

PROGRESS REPORT OF THE GRAY OWLS' EXPERIMENTS

İbrahim Türkmen 150140002, Kürşat Yaşar 150130064, Baran Kaya
150130032

Introduction

The purpose of the document is to report the progress of the experiments proposed by the Gray Owls group on the 24th of March, 2017 and to remind those whom it may concern of the importance of this project. Our progress has been faster than expected and we may finish earlier than what we initially proposed on our Proposal Report (which is 14/04/17). No problems have arisen and only two innocent mosquitoes that made the mistake of flying into my room have been confirmed as collateral damage. Our results so far are matching our theoretical estimations with a small error of margin. In this document we will repeat the importance and reason of these experiment as well as the conditions of these experiments, elaborate on our results and progress so far and report any new developments.

Project Description

This Project was proposed on 24th of March, 2017. This project is the creation of a new program or app that helps users find the locations of the closest establishments (ex: Satanic Cult Altars) that the user desires to find. The key point of the program is the sorting algorithm that is to be used once the establishments are found which brings us to the proposed experiments. We have already mentioned in the Proposal Report how different input types can change the performance of sorting algorithms. We wish to examine how the speed and growth rates of Insertion Sort, Merge Sort and Quick Sort changes when the input is of the float type. And then we will confirm the best suited algorithm for this program.

The setup of the experiments are:

- Operating System: Windows 10
- RAM: 6GB
- Compiler: Dev C++ 11, with option : -stdc++11
- Language: C++

For accurate measurements the 'chrono' library is utilized. Detailed Information is provided in our Project Proposal.

Progress

Completed tasks are:

- Writing and compiling the codes of Insertion, Merge and Quick Sort
- Designing simple programs that will allow us to easily control the inputs of the sorting algorithms and measure the speed of the startup time and execution time.
- Initializing a 1000 length array of random pairs of numbers that represent the coordinates of the establishments
- Setting up the desired environment and conditions for the experiment to be run
- Organizing and attending the funerals of the deceased mosquitoes

The tasks that are in progress are:

- Running all the trials through the programs
- Provision of sustenance

The tasks that are yet to be worked on are:

- Reaching a conclusion on which algorithm we should use for our app
- Beta testing the program
- Calling the police on the burglar that just broke into my home as I am typing this sente...

Insertion Sort Preliminary Results

Ibrahim Turkmen has run 4 trials of his corresponding program. On each trial the number of establishments is increased and coordinates of the user is altered so that when the distance between the user and the establishments are calculated a random array of floats is generated. The results are presented in figure 1.

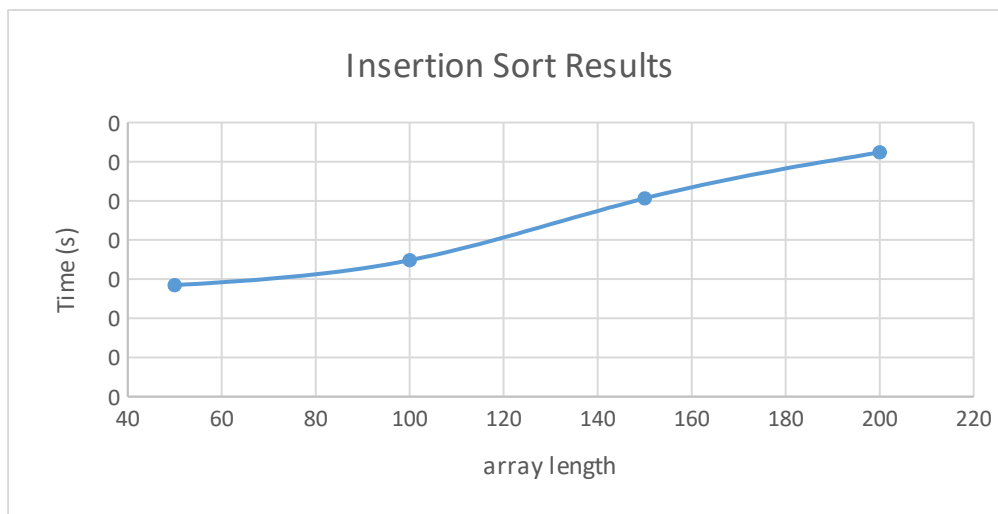


Figure 1. The results so far indicate that the algorithm has performed indifferently to its normal behavior. Further analysis may yield a different suggestion.

Quick Sort Preliminary Results

Baran Kaya has run the program under the same inputs five times and taken the average of the outputs in order to obtain more accurate results. His progress so far is shown in figure 2.

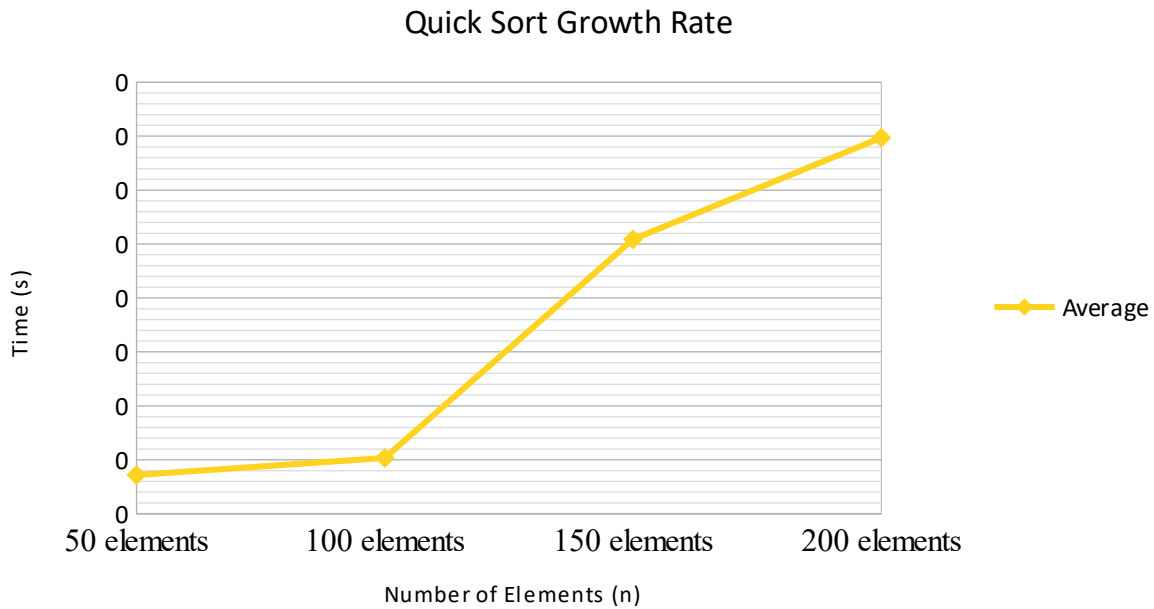


Figure 2. The growing rate of quick sort is similar to quick sort's normal rate graphs. My expectation for the future is graph is going to increase but the slope can be change.

Merge Sort Preliminary Results

Kürşat Yaşar has run the program to take 4 measurements with the same given input. In the given input X and Y coordinates were zero. Program has been run for 4 different array size and for each array size Kürşat Yaşar run the program five times to take average of results. The result can be seen in figure 3. Because merge sort algorithm is one of the $\Theta(n \log n)$ algorithms in the future graphs we expect using of merge sort will be more efficient and advantageous.

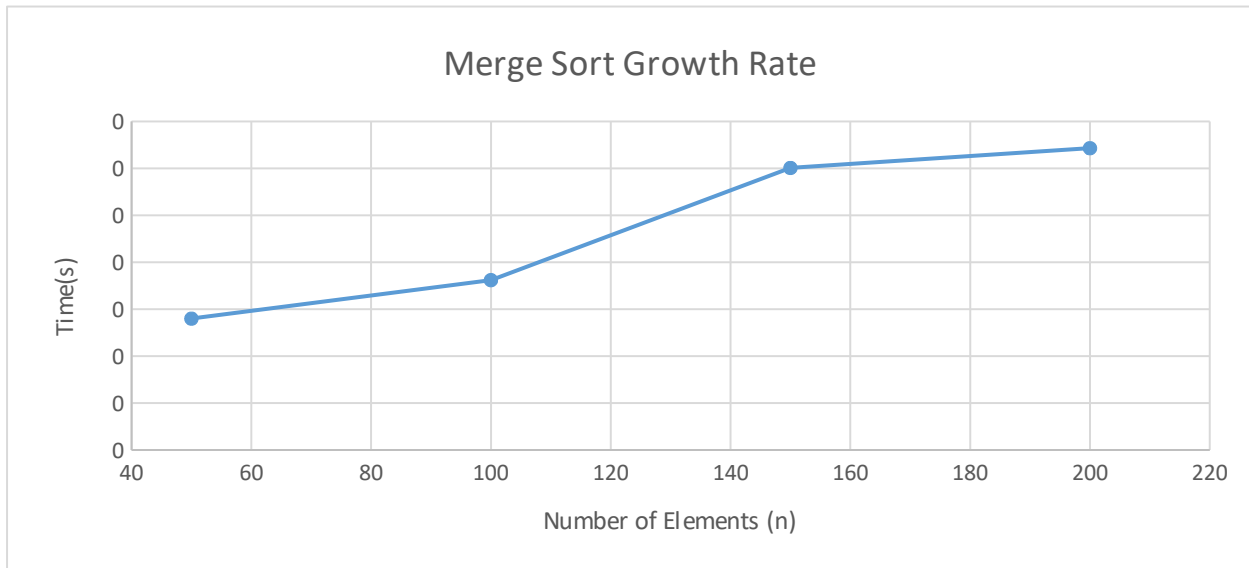


Figure 3. This graph shows time which is taken to sort n elements with merge sort algorithm.

Progress Assessment

The progress has been going faster than anticipated it can be inferred that the completion of the experiments will be ahead of schedule and the development of the app may resume then after.