Polytechnic University of the Philippines
College of Social Sciences and Development
Department of Cooperatives and Social Development
Sta. Mesa, Manila

Group Name	*Group 6*
Course	BSCPE 2-4
Subject	Data Structures and Algorithm
Title of the Output/ Written Work	**Data Search Entry**

Members:

Gupit, Shane Hazel S.

Nobleza, Bhea I.

Parrish, Paulean Marguerette F.

Quitong, Emanoel M.

Relucio, Elyza Jane G.

Templanza, Kristine Joy F.

Small Data Set

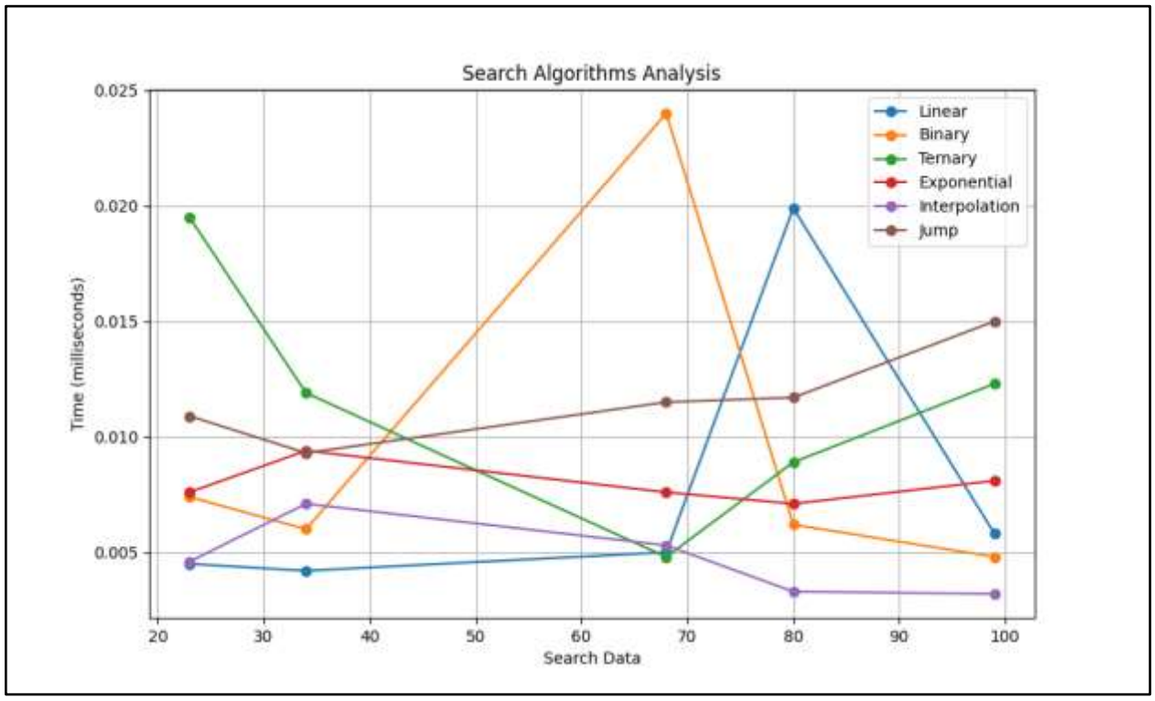


Table 1.1. The table presents a comparative analysis of various search algorithms. The dataset consists of five target values (100, 34, 68, 80, 99), and the time taken by each algorithm to locate these targets is recorded in milliseconds. The table provides a detailed breakdown of the execution times for each search algorithm across different target values.

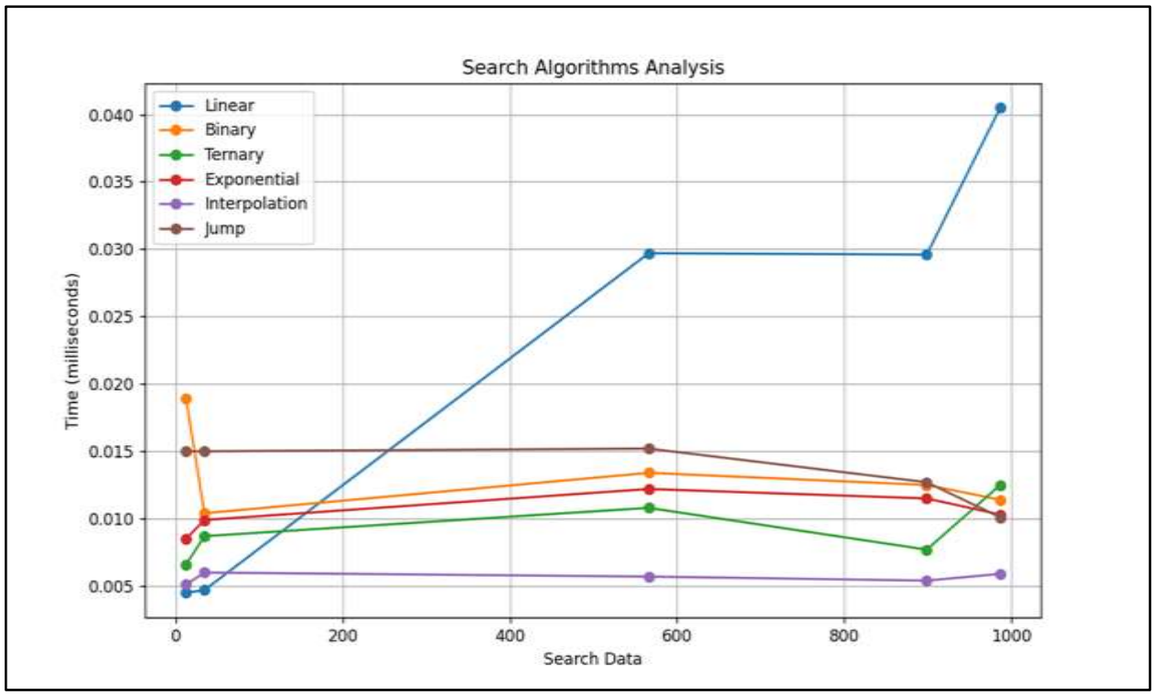
Observations:

In analyzing various search algorithms, distinct patterns in performance emerge. Linear search consistently maintains reliable execution times across different target values, making it a dependable choice for general searches. Conversely, binary search, tailored for sorted datasets, exhibits a consistently low and stable execution time. Ternary search, with its variable performance, offers flexibility but also introduces fluctuations in execution times.

Shifting focus to other searches, exponential search adapts well to unsorted datasets, showcasing slightly wider execution time ranges. Interpolation search excels in uniformly distributed datasets, providing consistent performance across diverse target values. Jump search, designed for large datasets, displays execution times that vary, highlighting its adaptability to different data scales. In summary, the selection of a search algorithm should align with the dataset's characteristics, such as its sorting status, distribution, and size, as each algorithm demonstrates nuanced performance based on these factors.

Polytechnic University of the Philippines
College of Social Sciences and Development
Department of Cooperatives and Social Development
Sta. Mesa, Manila

Medium Data Set



Observations:

Upon delving into the dataset with a target set of 1000, the performance of various search algorithms becomes discernible. For smaller search data values such as 12 and 34, the linear search method consistently proves swift, completing the task in a mere fraction of a millisecond. Binary and interpolation searches, while slightly slower, remain efficient, demonstrating their prowess, especially as the search data size increases. Notably, as the dataset expands to 566, the interpolation search emerges as the most expedient, outpacing other methods. However, for larger search data values like 899 and 987, the linear search becomes noticeably slower, accentuating the algorithm's sensitivity to the size of the dataset. Throughout, the jump search exhibits stability, maintaining competitive times across various dataset sizes, yet not consistently outperforming binary and interpolation searches.

Our exploration of search algorithms reveals that their effectiveness hinges on the size and organization of the dataset. Binary and interpolation searches shine when dealing with sorted or patterned data, while the straightforward linear search holds its ground for smaller datasets. Jump search, while consistently reliable, might not outpace the efficiency of binary or interpolation methods.

Polytechnic University of the Philippines
College of Social Sciences and Development
Department of Cooperatives and Social Development
Sta. Mesa, Manila

Large Data Set

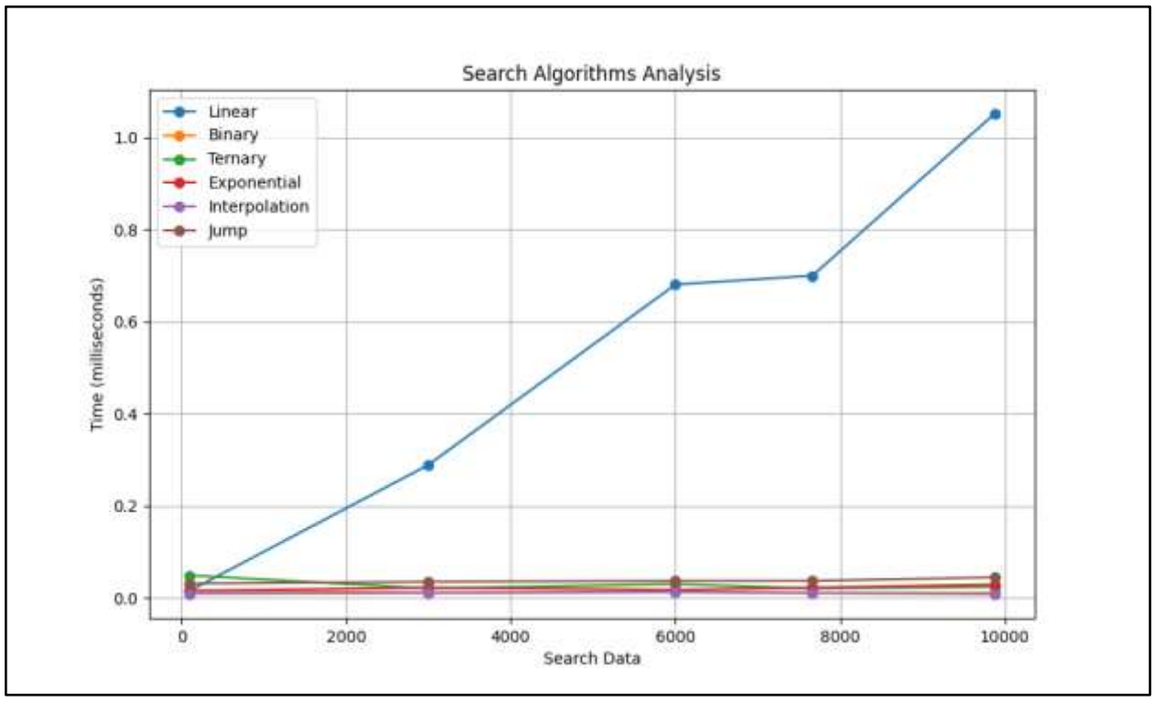
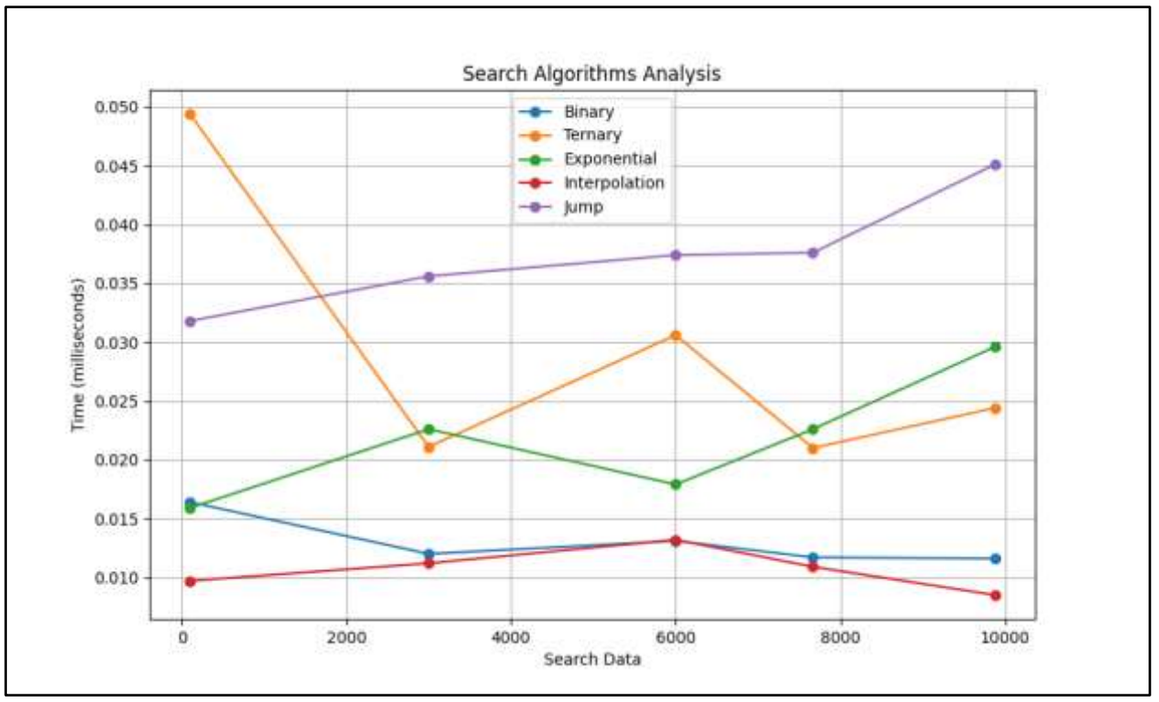


Table 3.1. The table presents a comparative analysis of various search algorithms. The dataset consists of five target values (100, 3000, 6000, 7666, 9877), and the time taken by each algorithm to locate these targets is recorded in milliseconds. The table provides a detailed breakdown of the execution times for each search



algorithm across different target values.

Table 3.2. The table presents a comparative analysis of various search algorithms. The dataset consists of five target values (100, 3000, 6000, 7666, 9877), and the time taken by each algorithm to locate these targets is recorded in milliseconds. The table provides a detailed breakdown of the execution times for each search algorithm across different target values. However, the result of the linear search method was removed to emphasize the results of the remaining methods.

Observations:

The table compares various search algorithms concerning their performance at different input sizes. Linear search generally exhibits moderate execution times that increase linearly with input size. Binary search performs well, particularly for larger

inputs, following a logarithmic pattern. Ternary search is similar to binary search but with slightly higher execution times and also follows a logarithmic trend. Exponential search demonstrates relatively low execution times for smaller inputs, increasing less rapidly than linear search. Interpolation search performs well, especially for larger inputs, showing a complex relationship with input size tied to data distribution. Jump search has mixed performance with execution times varying across input sizes and potential effectiveness for certain data distributions. In summary, algorithm selection depends on the specific data characteristics and task requirements. Binary and interpolation searches excel for larger datasets, linear search for smaller ones, while exponential search is promising for smaller inputs. Jump and ternary searches exhibit mixed results and may require further analysis for specific scenarios.

Analyze the results and answer the following questions:

a. Which search algorithm performed the best overall?

Determining the optimal search algorithm hinges on the specific demands of your task, as various algorithms excel in different scenarios. Analyzing the provided data offers valuable insights:

For smaller datasets, such as Target Set 100, Binary Search stands out as a robust performer with consistently low execution times. Additionally, both Interpolation Search and Jump Search demonstrate competitive performance in this category.

In the realm of medium-sized datasets, exemplified by Target Set 1000, Binary Search remains efficient, maintaining its competitive edge. Interpolation Search and Jump Search continue to deliver commendable results.

As datasets scale up to larger sizes, like Target Set 10000, Binary Search remains competitive, while both Interpolation Search and Jump Search showcase robust performance. Notably, Jump Search exhibits lower execution times in certain cases.

Exponential Search shows promise for smaller datasets but might face challenges maintaining competitiveness as dataset sizes increase. On the other hand, Linear Search generally exhibits higher execution times compared to the other algorithms.

In conclusion, the choice of the "best" algorithm depends on factors such as dataset size and the specific requirements of your search task. For smaller datasets, Binary Search emerges as a strong contender. For larger datasets, both Interpolation Search and Jump Search present competitive options. It is crucial to carefully consider the characteristics of your data and strike a balance between time complexity and implementation complexity when making the algorithmic choice.

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

Upon analysis of the provided table, it is evident that the performance of search methods varies depending on the characteristics of different datasets. In smaller datasets, such as Target Set 100, Binary Search emerges as a strong performer, demonstrating relatively low execution times. Additionally, Interpolation Search and Jump Search prove to be competitive, also exhibiting efficient performance.

As dataset sizes increase to medium-sized datasets, exemplified by Target Set 1000, Binary Search maintains its efficiency and competitiveness. Interpolation Search and Jump Search continue to deliver commendable performance, characterized by efficient execution times.

For larger datasets, like Target Set 10000, Binary Search remains competitive, showcasing efficiency even as dataset sizes grow. Interpolation Search and Jump Search demonstrate robust performance, with the latter showing lower execution times in specific cases.

In summary, Binary Search consistently performs well across all data sets, particularly excelling for smaller and larger datasets. Interpolation Search and Jump Search also prove to be viable alternatives, showcasing competitive performance depending on specific requirements. Ultimately, the selection of the best search method hinges on careful consideration of the dataset size, characteristics, and the desired trade-offs between time complexity and implementation complexity.

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

The performance of search algorithms in the provided table is significantly influenced by the size of the datasets. For smaller datasets, represented by Target Set 100, Binary Search consistently demonstrates low execution times, establishing itself as an efficient choice. Additionally, Interpolation Search and Jump Search showcase competitive performance, providing alternative solutions with efficient execution times.

As dataset sizes increase to medium-sized datasets, exemplified by Target Set 1000, Binary Search maintains its efficiency and competitiveness, showcasing adaptability to datasets of this scale. Interpolation Search and Jump Search continue to perform well, sustaining commendable execution times.

For larger datasets, represented by Target Set 10000, Binary Search remains competitive, highlighting its adaptability to larger datasets with efficient execution times. Furthermore, Interpolation Search and Jump Search exhibit robust performance, with Jump Search displaying lower execution times in specific cases.

In summary, the impact of dataset size on search algorithms varies. Binary Search consistently performs well across all dataset sizes, emphasizing its efficiency and adaptability. Interpolation Search and Jump Search also maintain competitive performance, emerging as viable alternatives, especially for larger datasets where their efficiency becomes more pronounced. The significance of dataset size underscores the importance of selecting search algorithms based on the specific characteristics and scale of the data under consideration.

d. Write a brief conclusion summarizing your findings

The optimal search algorithm selection depends on specific task requirements, with each algorithm excelling in distinct scenarios. Notably, for smaller datasets (e.g., Target Set 100), Binary Search stands out with consistently low execution times, while Interpolation Search and Jump Search also demonstrate competitiveness. In medium-sized datasets (e.g., Target Set 1000), Binary Search maintains efficiency, and both Interpolation Search and Jump Search deliver commendable results. Even as datasets grow larger (e.g., Target Set 10000), Binary Search remains competitive, and Interpolation Search and Jump Search exhibit robust performance, with the latter showing lower execution times in specific cases. While Exponential Search shows promise for smaller datasets, it may face challenges with larger sizes, and Linear Search generally exhibits higher execution times.

The choice of the "best" algorithm hinges on careful consideration of dataset size and specific task requirements, emphasizing the importance of weighing trade-offs between time and implementation complexity.