Report on IT Workshop Project

Siddharth Goyal 15UCS140 — Kanak Singhal 15UCS057

I. Introduction

The project outlines the analysis of the various algorithms used to sort arrays. This is of particular importance is handling large amount of data. An efficient algorithm gives better results for the organiser. Use of LaTeX and GNUplot has been extensive in the project to facilitate it.

II. Algorithm analyzed

The following algorithms have been analyzed:-

- Bubble Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Selection Sort

III. Method of analysis

To analyze the varios algorthms we have generated a random array of numbers. The arrays are varied in size to compute the working of algorithms in various environments. The same array is sorted using all the algorithms and the number of comparisions are calculated. The algorithm with the most comparisons is the least efficient. Time taken by different algorithms to sort the given data is also considered. This is indeed a good measure of performance, especially since its independent of the underlying hardware design.

IV. Analysis of algorithms

A. Bubble Sort

Bubble sort algorithm, also referred to as sinking sort algorithm, is a simple sorting algorithm. The algorithm sorts the given array by fixing from the first unit and then comparing the array in adjacent pairs from the last, it swaps the pair if it is in wrong order. Then the algorithm repeats itself for every element in the array except the last one. This continuous process sorts all the elemets in the desired order. The algorithm, which is a comparison sort, is named for the way smaller elements "bubble" to the top of the list.
Best case:

$$n^2$$

Average case:

$$n^2$$

Worst case:

$$n$$

B. Insertion Sort

Insertion sort algorithm is a simple sorting algorithm. The algorithm works by fixing on the second element from the start and then comparing the preceeding elements. The preceeding elements are sorted by comparing them in pairs and are swapped if necessary. Otherwise the point is replaced by the current temp value in the loop, This repeats for all the succeeding elements in the array.
Best case:

$$n^2$$

Average case:

$$n^2$$

Worst case:

$$n$$

C. Quick Sort

Quicksort also called as partion-exchange sort is an efficient sorting algorithm, serving as a systematic method for placing the elements of an array in order. Analysis of quicksort shows that, on average, the algorithm takes O(n log n) comparisons to sort n items. In the worst case, it makes O(n2) comparisons, though this behavior is rare. The worst case of the quicksort algorithm only occurs when the pivot choosen is the biggest or the smallest element in the array.
Best case:

$$nlogn$$

Average case:

$$nlogn$$

Worst case:

$$logn$$

D. Selection sort

Selection sort is a simple sorting algorithm. The process itself is very simple. The first element is taken as base, relative to which the array is scanned. When an element smaller than base comes up that element is made as base and the scanning continues. Then it is swapped with the first element. Now the same process is repeated from the second element till the (n-1) element. Although it is very inefficient in conventional sense due to high number of comparisions involved, the seletion sort is very simple to understand and use.
Best case:

$$n^2$$

Average case:

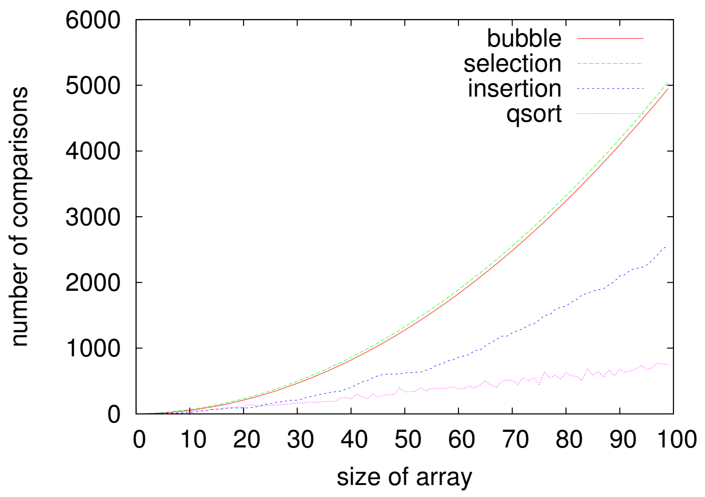
$$n^2$$

Worst case:

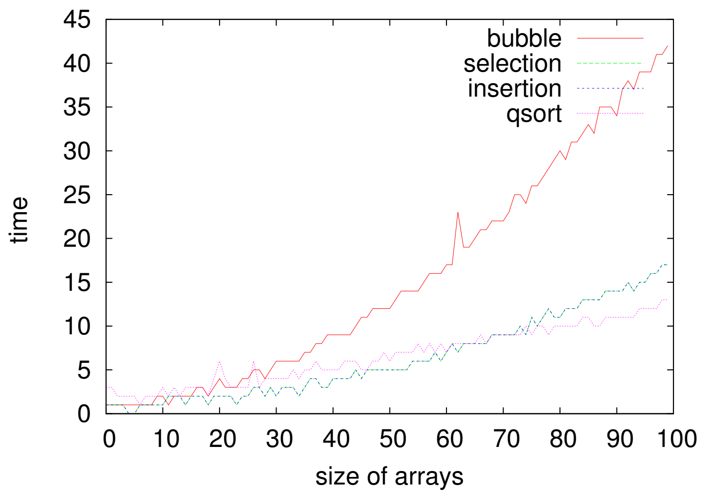
$$n^2$$

V. GNU Plots

GNU Plot of Number of comparisions and size of arrays



GNU Plot of execution of time and size of arrays



VI. Conclusion

The analysis of the above algorithms gives us that divide-and-conquer algorithms like Merge sort and Quick sort are faster and more efficient. Taking the random data into account we can conclude the following:-

- Selection sort is the worst algorithm to sort data even more than Bubble sort by the GNU plots.
- Quicksort is the best algorithms to sort an array when the pivot is taken randomly for a completely random data set.