ABSTRACT

The second experiment is composed of performing a binary search for a specific value within an array of five different sizes. One test file was created, test3.html, where the Javascript code for performing a binary search is written. The same binary search is performed on 5 different array sizes, 30 million, 15 million, 7.5 million, 3.75 million, and 1.875 million. The html that exists in the document is just for displaying the results of the experiment.

EXPERIMENT

test3.html

Array Dimension: 1D

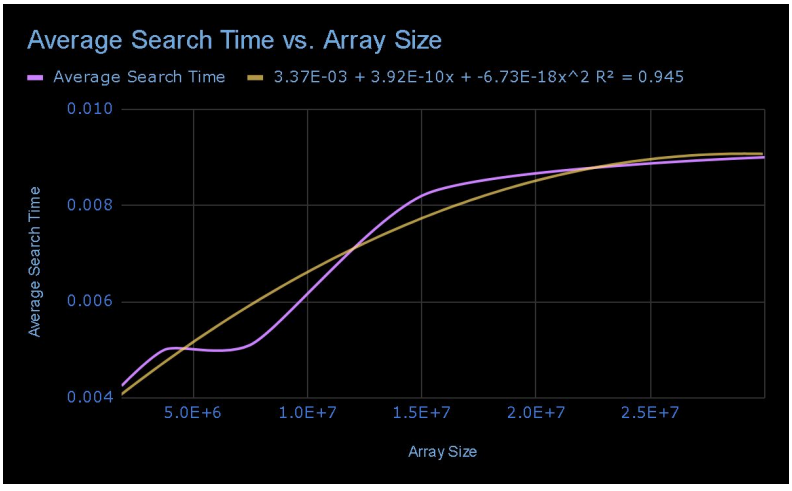
Array Size(s): 30000000, 15000000, 7500000, 3750000, and 1875000

Operation: Binary Search

At the start of the file, all the variables and constants are declared, including the array being inspected and the value to be searched for. The newly initialized array then enters a for loop where its contents are filled with sorted data (integers that increment by 1). Immediately before the script enters a while loop, it takes a snapshot of the current time in milliseconds using performance.now. After the first time snapshot, the script enters a while loop and begins searching for the indicated value (a value which is guaranteed to exist in the array, 16070104). Then, when the value is found, the script leaves the while loop and immediately another time snapshot is taken. This operation is performed 100 times so as to gather 100 data points for better result analysis. The difference is taken between the second snapshot and the first and the result is stored. Later, after the binary search has been performed 100 times, the results of all the time deltas are summed and then divided by 100 to obtain the average elapsed time to perform a binary search on an array. This experiment is performed 5 more times for the other four array sizes. Array size vs. average search time is plotted on a graph.

RESULTS

The results of this experiment were predictable and expected. The experiment proved that as an array gets larger, then the time it takes to read values from that array and perform a search on it also increases.



More specifically, the equation that best represents the association between array size vs. search time is $-6.73 * 10^{-18}x^2 + 3.92 * 10^{-10}x + 3.37 * 10^{-3}$. This equation is of polynomial form because there is roughly an x^2 correlation between array size and search time. It appears that after the array size exceeds 25 million, there is not much change in average search time, which is a little surprising. My initial hypothesis was that array size vs. search time was logarithmic and as array size increases, the search time also increases exponentially. But the graph of the data clearly shows that average search time plateaus after the array size exceeds 25 million. The R^2 of the equation (the degree to which the equation matches the data) is 0.945, which is very good. It shows that ~95% of the data matches the equation given by the curve.

CONCLUSION

The most likely reason for this relationship between array size and binary search time are somewhat obvious. As an array grows, the amount of values that must be inspected during the search will grow as well. For Javascript, and in this experiment specifically, there was indeed a positive polynomial correlation between

array size and search time. The most likely reason that it was best-fit by a polynomial equation is probably because the binary search algorithm is not really affected by an array's size. The underlying mechanisms of binary search is through a divide-and-conquer approach. While the search is being done, the array is split in half with each iteration and as the array size gets larger, the number of groups that the array is split into increases too but it plateaus after a certain array size. This is probably because if an array of, say, 500,000,000 values is split into a smaller array of 250,000,000 then the probability of finding the value within that group would not be that much greater than if an array of 510,000,000 was split into a group of size 255,000,000. In other words, the search time reaches a limit where any increase in size will not have much effect on search time.