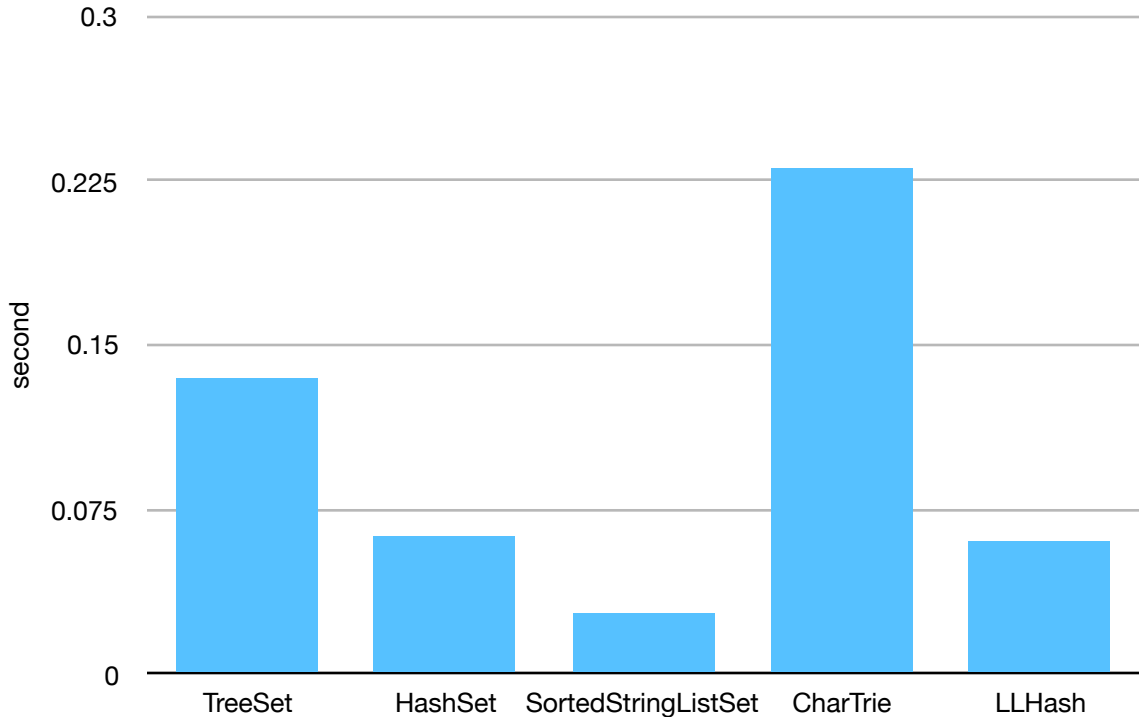
CSC212 SPELL CHECKING REPORT

Instructor: Prof. John. J. Foley

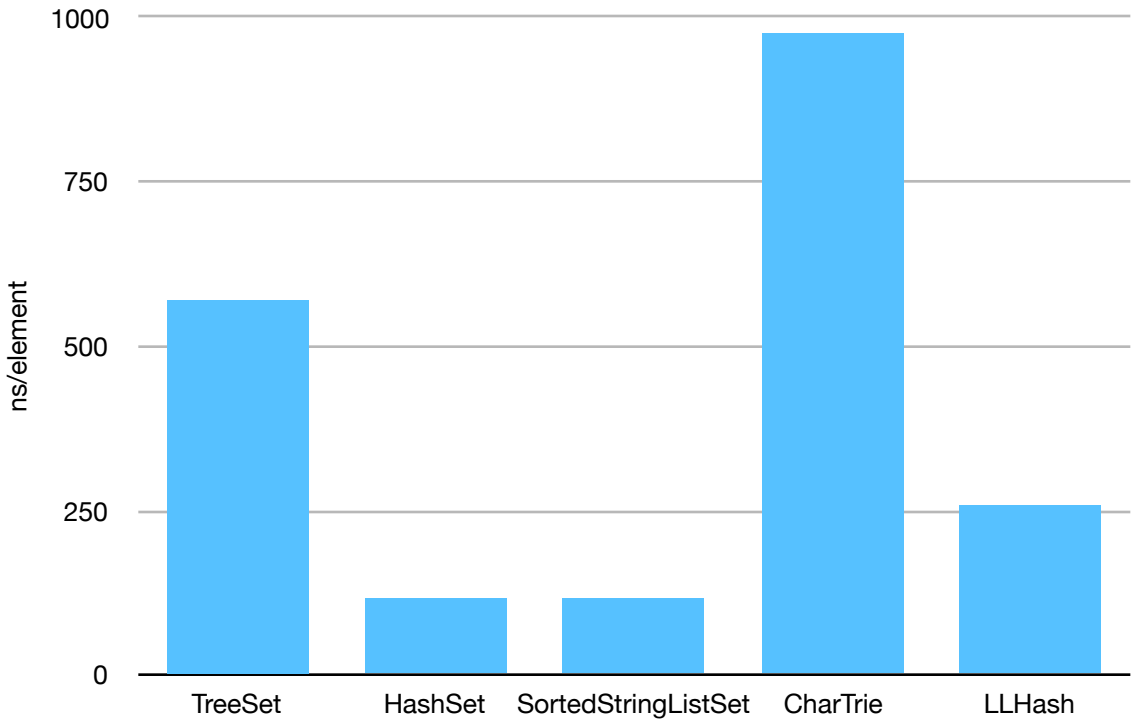
Author: Harry Wang

MEASURE CREATION (INSERTION) SPEED.

- How long does it take to fill each data structure?



- Plot insertion time per element for each of these data structures.



- Is there a timing difference between constructing `HashSet` and `TreeSet` with their input data or calling `add` in a for loop? If yes, use the words "balancing" and "resizing" to explain what's going on.

TreeSet input : 0.134442448 seconds
TreeSet calling add: 0.090030918 seconds
HashSet input: 0.027295044 seconds
HashSet calling add: 0.063091241 seconds

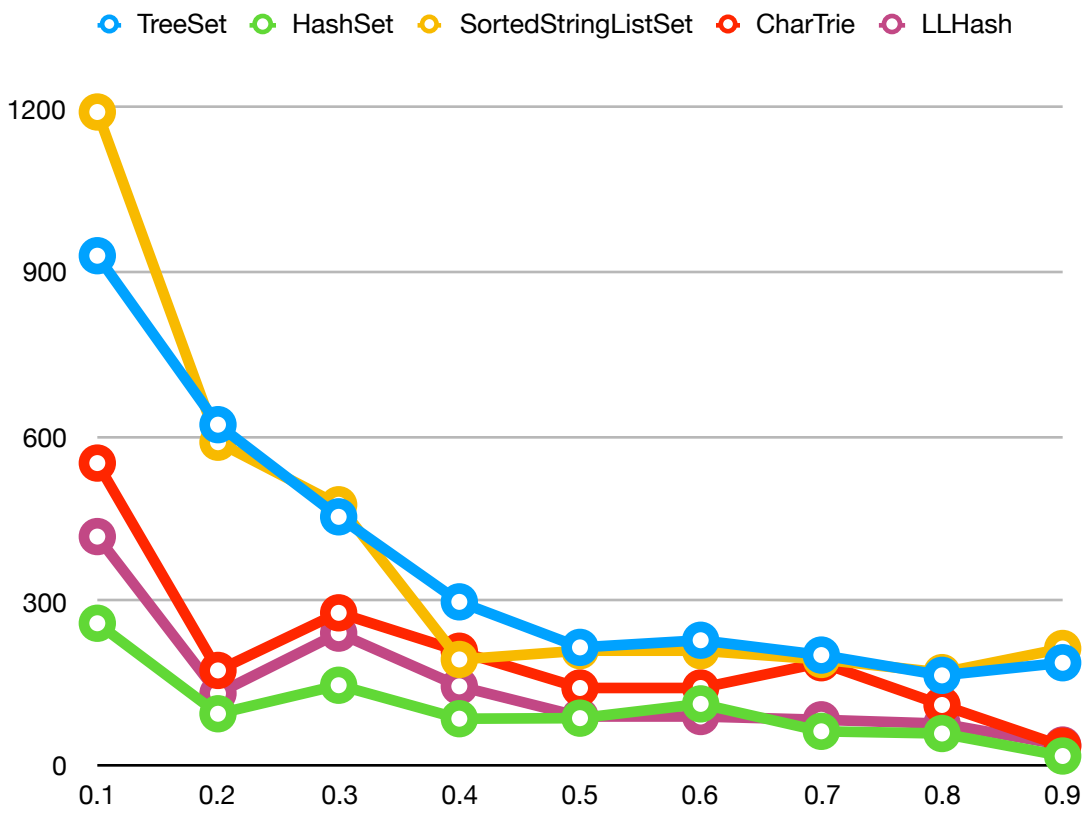
For TreeSet, it takes less time to calling add than input;
For HashSet, it takes less time to input than calling add.

TreeSet calling add: We add every element at the position at the tree.
TreeSet input data: Java need to sort the data and inserts in a pre-balanced order into TreeSet.

HashSet calling add: We need to calculate the hash value so the time would be longer than input. Besides, we need to resize the array.
HashSet input data: This is the good one.

Plot Query Speed

- Construct a dataset that has Strings that are both in and not in the dictionary.
- For full credit, devise a method to inject some percentage of hits and misses. Create a line plot as the percentage of hits goes from (0.0 to 1.0) in steps of 0.1, where each line is a different data structure.



SPELL-CHECK A PROJECT GUTENBERG BOOK

- What is the ratio that's “mis-spelled”?

9165 in 78735
0.116

- Are the query speeds the same over real-world data?

The query speed seems slower than real-world data

- What are some of the words that are “mis-spelled”?

http
pglaf
org
cont
pg
http
www
gutenberg

etc.
(some weird word, internet word, also name)

Conclusion: Which Data Structure do you recommend?

Overall, based on the 'experiment result' and the graphs, I recommend Hash for spell checking. Each element has it value, and we can use the hash value to get the data. (we don't need to go through the list, do traversals, etc.)

REFLECTION

What I learned:

- Learned how to use System.nanoTime(), Seeing how would time complexity effect the program.
- Learned how bucket work.
- You can't get both optimized space complexity and time complexity (relatively)
- Learned more about binary search (The google webpage is helpful)
- Some regex
- (Can we do multi-threads?)