**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.**

Student Name John O’Sullivan

Submission Date 07/11/2024

**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’.

It counts the number of misspelt words found in the text you are spell checking (war-and-peace.txt).

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 | 91% | 2705ms | O(n) |
| HashSet | 64% | 77ms | O(1) |
| Stack | 82% | 1140ms | O(n) |
| ArrayList | 88% | 1203ms | O(n) |
| ArrayDeque | 80% | 1270ms | O(1) |

1. Obtained with Intel(R) Core(TM) i7-14700 processor, 2.10 GHz, Java Version 23H2, Windows 11

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?

binarySearch() is faster than ArrayList due to the data being sorted

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

I changed the interface from Collection to List.

I changed the list variable from LinkedList to ArrayList.

I replaced the contains method with the binaryseach 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.
2. File used for dictionary (if you used a different one): [I used the provided dictionary]
3. Source of this file (give URL): [I used the provided dictionary]

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

(d) File on which spell checking is done: catextfile.txt

(e) Source of this file (give URL):­­­­­­­­­ https://www.gutenberg.org/

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

Hashset

1. Explain your answer Hashset would be the best choice as it the fastest of all the classes I tried
2. Any other suggestions you have for improving or extra ideas for this exercise A possible way to improve on this application would be adding a way to identify incorrect punctuation
3. References/Sources of information. Specify any sources you used.

<https://medium.com/@mckenziefiege/arrays-linked-lists-and-big-o-notation-486727b6259b>

<https://medium.com/@chakravartyutkarsh/understanding-why-hashset-provides-o-1-search-time-complexity-15cee2f96cec#:~:text=One%20of%20the%20most%20efficient,an%20element%20is%20incredibly%20fast>

<https://www.baeldung.com/cs/complexity-stack-queue-deque-set>

<https://sebastian-abarca.medium.com/stacks-queues-and-big-o-notation-4a555443260c>

<https://www.cs.cmu.edu/~mrmiller/15-121/Slides/09-BigO-ArrayList.pdf>

**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 \_JOHN O’SULLIVAN\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  T Number \_\_\_T00234079\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Class Group\_\_\_KCPGD-B-Y3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Assignment Title\_\_DATA STRUCTURES AND CONCURRENCY CA1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| **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 |  |
| Used the University’s approved referencing style throughout |  |
| 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 |    |
| Not sought or used the services of any professional agencies to produce this work |  |
| In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations |  |
|  |  |
| 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 \_\_\_\_\_\_\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |