// Assignment2 WriteUp

Eshan Parmar 4282240

When it came to the implementation for append the Predefined String builder was faster when it came to appending 10,000 and 20,000 items at the end, but as the number of items increased the MyStringBuilder caught up and the two implementations had close times. At 10,000 appends the StringBuilder had an avg time of 56.56ns per append while the MyStringBuilder had a avg time of 100.65 ns. On the other hand, at 160,000 appends the StringBuilder had a avg time of 28.91 ns while the MyStringBuilder had an avg time of 28.71 ns. So, at the higher number of appends MyStringBuilder and the Predefined StringBuilder become much closer.

The best implementation for delete was the MyStringBuilder implementation. MyStringBuilder was the clear winner in this case, at 10,000 appends the avg time was 72.55 ns while the second closest was the Predefined StringBuilder with an avg time of 159.49 ns. The time difference noticeably increases as the amount of deletions increases. At 160,000 deletions MyStringBuilder had an avg time of 68.30 ns while StringBuilder had an avg time of 1040.81 ns which is a sizable difference.

With the insert method the MyStringBuilder again had the most efficient implementation. At 10,000 operations the MyStringBuilder had an avg time of 63.15 ns while StringBuilder, which was the second closest, had an average of 137.00 ns. At 160,000 operations the gap in the time increased, for MyStringBuilder the avg time was 130.54 ns while StringBuilder had an avg time of 669.44 ns which again is a large gap. Here the clear winner was the MyStringBuilder implementation.

Overall, the MyStringBuilder seemed to be the most efficient of the implementations, at a higher amount of operations the time difference is definitely noticeable. When dealing with only a small number of operations the time for StringBuilder and MyStringBuilder were close enough where it would be fine to use either one.

