Name: Siyu Zhang

Course: CS0445 Java Data Structure

Instructor: John Ramirez

Time: 2/20/2018

Report about the Operation Comparisons among String, StringBuilder and MyStringBuilder

In this assignment, three operations: append, delete, and insert are timed under the implementation of three different classes, including String, StringBuilder and MyStringBuilder.

Here are the three comparison graphs of each operation (more specific data is in the Graph Comparisons.xlsx). The X-axis is the value of N (number of operations) and the Y-axis is the average time per operation (amortized time).

After the comparison of append() operation, String classes show the slowest runtime for append() method, because it is immutable and append() operation actually requires to create a new String object every time. Although the data for StringBuilder and MyStringBuilder are similar, StringBuilder is still considered as a “close” winner for the append() method.

As for the delete(0,1) method, MyStringBuilder would be seen as a clear winner. Delete(0,1) operation is implemented by substring method in String class, requiring a large amount of time to create new String objects in each operation, while delete(0,1) is the best case for MyStringBuilder class which only requires assigning the variable firstC to its next node.

On the other hand, in the insert() method, StringBuilder is the best implementation and it is a clear winner. In this operation, MyStringBuilder needs to go through the first half of the linked list every time in order to insert the certain data into the “middle” of it, costing a large amount of time.

Overall, since StringBuilder is outstanding in two operations, it is concluded to be best from a run-time point of view, which costs the least amount of time. Moreover, as the operation delete(0,1) is a kind of cheat method call for MyStringBuilder (the best case for this class), StringBuilder still shows a relatively short run-time in this operation. Therefore, StringBuilder would be best from a run-time point of view.