Text

Description automatically generated

The WordFrequency project is complete and functional. It reads a large text file, processes the content by removing specified punctuation, and counts the occurrences of each word using both TreeMap and HashMap implementations. It then prints the top 5 most frequently occurring words longer than 6 characters and records the execution time for both TreeMap and HashMap.

To test the program, I initially used a small text file to ensure that the word counting and sorting functions worked as expected. After confirming the correctness of the program, I switched to a large text file (over 200 Kbytes) downloaded from Project Gutenberg to measure the execution time of both TreeMap and HashMap implementations.

Timing Data:

HashMap has a slightly faster execution time than TreeMap. For example, in one run, the TreeMap execution time was 67 milliseconds, and the HashMap execution time was 24 milliseconds.(per screenshot)

Benefits and Drawbacks:

HashMap:

Benefits: Faster average-case performance (O(1) for insertion, retrieval, and deletion), uses less memory.

Drawbacks: Unordered data structure, slower worst-case performance (O(n) in case of hash collisions).

TreeMap:

Benefits: Sorted data structure, predictable O(log n) performance for insertion, retrieval, and deletion.

Drawbacks: Slower average-case performance than HashMap, uses more memory due to the balanced tree structure.

I learned the importance of choosing the right data structure for specific problems and comparing the performance of different data structures such as HashMap and TreeMap for a given task.

I liked that the project was pretty much up to us implementation wise and provided an opportunity to work with real-world, large text files and compare the performance of different data structures in a practical scenario.

What did I find confusing or would like to see done differently?

Removing punctuation marks was a bit confusing at first

What would I add or do differently with an extra hour or two?

Cant think of anything at the moment.