ITWS Project

Karan Agarwal, Namish Narayan & Umang Govil

I. Introduction

We are given a program that should compute the total number of comparisons made between the elements of the array for each type of sorting including Bubble Sort, Merge Sort, Selection Sort, Insertion Sort, Quick Sort until it outputs the sorted array.We also have to draw the graphs using gnuplot and okular to compare them.

II. Preparing The Data

A. Time taken by your program

- Usually, the efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations (space complexity).
- Linux Command "time ./a.out" gives us the amount of time taken by our program to run.
- Therefore, to compare between different sorting algorithms same input must be used.

B. Number of comparisons depend on array A, its size n and the algorithm used

- For Bubble Sort: Starting from first iteration to find the largest element, no. of comaparisons will be n-1, this process will continue till the n-1th comparison is made between the n-1th and the nth element.The total number of comparisons, therefore, is (n) + (n - 1)...(2) + (1). $C_b sort(A) = n(n+1)/2$
- For Selection Sort: The maximum no. of comparisons in Selection Sort is same as that of bubble sort and calculated the same way. $C_s sort(A) = n(n+1)/2$
- For Merge Sort: The maximum no. of comparisons in Merge Sort is calculated by induction. For n=2^k and solving through induction gives no. of comparisons to be nlogn. $C_m sort(A) = nlogn$
- For Insertion Sort: It is a simple sorting algorithm that builds the final sorted array (or list) one item at a time. $C_i sort(A) = n(n-1)/2$
- For Quick Sort: Quicksort is a sorting algorithm that uses the divide and conquer strategy. This algorithm finds the element called pivot , that partitions the array into two halves in such a way that the elements in the left sub array are less then and the elements in the right subarrays are are greater then the pivot element. And we apply the same algorithm on left subarray and right subarray separately (recursive). $C_q sort(A) = nlogn$

C. Algorithm For Selection Sort

SELECTION SORTfor$i \leftarrow 0$ to lengthdo for $j \leftarrow i + s$ to nif a[i] >a[j] store index of in tdoneexchange a[t] with a[i]

III. Graph Plotting Using GNUPLOT

Gnuplot is a command-line program that can generate two- and three-dimensional plots of functions, data, and data fits. It is frequently used for publication-quality graphics as well as education. The program runs on all major computers and operating systems (GNU/Linux, Unix, Microsoft Windows, Mac OS X, and others). It is a program with a fairly long history, dating back to 1986. Gnuplot can produce output directly on screen, or in many formats of graphics files, including Portable Network Graphics (PNG), Encapsulated PostScript (EPS), Scalable Vector Graphics (SVG), JPEG and many others. It is also capable of producing LaTeX code that can be included directly in LaTeX documents, making use of LaTeX's fonts and powerful formula notation abilities. The program can be used both interactively and in batch mode using scripts.

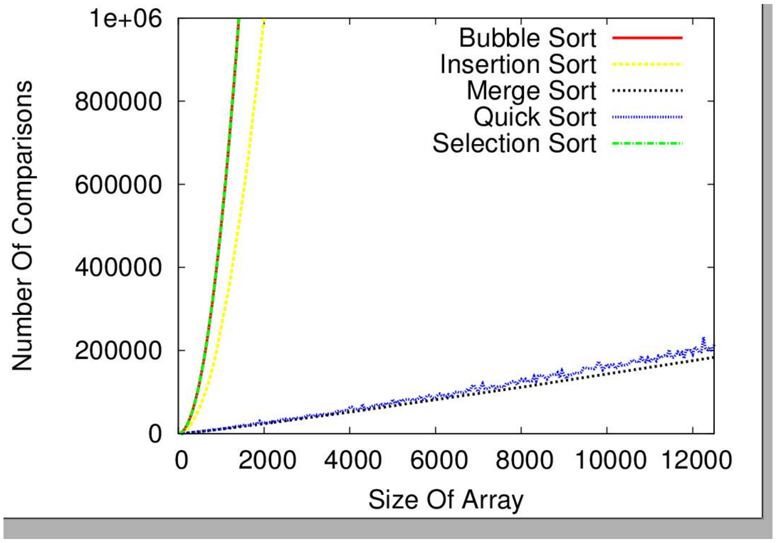

Fig. 1. GNUPlot

IV. Conclusion

According to the graph we are getting, we draw the conclusion that the merge sort and quick sort are the fastest sorting algorithms we have which give the result in n(log n) time. They are followed by insertion sort with n(n-1)/2 time. Then comes the selection sort and bubble sort with n(n+1)/2 time.