Joseph Kim

Professor Eduard Dragut

CIS 2168 – Section 008

Due 11 December 2017

Assignment 5 Report

“When you are submitting Assignment 5, please make sure to submit a one-page report that includes (1) mention the quadratic algorithm that you implemented, (2) a description of the two CompareTo methods, and (3) a table where you show the times of the 4 settings described above, i.e., the time it takes to run java sort based on length and lexicographical order and the same for your quadratic sorting algorithm.”

(1) This program implements hash tables with **bubble sorting**. Bubble sort is the most basic way to sort a collection. It works by sequentially going through your array and comparing two values at a time, swapping them if necessary. It then repeats the process until no swaps are required. Bubble sort requires a lot of steps in order to sort even a short collection. It takes a lot of iterations and comparison to small values to their final destination. I used bubble sort in order to sort the hashtags in alphabetical order. By using the compareLength method, I was able to swap two variables with the bubble sort algorithm. I also tried to implement **insertion sorting,** but I was unable to properly implement it, but it works. Insertion sorting is when you compare the key with the whole array and insert the new element in the place after that.

(2) The **compareLength** method – compares length of the first length of the string and sees whether or not it’s shorter or longer than the previous hashtag. If the next hashtag’s length is shorter than the previous hashtag, it’ll switch to have the shorter hashtag in front and the older hashtag after. If the next hashtag is longer, it’ll leave the previous hashtag in place and place the new hashtag after the old hashtag.

The **compareLexicographical** method – compares sequentially the hashtags that have the same position in both lengths against each other until one element is not equivalent to the other. The result of comparing these first non-matching hashtags is the result of the lexicographical comparison. If both hashtags compare equal until one of them ends, the shorter sequence is lexicographically less than the longer one.

(3) a table where you show the times of the 4 settings described above

|  |  |
| --- | --- |
| **Type of Quadratic Algorithm** | **Time based on 1000 hashtags** |
| Bubble Sort | 1.55 seconds |
| Insertion Sort | 0.80 seconds |
| compareLength | 4.6 seconds |
| compareLexicographical | 3.6 seconds |