**Assignment 5a**

**Problem**: Empirical evaluation of data structures and algorithms

Comparison of sorting algorithms: Merge sort vs Quick sort

vs some O(n^2) algorithm for sorting (e.g., Insertion sort, Bubble sort).

Problem size: 8 million or larger, RT: 3 secs or larger.

**Sorting Algorithms to compare**:

1. Insertion Sort
2. Merge Sort
3. Quick Sort

**Input to the algorithms:**

1. Randomly generated elements
2. Sorted elements
3. Sorted array in reversed order
4. Partially sorted array (by swapping 3rd and 4th position)

**Test Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Size | Input | Insertion Sort | Merge Sort | Quick Sort |
| 50000 | Random | 2569 | 15 | 13 |
| Sorted | 1 | 4 | 4 |
| Reversed Sorted | 3419 | 0 | 0 |
| Partially Sorted | 0 | 0 | 0 |
| 95000 | Random | 6413 | 25 | 21 |
| Sorted | 1 | 10 | 10 |
| Reversed Sorted | 12342 | 7 | 12 |
| Partially Sorted | 1 | 6 | 9 |
| 100000 | Random | 7110 | 22 | 44 |
| Sorted | 1 | 6 | 1 |
| Reversed Sorted | 14022 | 10 | 0 |
| Partially Sorted | 1 | 6 | 4 |

**Conclusion:**

1. Merge Sort gives better performance while sorting random integer array over Quick Sort and Insertion Sort.
2. Performance of Quick sort improves for increasing size of input data than that of Merge Sort.
3. For already sorted array insertion sort behavior improves a lot.
4. Insertion sort takes a lot of time when sorting reversely sorted array.

**Code Reference:**

**1**. http://www.java2novice.com/java-sorting-algorithms/merge-sort

**2**. http://www.java2novice.com/java-sorting-algorithms/quick-sort

**3**. http://www.java2novice.com/java-interview-programs/insertion-sort