**Analysis of Sorting Algorithms using Survey Data**

The Quick, Merge and Bubble sorting algorithms used in implementation of table sorting with different sizes of survey data, the execution time is recorded and provided in the following table and 2D chart.

|  |  |  |  |
| --- | --- | --- | --- |
| No. of Records | Time Taken for Quick Sort | Time Taken for Merge Sort | Time Taken for Bubble sort |
| 100 | 164 | 591 | 268 |
| 500 | 446 | 1648 | 2238 |
| 1000 | 933 | 3884 | 8176 |
| 2000 | 1707 | 9112 | 27895 |
| 3000 | 3558 | 11231 | 74235 |

**Based on execution time recorded in table and chart, it is clear that quick sort is fastest algorithm and bubble sort is slowest algorithm.**

**When the size of data is 100 (or small) all sorting algorithms performing same, but if the size of data grows the bubble sort is much slower (takes more time) than quick and merge sort.**

**Quick Sort:**

Quicksort is a divide and conquer algorithm. Picks an element, called a pivot (here the first element), from the array. Partitions the array such that all elements which are less than or equal to the pivot are moved to the left side of pivot and all elements which are greater than the pivot are moved to right side of pivot. After this partitioning, the pivot is sorted. Recursively the same process is applied on the left side of pivot and right side of the pivot until each sub list contains one element (means sorted).

As this algorithm is linearly operating on adjacent elements it is cache favoured one and thus it is fastest algorithm. The worst case can be eliminated if we can pick pivot randomly.

The best-case time complexity is O(nlogn).

**Merge Sort:**

Merge Sort is a Divide and Conquer algorithm. In merge sort, first it divides the list into half and then then these sub lists are further divided into half until each sub list contains only one element (Means sorted). Then these sub lists are repeatedly merged with Merge algorithm to produce the sorted list.

The drawback of this algorithms is it uses temporary array in the process of merge.

The best-case time complexity is O(nlogn). As per above analysis this slower than quick sort and faster than bubble sort.

**Bubble Sort**:

A bubble sort, also called a sinking sort or exchange sort, it performs n-1 iterations, in each iteration it compares all adjacent pairs and swaps them if necessary, causing the elements to "bubble" up toward their proper position. The process continues until no swaps are necessary or all n-1 iteration are exhausted.

The time complexity of the algorithm is O(n2).

The above analysis shows that the bubbles sort is slowest sorting algorithm.