CSE2/CSE5ALG– Algorithms and Data Structures – 2020
Assignment – Part 2

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

Due Date: 29 May 2020, Friday, 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. **Late submission policy does NOT apply to this assignment.**

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 command -line arguments, e.g. `java WordMatch in1.txt out1.txt in2.txt out2.txt`.

Return of Assignment: The tutor is to mark your assignment with a marking sheet in a face-to-face pattern during the lab classes in **Week 12**. You will be notified with your mark immediately after marking. **If you have any doubt your mark, please raise it to the tutor before your lab class ends**. **Any post-lab inquiry will NOT be accepted.**

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 primary goal 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 considered the correctness only, for Part 2 we are concerned with the efficiency. 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

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 AT LEAST TWO text files from which the words are to be read to build the lexicon (The argument is to specify the input files).
2. The second is the name of a text file to which the words in the lexicon are to be written (The argument is to specify the file containing the words and the neighbors in the lexicon).
3. The third is the name of a text file that contains ONLY ONE matching pattern (The argument is to specify the file containing the matching pattern).
4. The fourth is the name of the text file that contains the result of the matching for the given pattern (The argument specifies the file containing the output).

For this version, the **efficiency** with which the program performs various operations is a major concern, i.e. the sooner the program performs (correctly), the better.

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 Assignment 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. Violation of this requirement may lead to a mark of ZERO.

Task 2 (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 23 which has $10,000,000 = 10^7$ 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 125 (100 for Task 1 and 25 for Task 2) for CSE5ALG students. The percentage of contribution to the final will be the same, i.e. 20%.

In your solution to Task 2, as well as in every Java class, you must include your student ID and name, and the subject code.

How to submit your solution to Task 2: Your solution should be a PDF file named Task2.pdf, and be submitted using the same command `submit ALG`, i.e. `submit ALG Task2.pdf`

Marking Scheme

Your assignment is to be marked according to a **marking sheet**. Please check the **example marking sheet** on the LMS subject page for details.