CSE 332 Project 2 Write up

* Note: The last 3 questions require you to write code, collect data, and produce graphs of your results together with relatively long answers. Do not wait until the last minute to start this write up!

1. **Who is in your group?**

2. **What assistance did you receive on this project? Include anyone or anything *except* your partner, the course staff, and the printed textbook.**

3. **a) How long did the project take?**
   b) Which parts were most difficult?
   c) How could the project be better?

4. **What "above and beyond" projects did you implement? What was interesting or difficult about them? Describe in detail how you implemented them.**

5. **a) How did you design your JUnit tests & what properties did you test?**
   b) What properties did you NOT test?
   c) What boundary cases did you consider?

6. **a) Why does the iterator for Binary Search Tree need to use a stack data structure?**
   b) If you were to write an iterator specifically for the AVL Tree, how could you guarantee that no resizing of the stack occurs after iteration has begun (which may require changing the interface for `GStack`)?

7. **If `DataCounter`'s iterator returned elements in "most-frequent words first" order, you would not need to sort before printing. For each `DataCounter` (BST, AVL, MoveToFrontList, HashTable), explain how you would write such an iterator and what its big-O running time would be.**

8. **For your Hashtable to be <u>CORRECT</u> (not necessarily *efficient*), what must be true about the arguments to the constructor?**

9. **Conduct experiments to determine which `DataCounter` implementation (BST, AVL, MoveToFrontList, HashTable) & Sorting implementation (insertionSort, heapSort, OtherSort) is the fastest for large input texts.**

   a) Describe your experimental setup: 1) Inputs used, 2) How you collected timing information, 3) Any details that would be needed to replicate your experiments.

   b) Experimental Results (Your graph and table of results & Interpretation).
   You need to conduct experiments for all possible combinations, 4 DataCounter X 3 Sorting algorithms = 12 experiments. Don't forget to give title and label axis for graphs and state which combination is the best. Does the result match your expectation? If not, why?

   c) Are there (perhaps contrived) texts that would produce a different answer, especially considering how MoveToFrontList works?

   d) Does changing your hashing function affect your results?
   (Provide graph/table & interpretation)
   Conduct 6 experiments using HashTable (3 Sorting Algorithms X 2 Hashing functions = 6)
   Does the result match your expectation? If not, why?

10. **Conduct experiments to determine whether it is faster to use your *O(n log k)* approach to finding the top *k* most-frequent words or the simple *O(n log n)* approach (using the fastest sort you have available).**

    a) Produce a graph showing the time for the two approaches for various values of *k* (where *k* ranges from 1 to n).
    If you measure runtime including the time it takes to print, you should print same number of words (i.e. print top-k words for both n long k and n log n algorithm) to account for time it takes to print.
    Be sure to give your interpretation of the result. Does the result match your expectation? If not, why?

    b) How could you modify your implementation to take advantage of your experimental conclusion in a)?

11. **Using `Correlator`, does your experimentation suggest that Bacon wrote Shakespeare's plays? We do not need a fancy statistical analysis. This question is intended to be fun and simple. Give a 1-2 paragraph explanation.**

12. **<u>If you worked with a partner</u>:**
    a) Describe the process you used for developing and testing your code. If you divided it, describe that. If you did everything together, describe the actual process used (eg. how long you talked about what, what order you wrote and tested, and how long it took).
    b) Describe each group member's contributions/responsibilities in the project.
    c) Describe at least one good thing and one bad thing about the process of working together.

Appendix

Place anything that you want to add here.