How quick is quick sort?

CIS1154 Lab 8 Run Sheet

1. Please study the quick sort implementation and the test code in this repository https://github.com/sbunivedu/cs2-recursion. Create a driver class to measure the actual performance of quick sort. You can study "BubbleSortTest.java" and "QuickSortTest.java" to learn how to call the "sort" methods.

Increase the size of your arrays by a factor of 10 repeatedly and record the actual runtime in the following table:

Quick Sort		Bubble Sort	
array size	runtime	array size	runtime
100	0m0.070s	100	0m0.080s
1000	0m0.076s	1000	0m0.085s
10000	0m0.081s	10000	0m0.175s
100000	0m0.149s	100000	0m6.907s
1000000	0m0.229s	1000000	Program timed out
10000000	0m0.667s	10000000	
100000000	0m5.376s	100000000	

Note: to obtain your program runtime precede your command with "time" as shown in the following example; record the time labeled "user".

```
time java QuickSortTest
real    0m0.651s
user    0m0.092s
sys     0m0.044s
```

2. Repeat the same experiment with bubble sort (see implementation in the same repo). What are the factors we need to consider to make this a fair comparison?

The work for a bubble sort implementation will always take longer than a quick sort implementation because bubble sort requires comparing every two numbers each time it traverses through the array. Quick sort performs more like a binary search algorithm. Therefore, it will always take longer to perform a bubble sort than a quick sort.

For there to be a fair comparison, it may help to use smaller array sizes to experiment with as eventually the program will time out if the array is too large for either sorting method. (reference: pg. 230-232 bubble sort and quick sort algorithms explanations)