**Data Structures and Concurrency**

**Continuous Assessment 1**

**Due Monday 11th November 2024**

**You will be submitting this document when you have it completed.**

**N.B. Please keep your submission brief and to the point.**

**Before submitting, you should rename this document to include your TNumber in the document name – for example T00123456 CA 1.docx.**

Submit your completed document to Canvas->Data Structures and Concurrency. (Original will be used)

Student Name: Emmanuella G. Nyamekye

Submission Date: 12/11/2024

**SpellCheck Application (updated)**

For the SpellCheck application, various Collection classes can be used to store the dictionary of words – the words are read in from ‘words.txt’. For the CA, you may use another dictionary if you prefer – see ‘Sources of Data.docx’.

It counts the number of misspelt words found in the text you are spell checking (war-and-peace.txt). For the CA, please use **ANOTHER TEXT FILE** – other text files are available in ‘Sources of Data.docx’ or whatever source you prefer.

Use IntelliJ Profiler to generate % of time and actual time (in ms) for contains() method of your chosen Collection classes – code as given uses a LinkedList.

1. Complete the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Structure (Collection class) used to store dictionary** | **% of Time for contains() method** | **Time in ms for contains() method** | **Big Oh for contains() method**  **n – size of the dictionary** |
| LinkedList | 78 % | 1,228 ms | O(n) - 98342 |
| ArrayList | 71 % | 688 ms | O(n) - 98342 |
| HashSet | 0 % | 0 ms | O(1) - 98342 |
| TreeSet | 0 % | 0 ms | O(log n) - 98342 |
| LinkedHashSet | 0 % | 0 ms | O(1) - 98342 |

1. Obtained with Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz processor, Java Version 23.0.1, Windows 11 Home

When doing the above, try different Collection classes and see the different values you will get for the contains() method. The text file you use, to do the spell check on, must be large enough to allow you to discriminate between the different Data Structures (Collection classes) used to store the dictionary.

The Collection classes that you should use are the ones that we covered in:

OneDrive -> Data Structures and Concurrency 2023\_2024 -> “2. Java Collections Framework”

**that would be suitable to store the dictionary.**

When using ArrayList, you can do better than using the contains() method.

We will assume that the contents of the dictionary are in sorted order, so instead use the Collections binarySearch() method:

public static <T> int binarySearch([List](https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/util/List.html)<? extends [Comparable](https://docs.oracle.com/en/java/javase/18/docs/api/java.base/java/lang/Comparable.html)<? super T>> list, T key)

Change the SpellCheck.java code to handle this.

1. Explain why you would use binarySearch() method instead of contains() method for ArrayList?

**Time Complexity**: binarySearch() is significantly faster than contains() for large, **sorted** lists. binarySearch() has a time complexity of O(log n), while contains() has O(n), as it performs a linear search.  
**Sorted Data Requirement**: binarySearch() requires the ArrayList to be sorted. If the list is already sorted, binarySearch() is more efficient. If the list isn’t sorted, you’d need to sort it first, which might add overhead if the sorting step is repeated often.  
**Direct Access**: binarySearch() directly narrows down to the element's potential position through divide-and-conquer, making it better suited for quick lookups in a sorted list.

1. Specify the changes that you made to SpellCheck.java for ArrayList version

In the modified version (SpellCheckArrayList.java), the key change involves utilizing an ArrayList<String> for the dictionary to enable efficient spell checking using the binarySearch() method.

Big Oh values for the methods of Collection classes are available in the java api.

1. Give screenshots of the output from the Profiler showing the results that you have used to populate the 3rd column in the table above. You should have one screenshot for each row in the table and please give them in the same order. The screenshots should clearly show the Collection class used and the time for the **contains()** method or **binarySearch()**in the case of ArrayList.

LinkedList:

A computer screen with many colorful text

Description automatically generated

ArrayList:

A screen shot of a computer program

Description automatically generated

HashSet:

A screen shot of a computer program

Description automatically generated

TreeSet:

A screen shot of a computer program

Description automatically generated

LinkedHashSet:

A computer screen shot of a program code

Description automatically generated

1. File used for dictionary (if you used a different one): words\_alpha.txt

(b) Source of this file (give URL): <https://github.com/ian-IBCIRL/MTUSep24DSC/blob/main/08.%20Profiling/Spell%20Check%20code/words>

(c) Value of n (size of the dictionary): 98342

(d) File on which spell checking is done: The-Little-Prince.txt

(e) Source of this file (give URL):­­­­­­­­­­

https://nrzr.li/d3/y/1731444376/10000/e/lgrsfic/542000/30074a8efce0704891a10b263236d121.epub~/td1Dvve9etEisz71Qzuypw/The%20Little%20Prince%20--%20Saint-exupery%2C%20Antoine%20de%3B%20Testot-ferry%2C%20Irene%20--%200%20--%2030074a8efce0704891a10b263236d121%20--%20Anna%E2%80%99s%20Archive.epub

1. What Collection class would you recommend for the SpellCheck application?

HashSet

1. Explain your answer

HashSet is the ideal choice for the SpellCheck application due to its constant-time complexity for word lookups, lack of need for order, and efficient use of memory. These characteristics make it the most performant and resource-efficient option among the collections tested.

1. Any other suggestions you have for improving or extra ideas for this exercise

Explore parallel processing techniques to speed up the spell-checking process, especially for large documents.

Adding a spell suggestion feature could improve usability.

Expanding the application to support multiple languages would make it more versatile.

Developing a simple GUI for the application could make it more accessible to general users.

***Important Note: Plagiarism and Academic Dishonesty***

Please familiarize yourself with MTU-Kerry Campus Anti-plagiarism Policy and Procedures document – available at <http://www.ittralee.ie/en/InformationAbout/QualityAssurance/>

At this link you will see: A5 Assessment of Learners -> A5.2 Anti-Plagiarism Policy and Procedures

Please note that if the work you submit is not your own, a mark of 0 will be awarded.

**Appendix A**

**‘Declaration of Originality Form’**- MTU-Kerry.

|  |  |
| --- | --- |
| This form **must** be completed and signed and submitted with all assignments. | |
| Please complete the information below (using BLOCK CAPITALS). | |
| Name: Emmanuella G. Nyamekye  T Number: T00229159  Class Group: KCPIT\_C\_Y3  Assignment Title: Data Structures and Concurrency - Continuous Assessment 1 | |
| **Students are advised to inform themselves of the University Anti-Plagiarism Policy.** | |
| **I confirm that this assignment is my own work and that I have:** | |
| Familiarised myself with the University Anti-Plagiarism Policy | X |
| Used the University’s approved referencing style throughout | X |
| Clearly referenced, in both the text and the bibliography or references, all sources used in the work  Not made use of the work of any other student(s) past or present without acknowledgement. This includes any of my own work, that has been previously, or concurrently, submitted for assessment, either at this or any other educational institution | X  X |
| Not sought or used the services of any professional agencies to produce this work | X |
| In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations | X |
|  |  |
| DECLARATION:I am aware of and understand the University’s policy on plagiarism and I certify that this assignment is my own work, except where indicated by referencing, and that I have followed the good academic practices noted aboveSigned Emmanuella G. Nyamekye | |