**Program 4: Sorting**

**Introduction**  
This report will be comparing the following sorting algorithms: BubbleSort, InsertionSort, MergeSort, IterativeMergeSort, QuickSort and ShellSort. The purpose of this paper is to determine the relative performance of each algorithm and their BigO complexity. This will be achieved by creating graphs depicting the varying speeds of each algorithm in milliseconds, as the input vectors increase in size.

**Sorting Data for all Algorithms**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sorting Algorithms** | **100** | **1,000** | **10,000** | **25,000** | **50,000** | **75,000** | **100,000** | **200,000** |
| **BubbleSort** | **42** | **4,025** | **391,835** | **2,514,565** | **10,881,272** | **24,828,597** | **44,132,980** | **172,714,690** |
| **InsertionSort** | **8** | **663** | **65,335** | **428,108** | **1,742,130** | **4,009,342** | **7,554,151** | **29,567,174** |
| **MergeSort** | **13** | **118** | **1,421** | **4,151** | **8,116** | **14,322** | **18,644** | **38,279** |
| **IterativeMergeSort** | **9** | **110** | **1,276** | **3,394** | **7,720** | **13,141** | **16,869** | **35,379** |
| **QuickSort** | **9** | **86** | **969** | **2,660** | **7,022** | **9,652** | **12,312** | **25,148** |
| **ShellSort** | **9** | **108** | **1,605** | **5,302** | **11,430** | **17,279** | **24,440** | **56,831** |

**Graph 1: Sorting Algorithms Comparison**

**Graph 2: Bubble Sort vs. Insertion Sort**

**Graph 3: MergeSort vs IterativeMergeSort vs QuickSort vs Shellsort**

**Table 3: Sorting Algorithms’ Big O Complexity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sorting Algorithms** | **Best Case** | **Average Case** | **Worst Case** |
| BubbleSort | O(n) | O(n²) | O(n²) |
| InsertionSort | O(n) | O(n²) | O(n²) |
| MergeSort | O(n log n) | O(n log n) | O(n log n) |
| IterativeMergeSort | O(n log n) | O(n log n) | O(n log n) |
| QuickSort | O(n log n) | O(n log n) | O(n²) |
| ShellSort | O(n log n) | O() | O(n²) |

**Conclusion**

Graph 1 demonstrates the BubbleSort and InsertionSort are the slowest algorithms by far, with BubbleSort being the slowest algorithm overall. Their sorting times have increased exponentially as the vectors have increased in size.   
  
Graph 2 depicts how the difference in sorting times between BubbleSort and InsertionSort becomes more drastic as the vectors become larger. The difference starts off negligible, but BubbleSort quickly becomes extremely slow.   
  
Graph 3 shows the differences between the last four algorithms. QuickSort is the fastest by far. It is followed by IterativeMergeSort which is then followed closely by MergeSort. IterativeMergeSort is faster due to only needing one copy vector, but it is only a marginal difference. Last is ShellSort, which is still drastically faster than Bubble and InsertionSort, but the slowest of the bunch with the remaining functions.  
  
From this data, we can conclude that QuickSort is by far the fastest sorting algorithm. Additionally, we can conclude that BubbleSort and InsertionSort are the slowest of the bunch and therefore should only be used in select cases. Additionally, we can conclude that the remaining algorithms are relatively close in speed, with ShellSort tending to be a bit slower, and IterativeMergeSort and MergeSort being neck and neck.