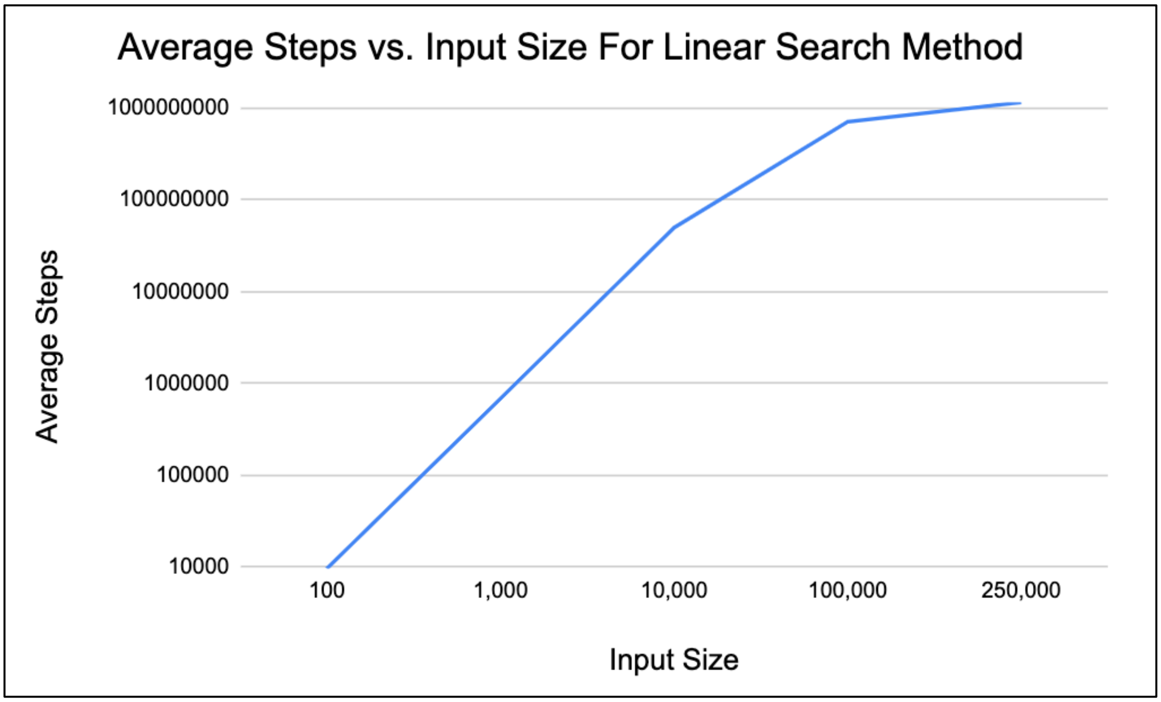
Q2 PDF Report

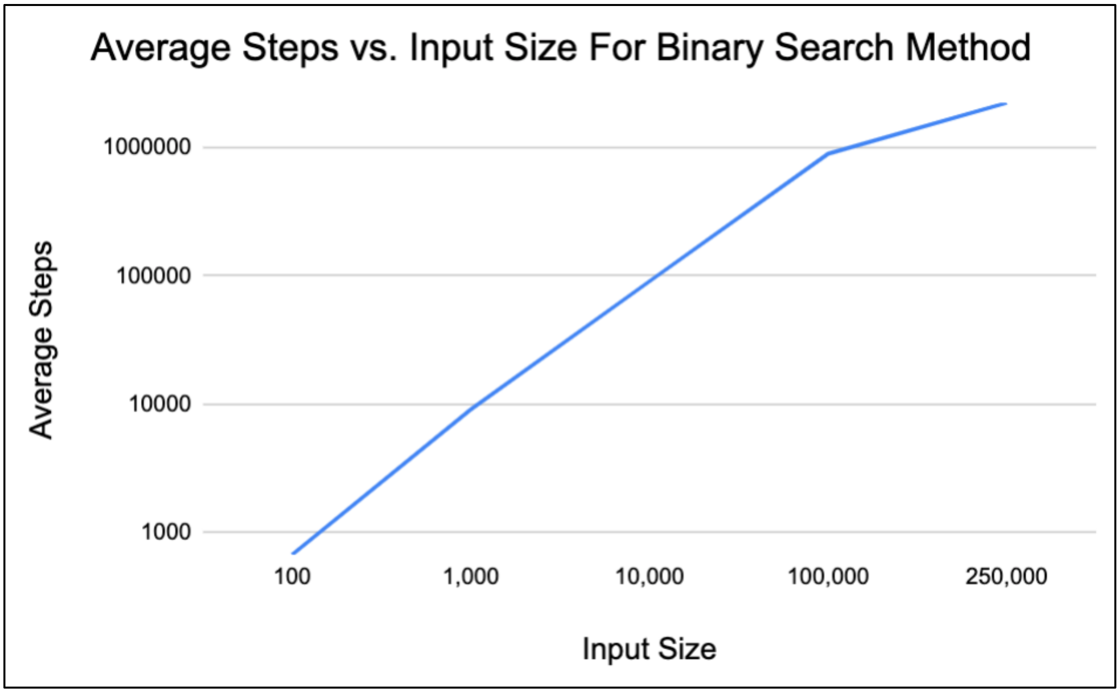
*****Y-AXIS ON GRAPH IS LOG SCALED!! (for the excel screenshot with data, please look at Q2 Folder)*

The three algorithms which were explored were Linear Search, Binary Search, and Binary Recursive Search. Each program takes in a key and an array of integers. The array of integers is randomly generated with different input sizes. The purpose of each program is to find the given key in the given array. Each program also has a step counter to count the number of times the program loops through to find the given key in the given array. Each program was tested with different sizes of arrays and run 3 times to get an average of the number of steps. This data was plotted and used to compare the algorithms to each other in terms of efficiency.

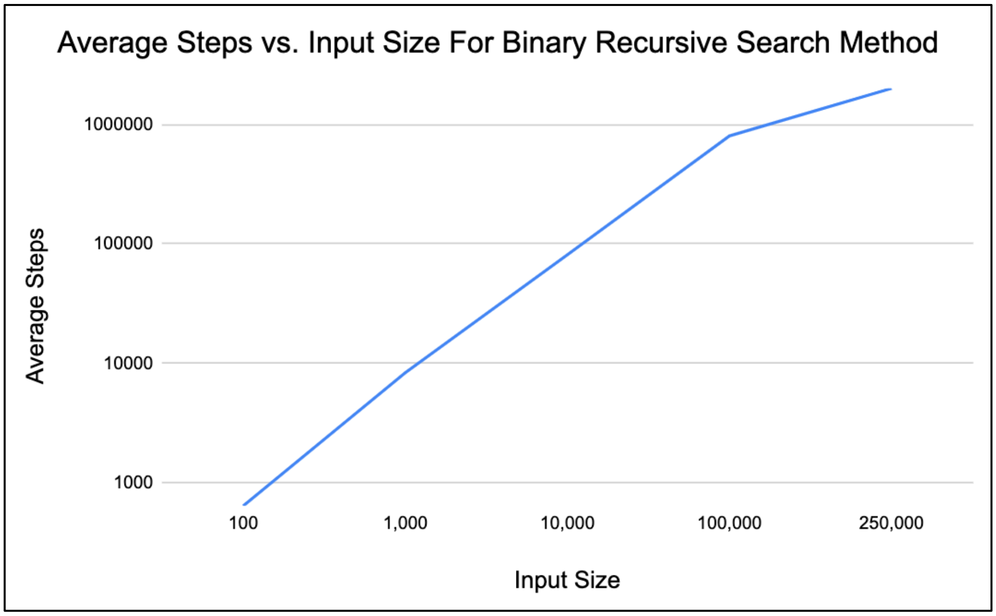
The Linear Search program is the simplest method where a for each loop is used to get each integer in the array and compare it to the key until the matching key is found which then it returns the key. If the key is not found, the program returns -1. The step counter is placed right after the for each loop in order to count the number of times the program has to run through the array to find the key. The following is the plot generated from the data collected by running Linear Search:



The Binary Search program uses a while loop and loops until low <= high. Low and High are the beginning position and the end position of the given array. Each time the loop executes, it finds the middle of the array. Then the program compares the integer at the middle of the array to the given key. If the key is less than the middle integer, then high is set to the integer preceding the middle integer and the loop runs again. If the key is more than the middle integer, then low is set to the integer following the middle integer and the loop runs again. Finally, if neither of those cases apply, then the integer at the middle of the array is the given key. The step counter is placed right after the while loop to count how many times the loop is executed. The following is the plot generated from the data collected by running Binary Search:



The Binary Recursive Search program uses recursion by calling another version of rank. Once again, the middle of the array is calculated by adding low to the difference of high and low all divided by 2. The key is compared to the integer at the middle of the array and if the key is lower, then rank is called to run on the half of the array which the key is in and the other half of the array is ignored. This same logic is followed if the key is higher than the integer at the middle of the array. The step counter is placed after both of the if statements which compare the key to the middle of the array. The following is the plot generated from the data collected by running Binary Recursive Search:



After collecting the data and comparing the number of steps counted from each input size, the conclusions are clear. The number of steps produced by the Linear Search program was always significantly higher than both the Binary Search program and the Binary Recursive Search program. When comparing the difference between the number of steps produced by the Binary Search program and by the Binary Recursive Search program, Binary Recursive Search always won. With smaller input sizes, the difference was super high. For example, for an input size of 100, the average number of steps was 668 for Binary Search, but for Binary Recursive Search, the average number of steps was 647. However, when the input size was as high as 250,000, the difference of the average number of steps was 250,085 which is much more significant. It is clear that the Binary Recursive Search is much more effective than Binary Search and Linear Search is the least effective of all three algorithms.