Lab Report 1

Liron Mizrahi - 708810

August 2016

1 Linear Search

1.1 Best and Worst Cases

Linear Search will search iteratively over a list to find the key.
The best case position would therefore be the first position in the list. There would only ever be 1 check to find the key regardless of the array size which would result in $O(1)$.

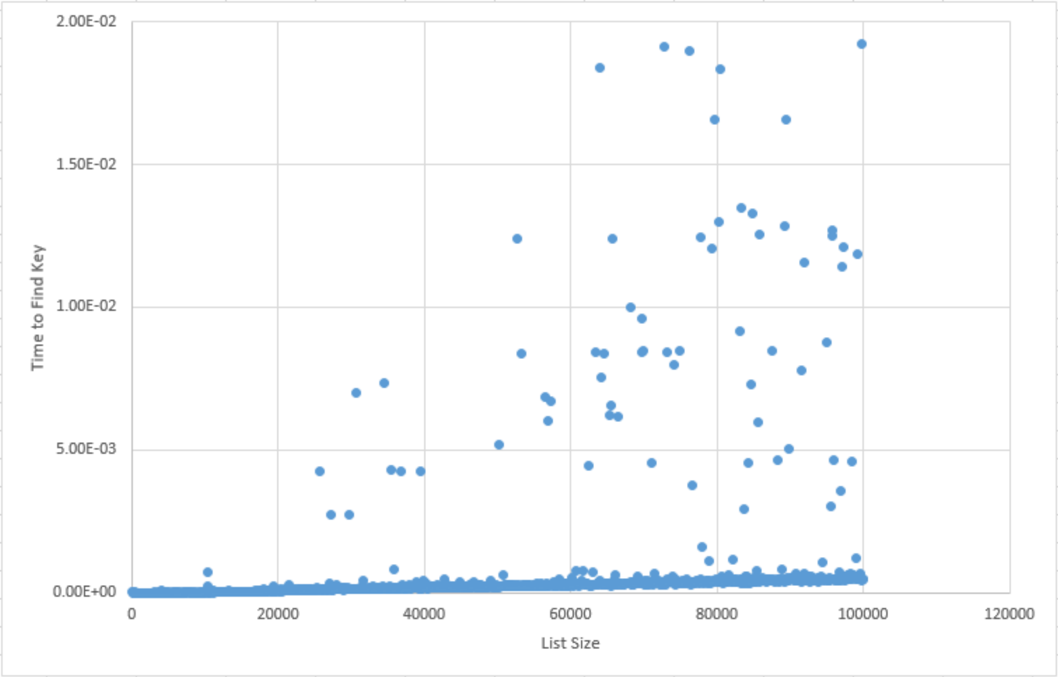

Figure 1: Linear search - Best case

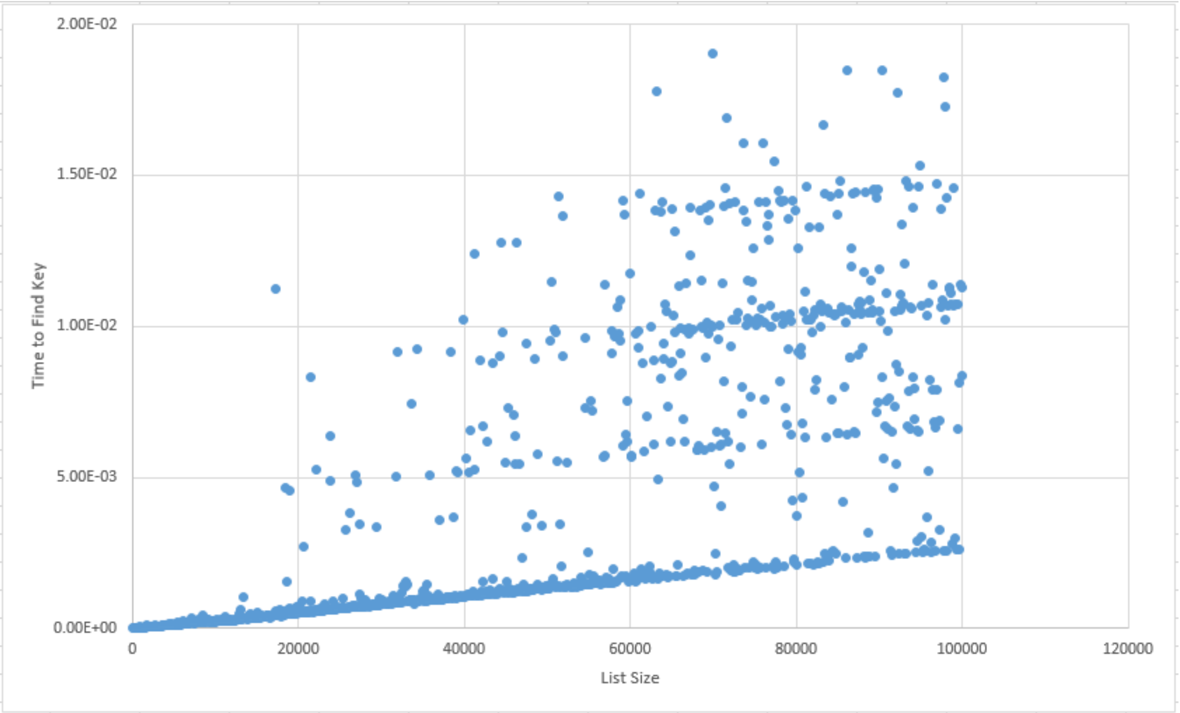

Figure 2: Linear search - Worst case

It can clearly be seen that the best case grows close to $O(1)$ while the worst case is $O(n)$.

1.2 Average Case

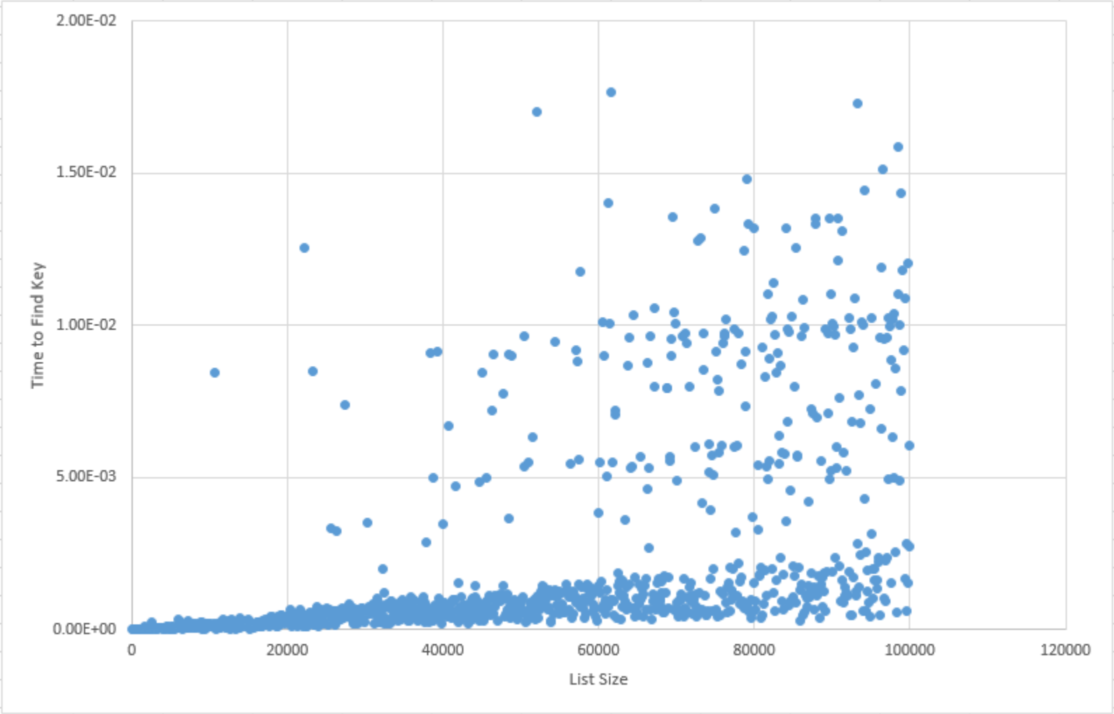

Figure 3: Linear search - Average case

The average case has a lot of data points in between the worst and best cases. The average case is also $O(n)$.

2 Binary Search

2.1 Best and Worst Cases

The best case for binary search would be if the key was at the mid point of the list. The worst case would be if the key was at the 2^{nd} last entry or at the midpoint-1.

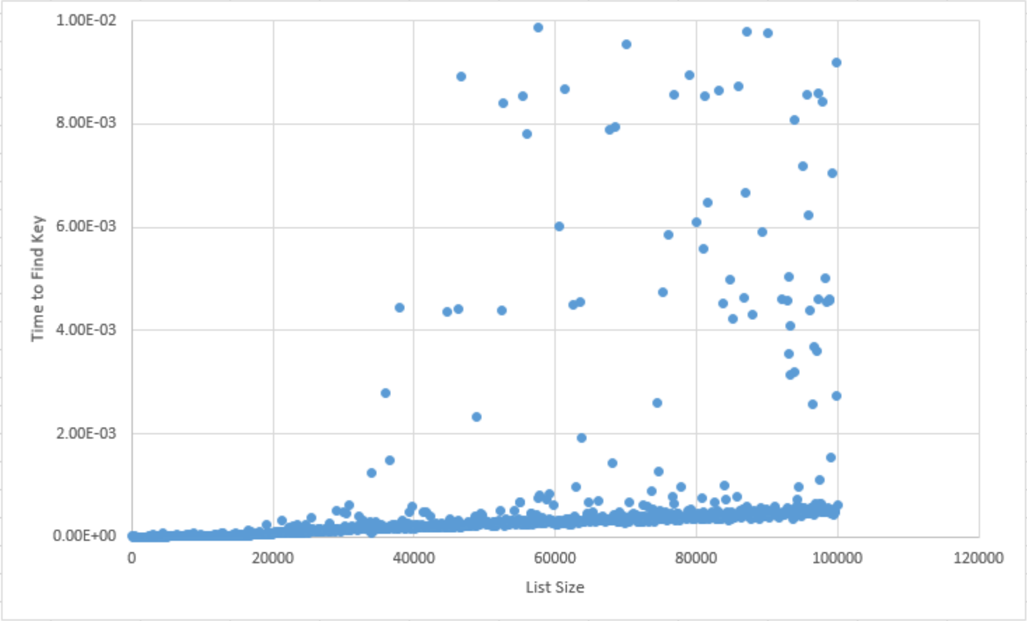

Figure 4: Binary search - Best case

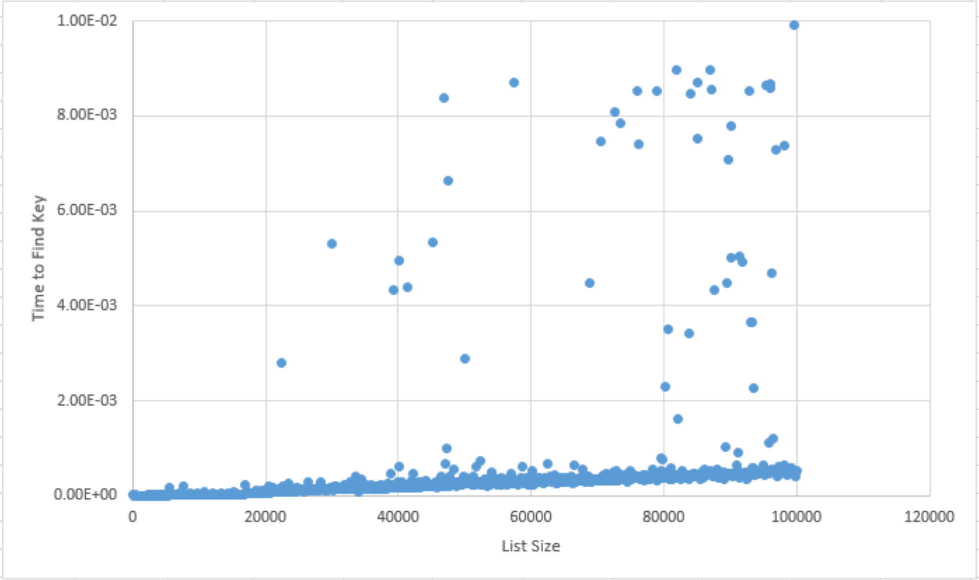

Figure 5: Binary search - Worst case

The best case for binary search would be $O(n)$, which can be seen in the graph above. The worst case would be $O(\lg n)$ as the search builds a tree with height $\lg n$. It's not obvious from the above graph but for large enough list size there would be a significant difference.

2.2 Average Case

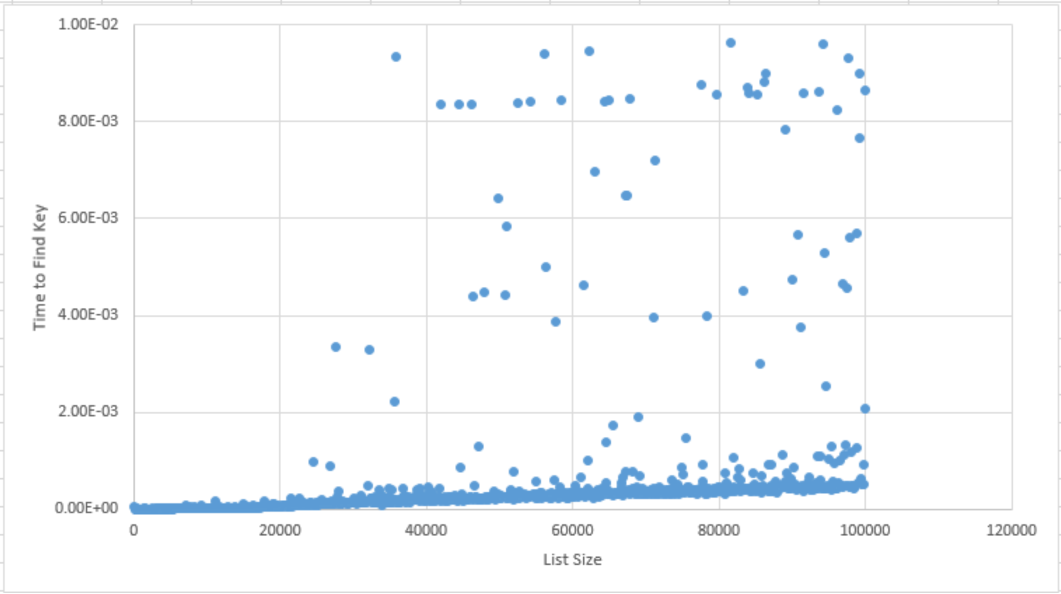

Figure 6: Binary search - Average case

The average case grows similarly to the worst case but will grow slightly slower. Therefore the average case will be $O(\lg n)$ but will smaller constants than the worst case.