Deborah Rempala

CSCI 201

Assignment 1: Sorting data using algorithms

Due Monday, January 20th, 2014

For this experiment we used random data to sort. For random data I would recommend using the Comb sort because it is much faster than the other types of sort. The reason why the comb sort is the fastest is because the larger items are at the end. The slowest sort is the bubble sort because it must go through all the elements in the array. Without sorting the array first this takes the most amount of time. Finally, the times of the insertion sort, selection sort and bubble sort seemed to double while the comb sort only doubled after the array was at 60,000 items.

In this experiment we used data that was already sorted. The fasted sorting algorithm was the insertion sort because it starts at the first position in the array and compares it to the second. Since the data is already it just has to go through the entire ray. The second fastest was to comb sort because it had to separate the data before it compares it. The slowest algorithm was the bubble sort because it has to put the data in a vertical column. For the comb sort and the insertions the time remains the same while the bubble sort and selection sort seemed to double.

In this experiment we used data that was sorted in the reverse order. Once again the comb sort was the fastest at sorting the data. The insertion sort was the slowest because it had to start at the first position and keep rearranging the data until it was in the right order. The selection sort and the bubble sort pretty much had the same time because they both look for the misplaced element first. Once again the times seemed to double as the array grew larger. Finally, when the array had only 20,000 items the times were pretty much together because the list was so small to begin with.

For this experiment we used data that was almost sorted. This means about 80% of the data was in order. When you only had 20,000 items in the array all the data was easily sorted while when you had 80,000 items the data took longer. Once again the comb sort proved to be the fastest because it only had to separate the last 20 % of data. The other sorting techniques took longer because they must go through each item and then sorting the remaining data. Finally, the times almost seemed to quadruple as the number of items in the array got larger.

Conclusion:

In conclusion, the comb sort seems to be the best algorithm to use when sorting data. The slowest algorithm seems to be the insertion sort because it only switches the data if it’s out of place. This experiment showed how different algorithms respond to the problem of sorting. The only problems I ran into were having to remember on how to the text editor in putty and when the server went down. Overall, this experiment showed me the different methods of how algorithms sort large quantities of data.