**CSE 373 Homework 4 Write up**

1. **Who is in your group (Give name, UW NetID & student number of each person)?** Emnet Gossaye 1221300

Vincent Joe 1470087

1. **a) How did you design your tests & what properties did you test?**

**b) What boundary cases did you consider?**

1. **Conduct an experiment to determine which DataCounter implementation (HashTable\_SC, HashTable\_OA) is better for large input texts.**

**a) Describe your experimental setup:**

**1) Inputs used**

For this experiment we used six text files that increased in size by a factor of 1000.

**2) How you collected timing information**

Using the sample code provided in the spec we calculated the average run time of creating each HashTable implementation and counting all the words in the given text file. To get the average run time we timed each input size nine times and ignored the first three runs to encounter the JVM warmup.

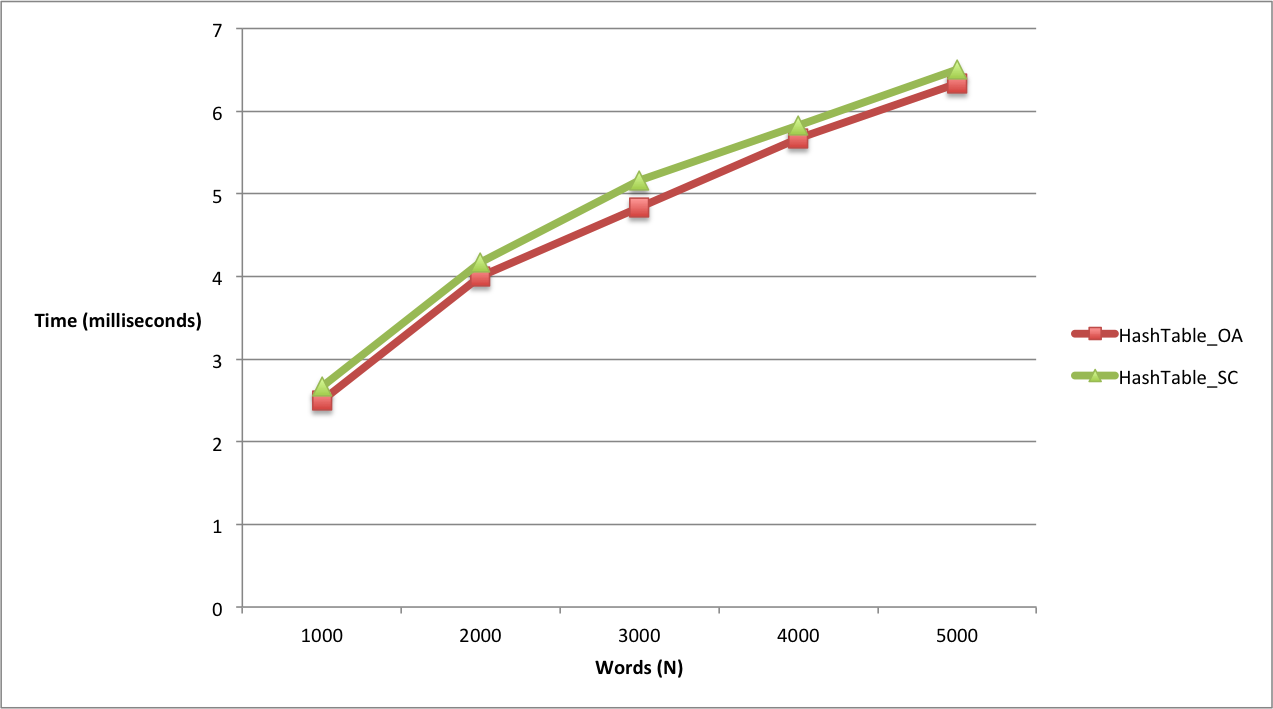
**3) Any details that would be needed to replicate your experiments**

When timing our WordCount.java in the write up experiment package we commented out the method call

to print out each word and its corresponding count.

**b) Experimental Results (Place your graphs and tables of results here).**

|  |  |  |
| --- | --- | --- |
| words | HashTable\_OA | HashTable\_SC |
| 1000 | 2.5 | 2.665 |
| 2000 | 4 | 4.166 |
| 3000 | 4.833 | 5.166 |
| 4000 | 5.666 | 5.833 |
| 5000 | 6.333 | 6.5 |
| 6000 | 7.333 | 7.833 |

****

**c) Interpretation of Experimental Results**

**1) What did you expect about the results and why?**

We expected our results to be linear because we were performing insert, which takes a constant amount of time O(1) for N number of words. Hence, asymptotically, it will take O(n) for the whole operation.

**2) Did your results agree with your expectations? Why or Why not?**

Our results were more or less in agreement with our expectations because as our input size increased the run time did so proportionally. The graph of test data isn’t exactly a straight line because we were computing average run times for each input size and incremented those sizes by a factor of 1000. If we used smaller intervals of N and tested even larger input sizes, the graph data would look more linear .

**3) According to your experiment, which Hashtable implementation, separate chaining or open addressing, is better?** For the input sizes we used in our experiment there wasn’t a significant difference in the average runtime of the two HashTable implementations. However our tests indicated that on average separate chaining performs slightly better than open addressing. We suspect this difference can be attributed to the fact that handling collisions in separate chaining is simpler than doing linear probing on the table array.

1. **Conduct experiments to determine if changing the hash function affects the runtime of your HashTable.**

**a) Brief description of your hash functions**

**b) Experimental Results (Place your graphs and tables of results here).**

**Experiment with at least 2 hash functions (2 Hashing functions = 2 experiments depending on how you measured the runtime)**

**Don’t forget to give each graph a title and label the axes.**

**c) Interpretation (Your expectations and why? Did it match your results? If not, why?)**

1. **Using Correlator, does your experimentation suggest that Bacon wrote Shakespeare's plays?**

**Show at least one (you can experiment with more texts if you want) correlation value for each of:**

**a) Shakespeare's work compared to Shakespeare's work**

‘Hamlet’ vs. ‘Romeo and Juliet’ = 2.620455785497558E-4

‘Romeo and Juliet’ vs. ‘Macbeth’ = 2.6413594784180204E-4

**b) Bacon's work compared to Bacon's work**

‘The New Atlantis’ vs ‘Essays’ = 3.9225828623651066E-4

‘Essays’ vs ‘The Advancement of Learning’ = 1.3442623411191485E-4

‘The New Atlantis’ vs ‘The Advancement of Learning’ = 3.806036695857482E-4

**c) Shakespeare's work compared to Bacon's work**

‘Hamlet’ vs ‘The New Atlantis’ = 5.657273669233966E-4

‘Romeo and Juliet’ vs ‘The Advancement of Learning’ = 4.451265609183206E-4

‘Macbeth’ vs ‘Essays’ = 4.5705225319640216E-4

**According to the results of your experiments, did Bacon write Shakespeare's plays?**

Since a small variance indicates that the data points tend to be close to each other and a larger variance indicates that they are more spread out from each other, we can say that Bacon did not write Shakespeare’s plays. When comparing the works of Shakespeare against his own works, the resulting variance is approximately 0.00026. Comparing Francis Bacon’s works to his own, results in variances that are approximately 0.0003. If Sir Francis Bacon were to have written Shakespeare’s plays we would expect then that the variance between his works and Shakespeare’s to be close to these values. Instead the variance between works attributed to Shakespeare and works attributed to Sir Francis Bacon is approximately 0.0005.

1. **Include a description of how your project goes "above and beyond" the basic requirements (if it does).**
2. **If you worked with a partner:**

**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).**

We began working on the project together early on in the weekend and started off by talking about how we would implement our StringComparator and hash function. Then, we moved on to implementing and testing our open addressing hashtable first and making sure we got the right word counts for both text files. After finishing HashTable\_OA we implemented separate chaining which was slightly more difficult because

**b) Describe each group member's contributions/responsibilities in the project.**

**Emnet**

Correlator, separate chaining, testing open addressing, comparator, timing experiment

**Vincent**

Open addressing, testing/fixing separate chaining, string hasher, comparator, debugging

**c) Describe at least one good thing and one bad thing about the process of working together.**

One of the benefits that we found of working with another person was that being able to discuss the code in greater detail with someone helped us find bugs and fix them quicker than being stuck with one bug for much longer than necessary. The only drawback of working with another person is that you both have to work at the same pace and keep up otherwise one cannot go on if they need the other to finish what they’re working on first. This of course can be a good thing however because it encourages both people to stay on task and be efficient.

1. **a) Which parts of the project were most difficult?**

One difficult part of the project was implementing the iterator for our hash tables. Neither of us had done that before so it was a bit tricky to figure out initially. The other challenging aspect of this assignment as we got further along in the assignment, there was a lot of code we had to sift through so it made it harder to pinpoint what was causing bugs and how to fix them.

**b) How could the project be better?**

We enjoyed working on this project and learned a lot one suggestion we have to improve this project would not have it be due the week after the midterm. Even though we had two weeks from the day it was assigned, we couldn’t give it our full attention/priority until after the exam. Otherwise it was a very interesting and engaging assignment.

**Appendix**

**Place anything else that you want to add here.**