This program compares the sort times for three basic sorting algorithms. The Bubble Sort, the Heap Sort and the Merge Sort. All three sorts are using the same set of integers. The runs use sets of 100,000, 200,000 and 1,000,000. I decided to try it with a million, the time is not that long. (The Bubble sort took less than 47 minutes to sort one million integers.) This does not seem like a long time to run a computer program. I had estimated the time should be less than fifty minutes based on earlier runs with fewer items and fitting a quadratic time. (If it had calculated to hours, I would have run it over night but it was easy to start it and go do something else that didn't involve the comp.)

I notice that the Heap and Merge sorts are close together in time for all three data sets. The difference as a percentage gets smaller as the size of the dataset increases. This is an expected characteristic of the nlogn time.

I am a little concerned about the use of a pseudo-random number generator with this many items. Experience has shown that sequences do repeat in a non-random way. Without more information about how the Random class works it is hard to say. I ran into this issue doing "Monte Carlo" simulations in Fortran. (I have done Monte-Carlo in Java but only with relatively small runs so the issue did not arise.)

I used both the Heap and Merge sorts because I'd never used either one before. I found the Heap sort especially to be interesting. Once I realized that I could use the ReHeap method to do the initial build it all fell into place. I thought about building ReHeap recursively but that seemed like a lot of extra work.

Note that both the Bubble sort class and the Heap sort class contain their own "swap" methods. I wanted the sort classes to be self contained. There are no swaps, per se, in the Merge sort.

SortTime.main({""});

The size is: 100000

Bubble Sort

Total time to sort = 26759 milliseconds // Just under 27 seconds

Array of integers sorted: true

Merge Sort

Total time to sort = 34 milliseconds

Array of integers sorted: true

Heap Sort

Total time to sort = 38 milliseconds

Array of integers sorted: true

SortTime.main({"" });

The size is: 200000

Bubble Sort

Total time to sort = 111448 milliseconds // One minute and 51 seconds.

Array of integers sorted: true

Merge Sort

Total time to sort = 60 milliseconds

Array of integers sorted: true

Heap Sort

Total time to sort = 58 milliseconds

Array of integers sorted: true

SortTime.main({"" });

The size is: 1000000

Bubble Sort

Total time to sort = 2817188 milliseconds // Just under 47 minutes. 47 minutes is 2,820,000 msec.

Array of integers sorted: true

Merge Sort

Total time to sort = 238 milliseconds

Array of integers sorted: true

Heap Sort

Total time to sort = 233 milliseconds

Array of integers sorted: true