Joseph Gibson

Introduction

The objective of this assignment was to implement five sorting algorithms (selection sort, insertion sort, bubble sort, merge sort, and quick sort), test each one for correctness and analyze their performance.

Theoretical Analysis

We can expect the average case of time growth for each algorithm we are analyzing. Thus, we expect that quick sort will be O(nlogn) while all others will be O(n^2).

Growth rate	n = 10	n = 50	n = 100	n = 500	n = 1000
O(n^2)	1000	25000	100000	2500000	10000000
O(nlogn)	1000	11609	33219	282192	106643

(fig. 1) expected growth rate of given inputs

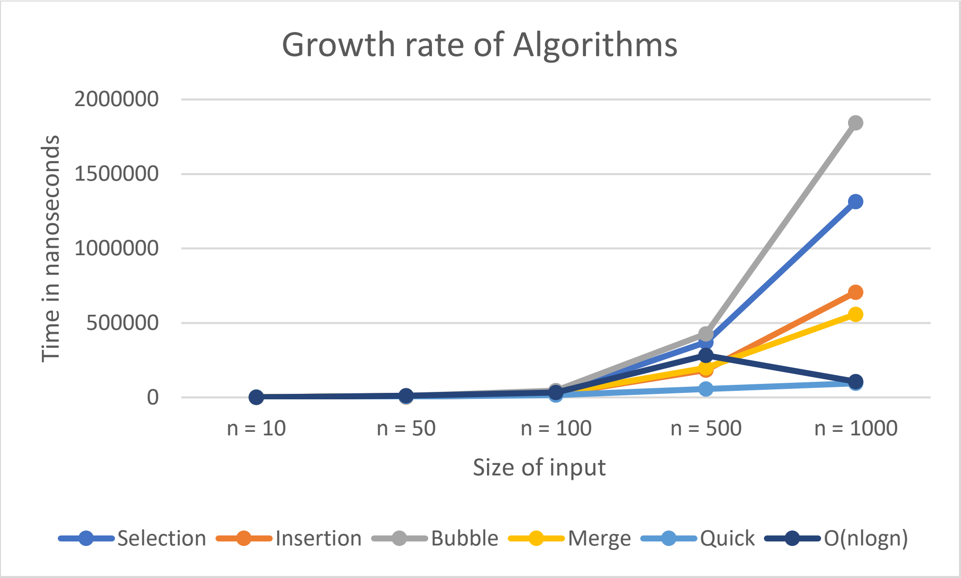
Experimental Setup

Tests for correctness were done simply via standard output and with the provided input samples. To analyze the time taken by each algorithm, a random number generator was used to generate 5 sets of random integers. The length of each set was such that S_1 is of size 10, S_2 is of size 50, S_3 is of size 100, S_4 is of size 500, and S_5 is of size 1000. Each algorithm is tested with each set 10 times, and the floor of the mean time taken per set is taken to compare each algorithm. The best and worse cases for each algorithm are not considered as they can be considered statistically unlikely and unhelpful for an analysis measuring average case.

The main output file implements a switch case to perform each algorithm where each algorithm is numbered 1-5 respectively, corresponding with the number at the top of each sample input file. The chrono library is used to set the start time right before the algorithm is called and an end time right after. The difference is the amount of time taken by the algorithm in nanoseconds.

Experimental Results

As displayed in fig. 2 we can see that all algorithms are at least within quadratic time, but only quick sort is within quasilinear time. A scale of theoretical quadratic time is omitted due to the scale of the chart being skewed. Additional n-sizes in between data points as well as greater n-sizes beyond n = 1000 would provide a nicer, smoother picture.



(fig. 2) algorithms and their growth rates compared to theoretical growth rate