performs various operations is a major concern.

For example, the files read in can be quite long and the lexicon of words can grow to be quite lengthy. Time to insert the words will be critical here and you will need to carefully consider which algorithms and data structures you use.

You can use any text files for input to this program. A good source of long text files is at the Gutenberg project (www.gutenberg.com) which is a project aimed to put into electronic form older literary works that are in the public domain. The extract from Jane Austen's book Pride and Prejudice used as the sample text file above was sourced from this web site. You should choose files of lengths suitable for providing good information about the efficiency of your program.

A selection of test files have been posted on LMS for your efficiency testing. You can consider additional test files if you wish.

As expected, the definition of a word, and the content of a query's result and display of this result are exactly the same as what described in Part 1.

All the Java files must be submitted. The program will be marked on correctness and efficiency. Bad coding style and documentation may have up 5 marks deducted.

With the exceptions of ArrayList and LinkedList, you are not permitted to use any of the classes in the Java Collections Framework, e.g. TreeMap, HashMap, Collections, Arrays.

Task 2 - Report (20 marks)

Write a report about the `WordMatch` program and the classes that support it. The report has the following sections:

- **Section 1:** In this section, describe the classes that you implement for Task 1. For each class, list and briefly describe its attributes and methods.
- **Section 2:** In this section,
  (a) Identify what you consider to be the main issues in the effort to make the program run efficiently; for example, what subtasks you think could be costly in terms of execution time
  (b) Point out the techniques/features that you have used to address those issues
- **Section 3:** In this section, explain how you test the functional correctness of your program.
- **Section 4:** In this section, explain how you test the efficiency of your program. Make sure you present quantitative information about its efficiency.

In the report, as well as in every Java class, you must include your student ID and name, and the subject: CSE2ALG or CSE5ALG. Up to 5 marks can be deducted if these details are missing

How to submit the report: The report should be a file named Task2.pdf or Task2.docx. The pdf format is the preferred one. Submit the report file, e.g. `submit ALG Task2.pdf`

Task 3 (Optional)

Develop a solution for the puzzle in which you are given two words, and you need to find a sequence of words to transform the first word into the second by changing one letter at a time.

The key idea is to represent the list of words of the same length as a graph in which each node represents a word, and two nodes are connected by an edge if the two associated words differ from each other only by one letter. Thus, the solution would involve algorithms on graphs which we are yet to cover.

Note: This optional work should NOT be attempted until all other tasks have been successfully completed. Good attempts at these extensions are worth up to an additional 10% of the mark for Part 2. However, it is not possible to end up with more than 100% for the total mark of Part 2.

Task 4 (For CSE5ALG students only)

Consider the B-trees of order M. Assume that we have the following result, which we will refer to as Lemma 1.

Lemma 1: The barest B-tree of height H contains $N = 2K^H - 1$ elements, where $K = \lceil \frac{M}{2} \rceil$.

Determine the upper bound for a B-tree of order 21 which has $1,000,000 = 10^6$ elements.

You must give an integer value as the upper bound of the B-tree.

You are not allowed to use the result given in the lecture regarding the upper bound for B-tree's height. Instead, you must work out the answer using Lemma 1 above.

Note: The total mark for Part 2 will be 100 for CSE2ALG students and 110 (100 + 10 for Task 4) for CSE5ALG students. The percentage of contribution to the final will be the same.

■