Lab 0

Martin Mueller

February 6^{th}, 2019

1 Introduction

The goal of this lab is to compare Java's built in sorting algorithm with selection sort. First, a program was created that constructs two identical arrays with set numbers. The arrays were then sorted. The times it took to sort with both Java's default sorting algorithm and selection sort were then recorded. This process was repeated with arrays of various sizes. Each array size as well as the speed of each sorting algorithm in milliseconds was recorded and put into the table below:

2 Data

Array Size	**Default Sorting Algorithm**	**Selection Sort**
10	0	0
100	0	0
1000	1	2
10000	2	36
100000	9	883
1000000	99	92725

3 Lab Questions

1. How long would it take for each algorithm to sort 10 billion numbers?

It's easy to see a numerical pattern in the table of values for Java's default sorting algorithm. It looks approximately linear. Since 10 billion is 1000 times greater than 1 million, the largest array size provided by the table, one must simply multiply the sorting algorithm's speed at 1 million by 1000 to get a decent estimation. Therefore, it would take approximately $\boxed{99,000}$ (1 billion*1000) milliseconds for Java's default sorting algorithm to sort through an integer array of 10 billion units.

In contrast, it is difficult to see a numerical pattern in the table of values for selection sort. Because of this, it might be a good idea to take a look at the code. From there, it's easy to see that it has a time complexity of $O(n^2)$, so an increase in the size of the array by a factor of 1000 would mean that selection sort would take 1 million (1000^2) times the amount of time it did to sort the array containing 1 million units. Using this, one can get an estimation of $\boxed{9.2725 * 10^{10}}$ milliseconds for the run time.

2. 1 trillion numbers?

Using a similar methodology as above, we can then calculate estimates for an array containing 1 trillion numbers (100 times larger than a 10 billion unit array). For Java's default sorting algorithm, it would take approximately $\boxed{99,000,000}$ milliseconds ($100 * 99,000$), and for selection sort, it would take approximately $\boxed{9.2725 * 10^{14}}$ ($100^2 * (9.2725 * 10^{10})$) milliseconds.

3. 1 quadrillion numbers?

1 quadrillion is 1000 times greater still than 1 trillion. For the default sorting algorithm: $1000 * 99,000,000 = \boxed{9.9 * 10^9}$. For selection sort: $1000^2 * 9.2725 * 10^{14} = \boxed{9.2725 * 10^{20}}$.