Introductions

The aim of this report is to measure the execution of string concatenation, String builder appending, insertion sort and merge sort by calculated how many concatenations in one second for String and String builder with one character and 80 characters. Moreover, the runtime for insertion sort and merge sort.

**Exercise 2: Concatenation and String Builder:**

To test the executions time, I created a String with one character and a string builder that appends one character, and a string with 80 characters and a string builder that appends 80 characters. Start time created to give the current time in milliseconds and end time created to give the current time after finishing the concatenation process inside while loop for one second.

The executions time calculated is the difference between end time and start time.

Table 1. String with one character

|  |  |  |  |
| --- | --- | --- | --- |
| ExNo | Time (Milliseconds) | Concatenation | Length |
| 1 | 1000 | 46583 | 46583 |
| 2 | 1000 | 52294 | 52294 |
| 3 | 1000 | 61896 | 61896 |
| 4 | 1000 | 61599 | 61599 |
| 5 | 1000 | 61146 | 61146 |
| Average | 1000 | 56704 | 56704 |

Table 2. String with 80 characters

|  |  |  |  |
| --- | --- | --- | --- |
| ExNo | Time (Milliseconds) | Concatenation | Length |
| 1 | 1000 | 3497 | 279760 |
| 2 | 1000 | 4250 | 340000 |
| 3 | 1000 | 4873 | 389840 |
| 4 | 1000 | 4837 | 386960 |
| 5 | 1000 | 4837 | 386960 |
| Average | 1000 | 4459 | 356704 |

Table 3. StringBuilder with 1 character

|  |  |  |  |
| --- | --- | --- | --- |
| ExNo | Time (Milliseconds) | Concatenation | Length |
| 1 | 1000 | 24325655 | 24325655 |
| 2 | 1000 | 24789029 | 24789029 |
| 3 | 1000 | 24547372 | 24547372 |
| 4 | 1000 | 24459398 | 24459398 |
| 5 | 1000 | 24907917 | 24907917 |
| Average | 1000 | 24605875 | 24605875 |

Table 4. StringBuilder with 80 characters

|  |  |  |  |
| --- | --- | --- | --- |
| ExNo | Time (Milliseconds) | Concatenation | Length |
| 1 | 1372 | 171966480 | 171966480 |
| 2 | 1133 | 171966480 | 171966480 |
| 3 | 1097 | 171966480 | 171966480 |
| 4 | 1037 | 85983280 | 85983280 |
| 5 | 1000 | 160083760 | 160083760 |
| Average | 1128 | 152393296 | 152393296 |

**Result:**

The experiment suggests that StringBuilder is faster than String concatenation because, String concatenation with (+) operator makes a copy for each concatenation which requires memory and time while StringBuilder adds the new String at the end position and only make a copy in some cases such as during resize or if inserting element in the middle.

**Exercise 3: Sorting Algorithms:**

The aim of this experiment is to see how many Integers and String can be sorted by insertion sort algorithm and merge sort algorithm in one second by creating a random array of integers and string with deferent length.

Table 5. Integer Insertion Sort

|  |  |  |
| --- | --- | --- |
| ExNo | Time (Milliseconds) | Length |
| 1 | 1004 | 95045 |
| 2 | 1018 | 95044 |
| 3 | 1025 | 95042 |
| 4 | 1024 | 95041 |
| 5 | 1003 | 95041 |
| Average | 1015 | 95043 |

Table 6. String Insertion Sort

|  |  |  |
| --- | --- | --- |
| ExNo | Time (Milliseconds) | Length |
| 1 | 1001 | 11506 |
| 2 | 999 | 11497 |
| 3 | 1000 | 11501 |
| 4 | 1001 | 11515 |
| 5 | 1003 | 11549 |
| Average | 1001 | 11514 |

Table 7. Integer Merge Sort

|  |  |  |
| --- | --- | --- |
| ExNo | Time (Milliseconds) | Length |
| 1 | 1000 | 3529168 |
| 2 | 1000 | 3528061 |
| 3 | 1000 | 3530053 |
| 4 | 1000 | 3528582 |
| 5 | 1000 | 3532141 |
| Average | **1000** | **3529601** |

Table 8. String Merge Sort

|  |  |  |
| --- | --- | --- |
| ExNo | Time (Milliseconds) | Length |
| 1 | 1000 | 733118 |
| 2 | 1000 | 730013 |
| 3 | 1000 | 730112 |
| 4 | 1000 | 730010 |
| 5 | 1000 | 730120 |
| Average | **1000** | **3529601** |