Assignment 6

Report on the speed of several sorting algorithms

Ryan Lindner
2260250
Chapman University
12/14/18

***Abstract*—For this assignment, I will be testing the run-times of Quick Sort, Insertion Sort, Bubble Sort, and Bogo Sort. This is all written in C++ and tested by using the chrono functions within the standard library. I will be displaying the time elapsed from the start to the end of the sorting.**

***Index Terms*—Quick Sort, Insertion Sort, Bubble Sort, Bogo Sort**

I. Were the time differences more drastic than you expected?

Yes, I was surprised by such the large differences. First, I was shocked at how awful the randomness of Bogo Sort performs. It has a worse case of O(infinity), but on average is O(n*n!). An example of how bad this is: Taken a list of 10 numbers, the other three sorting algorithms took less than 1 millisecond, whereas Bogosort took 15,912 milliseconds. It is truly awful. That being said, the other three still had significant differences in time. Quicksort was by far the quickest, sorting 10,000 numbers in 2 milliseconds, whereas Insertion Sort took 90, and Bubble Sort took 532.

II. What tradeoffs are involved in picking one algorithm over another?

The tradeoff in picking one over another has to be complexity vs. efficiency. Quick Sort was by far the fastest sorting algorithm, but also difficult to implement, which means that if you are pushing a deadline, you may not want to go with this. On the other hand, Bubble Sort was very simple to implement, but it ended up being quite a bit slower than Quick Sort. I never recommend using Bogo Sort.

A. How did you choice of programming lnaguage affect the results?

C++ made coding easier for me personally, as I had control of pointers and could manipulate them as I wanted. However, C++ has no easy way to timestamp a before and after time of an algorithm. I had to resort to using chrono's high-resolution-clock, which may not be as accurate as other tests may be.

III. What are some shortcomings of this empirical analysis?

Every time I ran this test, I would get different results. Of course, they all showed the same patterns, and I am probably better off averaging them and calculating a standard deviation. However, I used the different results I got to form a trend and see a clear difference in the speed of these sorting algorithms.