Group Laboratory Exercise

Group: 4
Members: Calma, Eugene Marie S.
Costa, Kenneth John P.
Feranil, Ted Paulo A.
Herrera, Derick Carlo S.
Suaverdez, Leoj M.
Vergel, Chean Bernard V.

		Linear	Binary	Ternary	Exponential	Interpolation	Jump
Target Set	Search Data	Time in Milliseconds					
100	8	0.0039	0.0037	0.0041	0.0066	0.0040	0.0114
	27	0.0021	0.0042	0.0037	0.0052	0.0038	0.0074
	35	0.0030	0.0029	0.0036	0.0055	0.0036	0.0059
	60	0.0038	0.0032	0.0033	0.0106	0.0031	0.0055
	94	0.0036	0.0023	0.0029	0.0061	0.0023	0.0061
		0.00328	0.00327	0.00351	0.00679	0.00336	0.00726
1000	45	0.0193	0.0193	0.0112	0.0167	0.0177	0.0924
	73	0.011	0.011	0.0113	0.0124	0.0075	0.0201
	102	0.0168	0.0208	0.0168	0.0115	0.0107	0.0222
	576	0.0619	0.0111	0.0223	0.0333	0.0098	0.0268
	968	0.1089	0.0175	0.0123	0.0179	0.0083	0.0315
		0.04358	0.01594	0.01478	0.01836	0.0108	0.0386
10000	243	0.0268	0.1584	0.4249	0.9148	0.0099	2.2813
	2004	0.7448	0.0175	0.0271	0.0202	0.0091	0.0269
	4006	0.4386	0.0290	0.0134	0.0242	0.0152	0.0312
	8012	0.8564	0.0139	0.0418	0.0316	0.0121	0.0414
	9899	1.0282	0.0216	0.0214	0.0230	0.0108	0.2204
		0.61896	0.04810	0.10574	0.20276	0.01142	0.52026

TABLE 1. *Search time results of different search algorithms per data set in milliseconds.*

Figure 1. Line graph on time comparison of search algorithms(1-100 Data Sets)

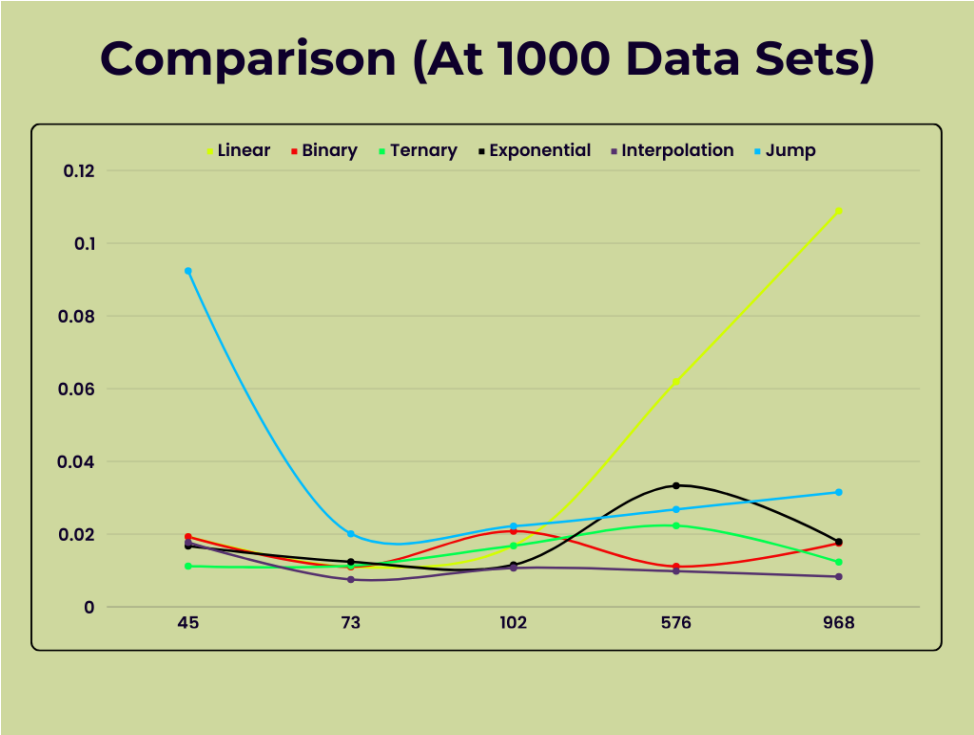

Figure 2. Line graph on time comparison of search algorithms(1-1000 Data Sets)

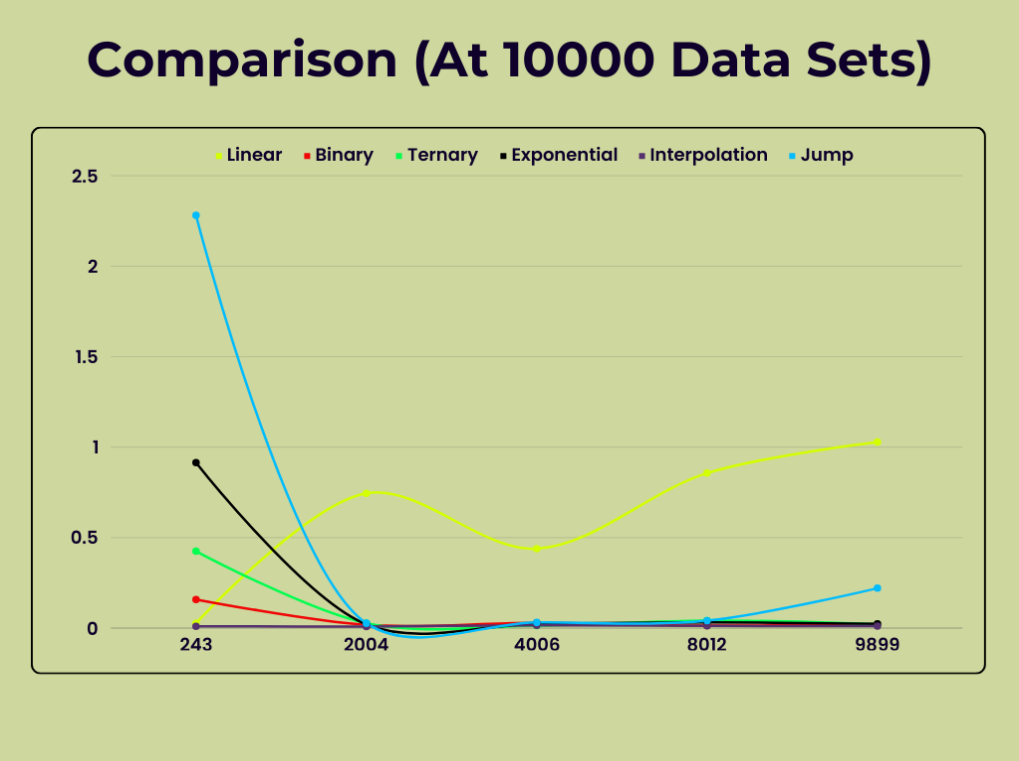

Figure 3. Line graph on time comparison of search algorithms(1-10000 Data Sets)

a. Which search algorithm performed the best overall?

It's clear that the fastest search algorithm depends on the size of the target set. In smaller target sets (e.g., 100), Binary Search proves to be the most efficient, utilizing its logarithmic time complexity in sorted datasets.

However, as the target sets increase in size (1000 and 10000), Interpolation Search takes the lead, demonstrating its effectiveness in scenarios with larger and evenly distributed data. The interpolation strategy, which estimates the likely position of the target based on neighboring values, becomes increasingly advantageous in these contexts.

Therefore, the optimal selection of a search algorithm relies on the specific characteristics and scale of the dataset. For smaller datasets, Binary Search may be the preferred choice, while for larger datasets, Interpolation Search emerges as the algorithm that performs best. In summary, given that Interpolation Search outperforms other search algorithms in two datasets, we can conclude that the Interpolation Search algorithm demonstrated the most optimal overall performance.

b. Did any search algorithms perform better on specific data sets?

In the context of varying data set sizes, specific search algorithms display different performance characteristics. In smaller data sets with 100 elements, Binary Search demonstrates consistent efficiency across various target numbers, closely followed by Linear Search, Interpolation Search, and Ternary Search, all showcasing relatively low search times. However, Jump Search exhibits comparatively lower performance among these algorithms. In medium-sized data sets comprising 1000 elements, Interpolation Search stands out for its consistent performance across different target numbers. Jump Search also demonstrates decent performance but with some variability depending on the specific target number. Linear Search, in this scenario, shows increased time requirements compared to more optimized algorithms. In larger data sets with 10000 elements, Binary Search's efficiency notably decreases, evident in significantly increased search times compared to smaller data sets. Conversely, Interpolation Search maintains relatively stable performance across various target numbers within this larger data set. Additionally, other algorithms, such as Jump Search, exhibit varying performance depending on the specific target number being searched.

In summary, Binary Search tends to perform well for smaller data sets but shows degradation in performance as the data set size increases significantly. Therefore, Interpolation Search maintains relatively stable performance across different data set sizes and target numbers, showing adaptability to varying scenarios.

c. How did the size of the data set affect the performance of the search algorithms?

The size of the dataset significantly influenced the performance of the search and proportional increase in search time with larger datasets, reflecting its linear time complexity. In contrast, logarithmic algorithms like binary and ternary search showcased more efficient scaling, with a slower rise in search time across dataset sizes of 100, 1000, and 10000. The performance of interpolation search varied, indicating a sensitivity to dataset characteristics, with an increase in time from 100 to 1000 and potential stabilization or even a decrease at a dataset size of 10000. The comparison underscores the impact of dataset size on algorithm efficiency, highlighting the need for tailored approaches based on specific dataset characteristics and the inherent time complexities of the search algorithms employed.algorithms, as evident from the provided data. Notably, linear search exhibited a direct.

d. Write a brief conclusion summarizing your findings

The group's experimentation with various search algorithms revealed distinct performance differences based on their search speed measured in milliseconds. The interpolation algorithm stood out with the best overall performance, averaging 0.01053

milliseconds. Notably, they found that the algorithm's efficiency varied with the size of the target set: for ranges of 1-100, the Binary Search excelled at 0.003 milliseconds, while the Interpolation algorithm showed superior performance in larger ranges, averaging 0.0108 milliseconds for 1-1000 and 0.0114 milliseconds for 1-10000. This led to the conclusion that dataset size significantly impacts search algorithm performance, with larger sets potentially diminishing search efficiency.