Table 1: Running Times (Measured in seconds)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sort Method | 1N =  1000  keys | 2N =  2000  keys | 4N =  4000  keys | 8N =  8000  keys | 16N =  16000  keys | 32N =  32000 |
| Selection Sort | 0.714 | 5.038 | 19.387 | 71.882 | 253.515 | 1017.53 |
| Quicksort | 0.027 | 0.045 | 0.145 | 0.65 | 0.45 | 1.388 |
| STL Sort | 0.019 | 0.051 | 0.216 | 0.649 | 1.021 | 1.522 |
| Process Exited | 7.758 | 10.38 | 25.99 | 77.05 | 258.1 | 1024 |

**Questions:**

**1) How well do your measured times conform with the order-of-magnitude estimates for selection sort and quick sort? Note that you may need to run more tests to answer this question effectively.**

The complexity estimates for selection sort and quicksort do seem to match my times measured.

While selection sort greatly increases in run-time, the plotted line of points increases like a quadratic function. This matches the order-of-magnitude estimate of both the average and worst case time complexity for selection sort which is O (n^2).

The quicksort, however, more closely matches the average case of O (n log n). The growth rate for this sort is much slower when compared to that of selection sort. When the number of keys is 16000, the quicksort run-time is still under 1 second. And with twice as many keys, 32000, the quicksort takes just over 1 second while the selection sort takes over 1000 seconds.

Comparing the run-times of the quicksort and selection sort, as well as the plotted graph, the results show that quicksort will execute extremely faster than the selection sort.

In this analysis, both the selection sort and quicksort run-times conform closely to their respective order-of-magnitude estimates.

**2) Using the code in sort.cpp and your measured execution times, develop a Big-O estimate of the execution time of the STL sort routine. Briefly explain your answer.**

After analysis of both the table of run-times and plotted graph, the STL sort can be said to resemble the quicksort in nature.

Both run-times are very similar for values of N keys under 16000. While there is some divergence after the number of keys doubles from this point, a noticeable trend between the two sorting methods can be observed. Furthermore, on the plotted graph, these two sorts are almost entirely indistinguishable when compared to the run-time of the selection sort.

Therefore, an order-of-magnitude estimate for the STL sort would much more closely match that of the quicksort rather than the selection sort. This estimate can be characterized as a case of O (n log n).

It should be noted that the O (n log n) estimate for the quicksort is only the average case. And the worst case for the quicksort is O (n^2). However, in this demonstration, the time complexity of the quicksort and STL sort more closely resembled O (n log n).