**Lab 5\_3**

**SpellCheck Application**

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 (alice30.txt ). A larger text file war-and-peace.txt is also given.

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

Complete the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Structure 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% | 9928 | O(n) |
| ArrayList | 85% | 6900 | O(n) |
| HashSet | 2% | 12 | O(1) |
| TreeSet | 3% | 24 | O(log(n)) |
| ArrayDeque | 89% | 5838 | O(n) |
| LinkedHashSet | N/A | N/A | O(1) |

Obtained with Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz processor, 1.60 GHz, Java Version 23.0.1, Windows 11 Enterprise

Try different Collection classes and see the different values you will get for the contains() method. Use the larger text file if the smaller file give values that are too small.

What Collection class would you recommend for the SpellCheck application?

The Hash Set is the most efficient with regards to the contains() method. The context of the application is important, though, and the use cases.

Explain your answer

While also being the most time efficient, the HashSet, being a Set, stores each word only once, which matches the use of a dictionary.

It took 96ms to read in the dictionary into the ArrayDeque, while it was the second most inefficient using the contains() method. It took 154ms to read the dictionary into the HashSet. These are aspects to consider, but are not all that important as main() took 430ms with HashSet and 7050ms with ArrayDeque.