ASher Bernardi

CSCI245

Dr. Devin Pohly

February 01, 2017

**Project 1 Write Up**

This project consists of testing many different sorting algorithms to analyze their efficiencies and determine whether the number of comparisons that the algorithm has to make is a good indicator of its speed. The algorithms were bubble sort, selection sort, insertion sort, shell sort, merge sort, and quicksort (implementing the Lomuto partition scheme). For the experiment, each algorithm sorted 14 different, randomly generated arrays of integers of various lengths from 10 to 70,000 elements. The number of comparisons made between values in the array and the time it took each algorithm to finish sorting were measured. Data chart is provided on the next page.

In all cases, merge sort performed the most quickly and bubble sort the most slowly. The algorithms could be divided into two groups: the slow sorts (bubble, selection, and insertion), which take several seconds to solve arrays of 50,000 or more elements, and the fast sorts (shell, merge, quick), which can perform the same task in less than a tenth of a second.

The following graph shows the ratio of comparisons to runtime for each array and for each sorting algorithm. The data for arrays smaller than 10,000 elements is not included because the time it takes for each algorithm to sort the arrays is so minimal as to be irrelevant.

The graph shows that for any given algorithm, the number of comparisons is directly correlated to its runtime, because the lines are relatively horizontal. However, it would not be possible to know the runtime of an algorithm only by knowing the number of comparisons it made. For example, on an array of 25,000 elements, selection sort performed more than twice as many comparisons as did insertion sort (312,487,500 vs. 154,001,729 comparisons), but it completed the sorting in less time (595 vs. 620 milliseconds). We conclude that the lines that are higher up on the above graph represent algorithms that are more efficient with their comparisons, in that they perform more comparisons per millisecond than the algorithms found at the bottom of the graph.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | bubble | selection | insertion | shell | merge | quick |
| for 10: | | | | | | |
| Comparisons: | 90 | 45 | 45 | 32 | 25 | 29 |
| Run Times (milliseconds): | 0 | 0 | 0 | 0 | 0 | 0 |
| for 50: | | | | | | |
| Comparisons: | 2205 | 1225 | 550 | 319 | 217 | 338 |
| Run Times: | 0 | 0 | 0 | 0 | 0 | 0 |
| for 100: | | | | | | |
| Comparisons: | 8316 | 4950 | 2475 | 875 | 545 | 582 |
| Run Times: | 0 | 0 | 0 | 0 | 0 | 0 |
| for 150: | | | | | | |
| Comparisons: | 18625 | 11175 | 5782 | 1589 | 904 | 963 |
| Run Times: | 0 | 0 | 0 | 0 | 0 | 0 |
| for 500: | | | | | | |
| Comparisons: | 232534 | 124750 | 63111 | 8248 | 3852 | 5088 |
| Run Times: | 1 | 0 | 0 | 1 | 0 | 0 |
| for 10000: | | | | | | |
| Comparisons: | 98920107 | 49995000 | 24952644 | 330479 | 120063 | 578696 |
| Run Times: | 314 | 95 | 100 | 2 | 1 | 2 |
| for 15000: | | | | | | |
| Comparisons: | 222450169 | 112492500 | 55262523 | 524889 | 188949 | 1266390 |
| Run Times: | 736 | 215 | 223 | 3 | 2 | 3 |
| for 20000: | | | | | | |
| Comparisons: | 390360481 | 199990000 | 98673730 | 728740 | 260327 | 2154597 |
| Run Times: | 1336 | 381 | 397 | 4 | 3 | 5 |
| for 25000: | | | | | | |
| Comparisons: | 618700251 | 312487500 | 154001729 | 975709 | 333358 | 3333776 |
| Run Times: | 2135 | 595 | 620 | 6 | 3 | 8 |
| for 30000: | | | | | | |
| Comparisons: | 885870470 | 449985000 | 222882558 | 2812477 | 407683 | 4743460 |
| Run Times: | 3093 | 858 | 899 | 14 | 4 | 11 |
| for 35000: | | | | | | |
| Comparisons: | 1210965400 | 612482500 | 300872665 | 2105031 | 483470 | 6437703 |
| Run Times: | 4276 | 1167 | 1215 | 11 | 5 | 15 |
| for 40000: | | | | | | |
| Comparisons: | 1581480462 | 799980000 | 396683188 | 1456191 | 560771 | 8344814 |
| Run Times: | 5621 | 1523 | 1604 | 8 | 6 | 19 |
| for 45000: | | | | | | |
| Comparisons: | 1998000599 | 1012477500 | 500067654 | 1647509 | 638452 | 10462935 |
| Run Times: | 7158 | 1930 | 2027 | 9 | 6 | 24 |
| for 50000: | | | | | | |
| Comparisons: | 2468500628 | 1249975000 | 620380805 | 1815343 | 716623 | 12875166 |
| Run Times: | 8876 | 2378 | 2506 | 9 | 7 | 30 |
| for 70000: | | | | | | |
| Comparisons: | 4846310765 | 2449964999 | 1218141163 | 2718846 | 1036437 | 25193288 |
| Run Times: | 17549 | 4670 | 4927 | 14 | 10 | 57 |