Sidney Sanders

Professor Venu Dasigi

CS 4120 – Design and Analysis of Algorithms

25 October 2021

Project 1 Report

Ranking in the order i believe the algorithms will perform in from worst to best is first with bubble sort with the time complexity of 0(n^2) and insertion sort algorithm with the time complexity of 0(n^2). Then continuing with quick sort with a complexity of 0(nlogn), Heap with the time complexity of 0(nlogn), and merge sort with the complexity of 0(nlogn).

My initial Expectation for the algorithms is that bubble sort would rank towards the back of most categories compared to the others. I think Insertion would be quick with the smaller sets of data and maybe not as quick with the larger sets. I think quick sort, when sorted from middle will be slower than the quick sorted pivoted by the random point. Then heap sort and Merge as the two fastest at sorting the data. Below are 6 images providing the runs of the algorithms with the three different types of data with variable sizes recorded as well.

  

  

 After reviewing the data, I do believe that the most efficient sorting algorithm with large random data sets is the quicksort method where the pivot is the middle point of the data. It sorted the list of 8000 random integers (figure 1) in half of the time that the quick sorting algorithm with random pivots which was the second fasted method in that category. My initial thoughts of the bubble sort were that it would not be very efficient and i think the data provided can show that bubble and we can see in Figure 1 it does rank in last with this category of data and does in most other categories. My guess on insertion sort was not correct after the data was collected. I assumed it would perform well in the smaller data set category but i was wrong and with the random data sets of 1000 it ranked in the second to last position only preforming better than bubble sort as seen in figure 2. My other assumptions were ok in assuming the order in ranking.

Figure 2 Random 1000 integer Set Ranking

Figure Random 8000 integer Set ranking