Sorting Hypotheses

While using my Surface Pro 3, Intel i5 2.50 GHz, I plan to test the three popular O(n2) sorting functions Selection Sort, Insertions Sort, and Bubble Sort, against the built-in java O(nlogn) Quicksort. I will create random integer data using the following arrays sizes: 10, 100, 1000, 10000, and 100000. The data will be random integers representing the entire integer range from 0 - 1000.  I will test each array size three times to accumulate a total of 15 different times for each test, then will analyze the time to see if caching, heap or JIT are causing any huge difference in time. I will drop the first time of all test subjects because those are the biggest discrepancies.

I Predict that the popular Selections, Insertion, and bubble will lag way behind the Java built in Quicksort. Out of the three popular ones, I believe that Bubble will be the fastest, followed by Selection, and finally the Insertions Sort will come in last.

Results

With my results I have come to the conclusion that my prediction was almost correct. Obviously the built in java Quick sort will beat the times of the others and it will them by a huge margin. As expected, the bubble sort came in 1st place out of the three for being the fastest. The Insertion Sort was where my prediction was wrong, it had beat Selection Sort in almost all the array sizes. Finally, the Selection Sort came in last place and also having some pretty high numbers, almost doubling the times from array size 100,000 for Insertion Sort. Overall, the race looked close between the O(n^2) methods while the O(nlogn) swept the victory by a long shot.