Robel Tadele

COSC-2203

String Sorting Lab Report

Introduction

What is Sorting?

Sorting is the act of arranging similar data or objects into groups, types or in a certain sequence. In java we use sorting to arrange data in a certain sequence I.e. Ascending, non-descending order etc....

In this String Sorting lab, I will be implementing and testing the speed of the seven types of sorts discussed in class namely: Bubble, Selection, Insertion, Shell, Heap, Merge and Quick Sort. First, I will be reading in 45, 000 words from a text file named “undictionary.txt”; and then I will sort the words through an Index because sorting strings by themselves is demanding. Then I will be using System.nanoTime() to time how long it would take to sort words in 5,000-word increments using the different sorting algorithms.

1. **Bubble Sort**

Number of Words Read in	Time elapsed in Seconds
5,000	0.206691
10,000	0.8840965
15,000	1.9865198
20,000	3.9224497
25,000	6.7788388
30,000	11.9781795
35,000	21.4246637
40,000	37.7296952
45,000	55.6603829

2. Selection Sort

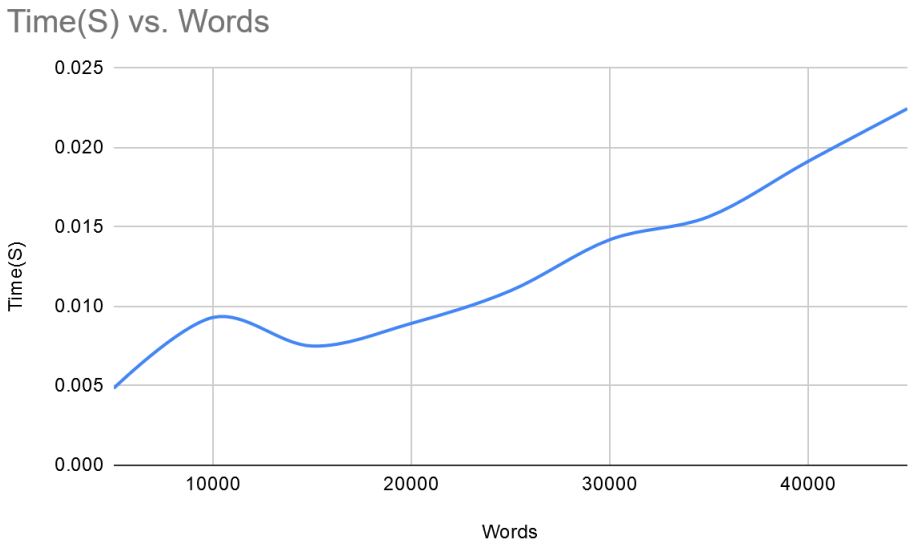
Number of Words Read in	Time elapsed in Seconds
5000	0.1703817
10,000	0.722423
15,000	1.6862096
20,000	3.2070432
25,000	5.5163195
30,000	8.9637847
35,000	13.0280313
40,000	19.0302463
45,000	26.4540951

3. Insertion Sort

Number of Words Read in	Time elapsed in Seconds
5000	0.0476415
10,000	0.150043
15,000	0.2197562
20,000	0.3173972
25,000	0.4861691
30,000	0.6914247
35,000	0.9103502
40,000	1.3045368
45,000	1.6406607

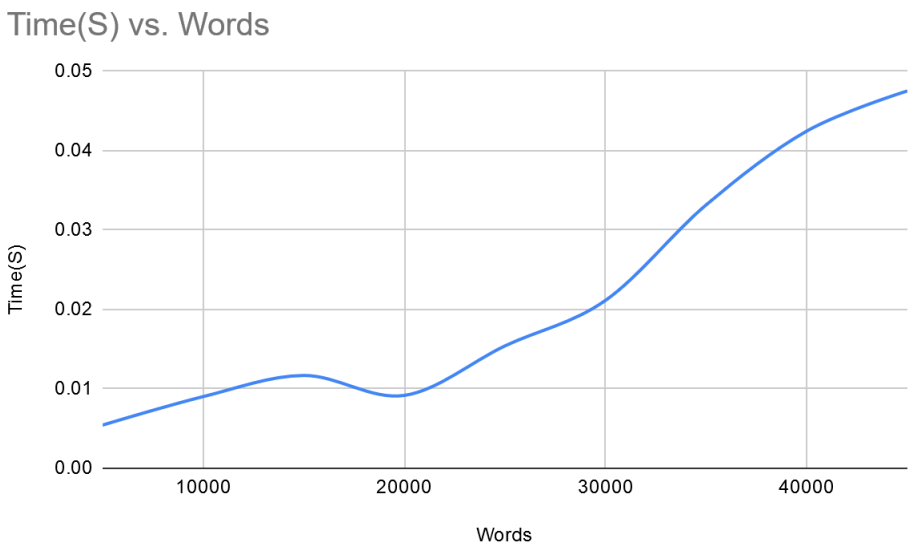
4. Heap Sort

Number of Words Read in	Time elapsed in Seconds
5000	0.0048566
10,000	0.0093105
15,000	0.0075088
20,000	0.0089368
25,000	0.0109941
30,000	0.0142046
35,000	0.0156598
40,000	0.0191389
45,000	0.022453



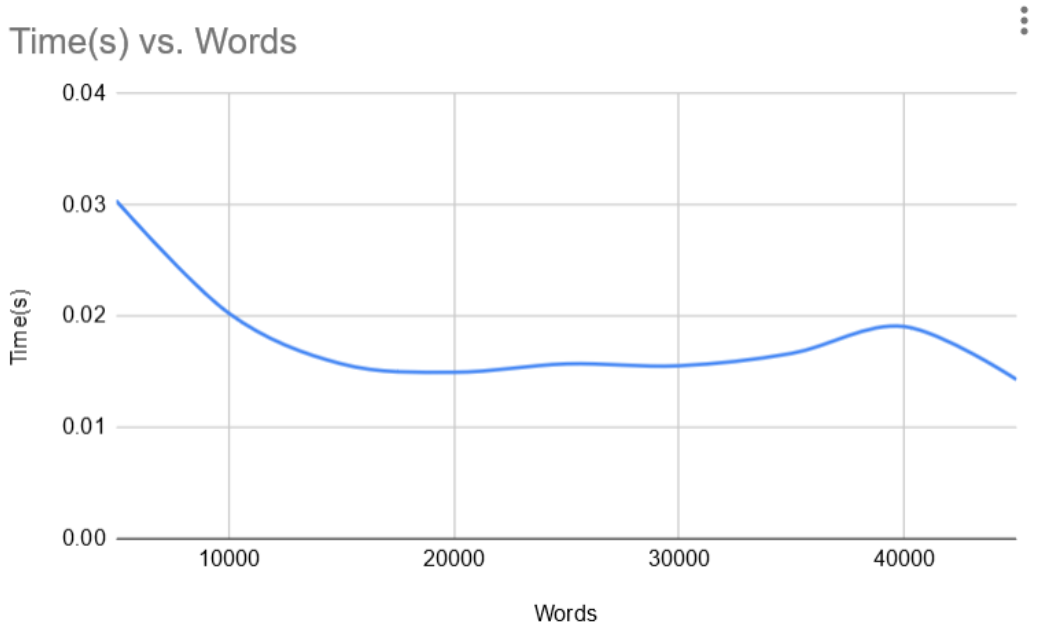
5. Shell Sort

Number of Words Read in	Time elapsed in Seconds
5000	0.0054433
10,000	0.0090267
15,000	0.0116899
20,000	0.0091778
25,000	0.0154124
30,000	0.0211517
35,000	0.0331756
40,000	0.0424672
45,000	0.0475213



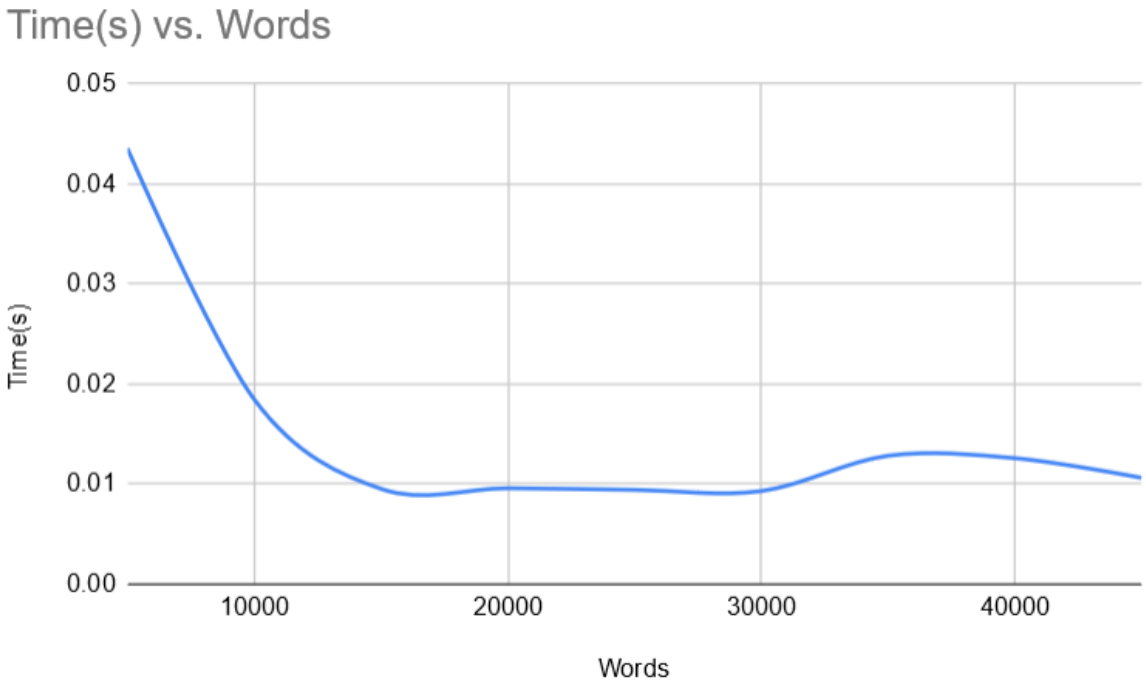
6. Merge Sort

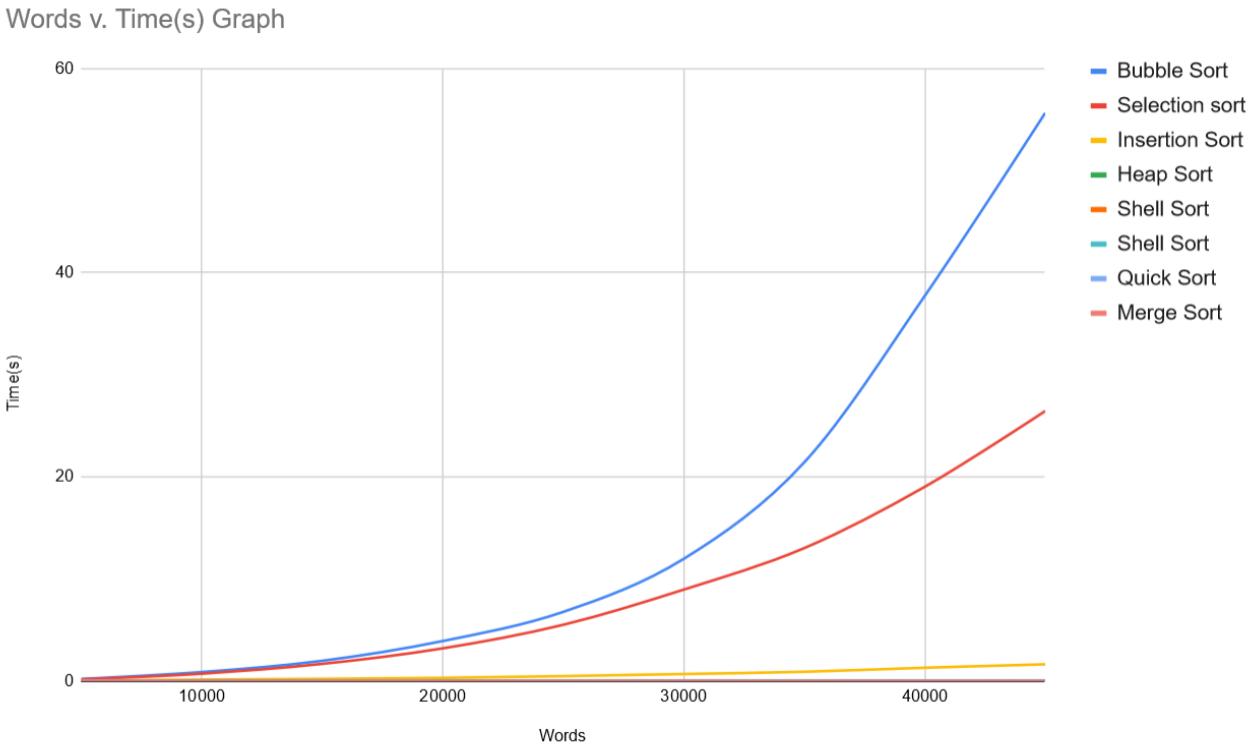
Number of Words Read in	Time elapsed in Seconds
5000	0.0303816
10,000	0.0202706
15,000	0.0157209
20,000	0.0149654
25,000	0.0157201
30,000	0.0155487
35,000	0.0166594
40,000	0.0190784
45,000	0.0143061



7. Quick Sort

Number of Words Read in	Time elapsed in Seconds
5000	0.0435154
10,000	0.0184566
15,000	0.0095333
20,000	0.0095985
25,000	0.0094375
30,000	0.0093167
35,000	0.012846
40,000	0.0126013
45,000	0.0106346





According to the line graph displayed above, it's evident that the runtime of quadratic sorts quickly increases as the amount of data sets increase. Though quadratic sorting algorithms are easier to build and maintain from scratch, they're generally not recommended for sorting larger amounts of data since they are slow and require lots of resource. Though quadratic sorts are not efficient for large data sets, their simplicity makes them ideal for sorting smaller ones. We can also notice the sharp decline in the runtime as we move to more efficient sorting algorithms like the Heap and Shell sort. Finally, we can see that the Merge sort and Quick sort are the fastest sorting algorithms.

The results in the line graphs above aren't the exact runtime of the sorting algorithms because other tasks like calling methods is involved when sorting, and this generally increase the runtime recorded.