CSE2ALG / CSE5ALG: Algorithms and Data Structures

Assignment Part 2, 2021

Department of Computer Science and IT, La Trobe University

Assessment

This part of the assignment is worth **15%** of the final mark.

Deadline

25 May 2021, Tuesday, at 10:00 AM.

Delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. You must plan to allow for both scheduled and unscheduled downtime. **Late submission policy does NOT apply to this assessment**.

As this part of the assignment is worth **15% or more** of the final mark for this subject, please apply for **Special Consideration** if you need request an extension of time for submission due to any unforeseen circumstances that substantially affect your ability to complete it on time. More details can be found via the following link (https://www.latrobe.edu.au/students/admin/forms/special-consideration). **Once your application is approved by the University, the subject coordinator will grant you an extension of time for submission**.

Plagiarism

Plagiarism refers to submitting somebody else's work in a manner that gives the impression that the work is your own, which is an **academic misconduct**. The CSIT Department treats academic misconduct seriously. When it is detected, penalties will be strictly imposed.

Submission

You are supposed to submit ALL the files (including Java and PDF files) that are required for the tasks, from your **latcs8** account. Please make sure you have navigated to in the same directory where the files being submitted are. You must submit each file separately using the submit command **`submit ALG`**. For example, if the file is named *LexiconTester.java*, please use the following command for submission:

```
submit ALG LexiconTester.java
```

After submitting the files, you may 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 deadline. The previously submitted file will be replaced by the latest one. If you encounter any problem regarding submission, please email the subject coordinator for assistance.

Platform

While you are free to develop the code on any operating system, **your solution must compile and execute using `javac` and `java` commands on the latcs8 server.**

Return of Result

The subject coordinator will mark your submission with a marking sheet during the next face-to-face / online lab classes after the deadline (i.e., 25 and 28 May 2021). You will be notified with your mark soon after marking. If you have any doubt, please immediately raise it to the subject coordinator before the lab class ends. **Any post-lab inquiry will NOT be accepted**. If you cannot attend the lab classes, please email the subject coordinator for an alternative arrangement.

Restriction

You are **NOT** allowed to use **ANY** of the classes in the **Java Collections Frameworks** (JCF); the only exceptions are the *ArrayList* class and the *LinkedList* class.

More details about the JCF can be found via the following link (https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html).

Violating this restriction will lead to a mark of ZERO for this part of assignment.

Marking Criteria

You mark for this part of the assignment relates to the program's capability (i.e., whether being compiling and executing), the program's efficiency (i.e., how fast it produces the result), the code's quality (i.e., whether being reasonable and in good style), and the submission's accomplishment (i.e., whether satisfying the task requirements). Details can be found in the **example marking sheet**.

Description of Tasks

This part of the assignment is an extension of the part 1. Whereas for part 1 we considered the correctness only, we are concerned with the efficiency for this part.

Specifically, you are required to accomplish the tasks described below. Besides the description below, please refer to the instruction of part 1 for any information you need.

Task 1 Design and implement the program *LexiconTester.java*. This program will

1. Read **one** text file using the first command-line argument and construct a lexicon that contains words from the file. Suppose the input the file is *in.txt*, the command will execute as follows:
   ```
   java LexiconTester in.txt
   ```
2. Write the words, including spellings, frequencies and neighbors, from the lexicon to the text file *out.txt*. **The format must be the same as in the part 1 of the assignment.**
3. Include only **one** solution. It is entirely up to you which data structure(s) and/or sorting algorithm(s) will be used, as long as they are included in the subject; however, you must take the program's efficiency in consideration, as the test file is much larger in this part.
4. In principle, you program **MUST** accomplish the task **less than 60 seconds** on the latcs8 server. Otherwise, your mark will be severely impacted.

Task 2 Write a report *LexiconReport.pdf* in the PDF format. This report will

1. Describe which data structure(s) and/or sorting algorithm(s) are selected; if they are different from those in the part 1, describe the reason for your choice.
2. Describe the time complexities (i.e., the Big-Oh) of the involved data structure(s) and/or sorting algorithm(s) in the best, worst and average cases.
3. Describe any strategies or tricks from you to improve the program's efficiency in addition to the choice on data structure(s) and/or sorting algorithm(s).

Note Submitting additional Java files as auxiliaries is allowed, as long as these files are created originally by you and can compile on the latcs8 server.

CSE2ALG / CSE5ALG Assignment Part 2: Marking Sheet

Semester 1, 2021

		Mark	Max
1	Use any of the classes in the JCF, other than the *ArrayList* class and the *LinkedList* class.		-100
Task 1: LexiconTester.java (Max Mark: 85)			
Section 1: Blackbox Testing (Max Mark: 70, If the program takes more than **60 seconds** to accomplish the task, the mark given in Section 1 will **NOT** exceed **30**.)			
2	**`javac LexiconTester.java`** No error occurs when the program compiles. If other java files are submitted, then all files must compile.		10
3	**`java LexiconTester in.txt`** No error occurs when the program runs, and *out.txt* is created. (If the program takes more than **60 seconds** to accomplish the task, the rest criteria in Section 1 will be skipped.)		5
4	**The program accomplishes the task and terminates efficiently. Suppose the time usage is denoted by t (seconds),** • If t⩽10, the mark is 40; • If 10<t⩽20, the mark is 30; • If 20<t⩽30, the mark is 25; • If 30<t⩽40, the mark is 20; • If 40<t⩽50, the mark is 15; • If 50<t⩽60, the mark is 10; • If t>60, the mark is 0.		40
5	In *out.txt*, words and neighbours are all available and ascendingly sorted. Each exception results in deduction of 1 mark.		5
6	In *out.txt*, word frequencies are all available and calculated correctly. Each exception results in deduction of 1 mark.		5
7	In *out.txt*, neighbours are all available and selected correctly. Each exception results in deduction of 1 mark.		5
Section 2: Code Checking (Max Mark: 15)			
9	Code is reasonably written (i.e., it contains necessary classes and functions, indicating the author made efforts to accomplish).		8
10	Code is in a good programming style (i.e., it helps one to read and understand code and to avoid introducing errors).		7
	Task 2: LexiconReport.pdf (Max Mark: 15)		
11	The report describes the involved data structure(s) and/or algorithm(s) (including their time complexity).		8
12	The report describes strategies or tricks to improve the program's efficiency in addition to the involved data structure(s) and/or sorting algorithm(s).		7
Total			100