

## Timing Search Algorithms Report

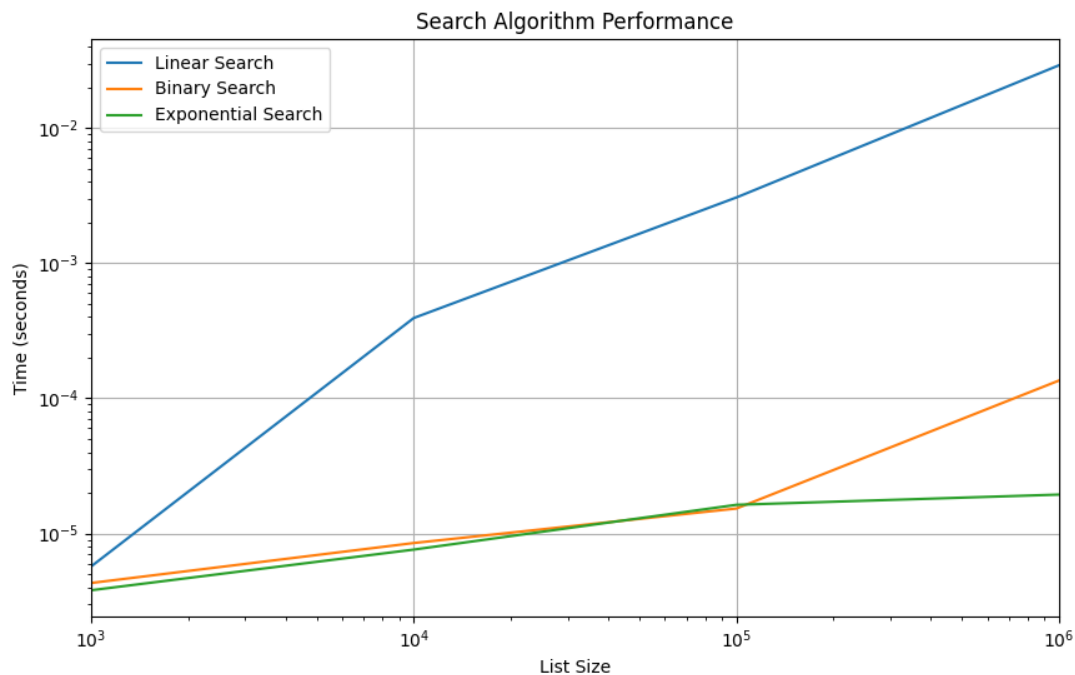
### Overview:

In this project I performed timing experiments for different search algorithms (linear, binary, and exponential) on lists of varying sizes. It measures the time taken for each search algorithm to find a target value in the list and writes the results to a CSV file named "searchTimes.csv". The CSV file contains the list size and the corresponding search times for each algorithm. Additionally, it generates a plot to visualize the performance of the search algorithms. I tried to profile the code to find what functions it spent the most time on, but it unfortunately ended up taking too long to run.

### Analysis:

List Size(ints)	Linear Search(seconds)	Binary Search(seconds)	Exponential Search(seconds)
1000	6.61E-05	7.00E-06	8.00E-06
10000	5.24E-04	9.30E-06	6.90E-06
100000	1.99E-03	1.25E-05	1.50E-05
1000000	6.64E-02	2.06E-05	2.11E-05

Table 1: Runtimes for Linear, Binary, and Exponential Search Algorithms on varying sized lists of integers



### Conclusion:

The results of this project highlight the significant differences in performance between linear vs binary and exponential search algorithms as list sizes increase. As expected, linear search, with its  $O(n)$  time complexity, exhibited the slowest performance, with its runtime increasing substantially as the list size grew. In contrast, binary and exponential, with their  $O(\log(n))$  time complexity, searches demonstrated much more efficient scaling, with their runtimes remaining relatively small even for larger lists.