Data Structures and Algorithms

ANALYSIS & CONCLUSION

SEARCH ALGORITHMS
BSCPE 2-6 | Group 5

MEMBERS

Berte, Gary Delfin

Bulado, Arvey Tristan

Delos Reyes, Kean Angela

Gumafelix, Mary Ann

Matiga, Samantha Heart

Perez, Lea Nikka

Roxas, Angeline

Sutelleza, John Keybird

TIME COM

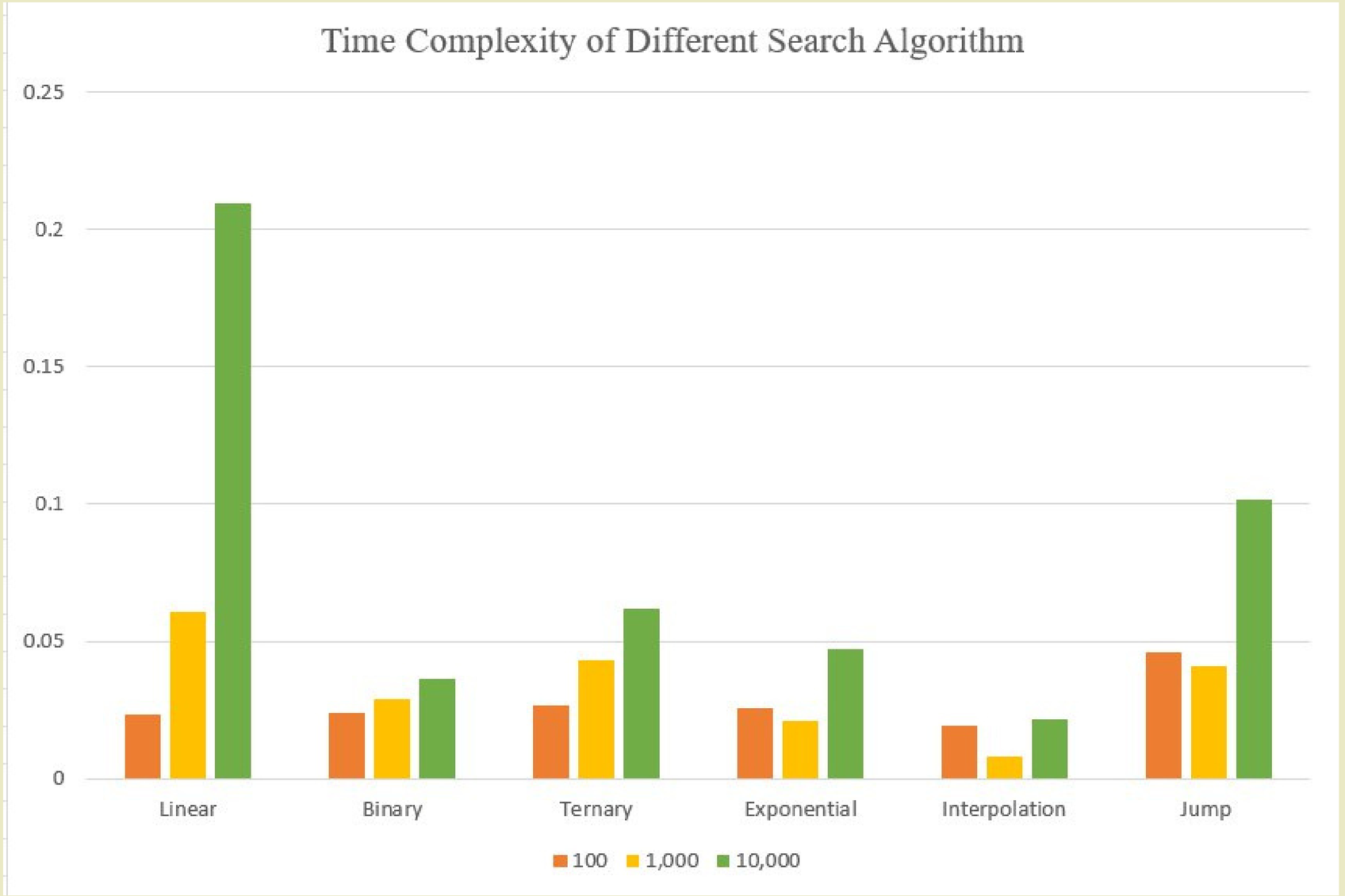

Figure 1

SMALL DATA

Figure 2

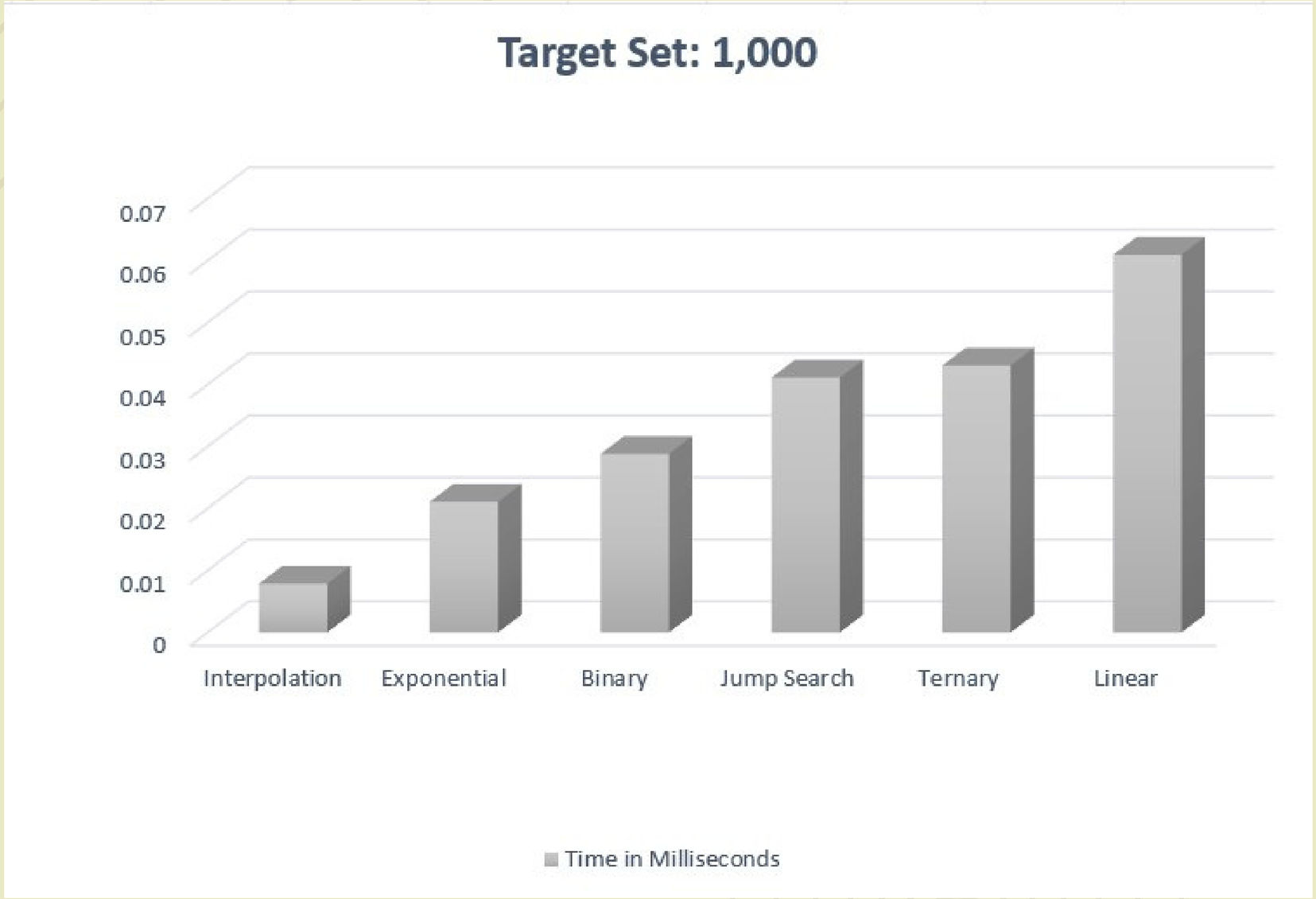
Target Set: 1,000
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Interpolation
Exponential
Binary
Jump Search
Ternary
Linear
Time in Milliseconds

Figure 3

LARGE DATA

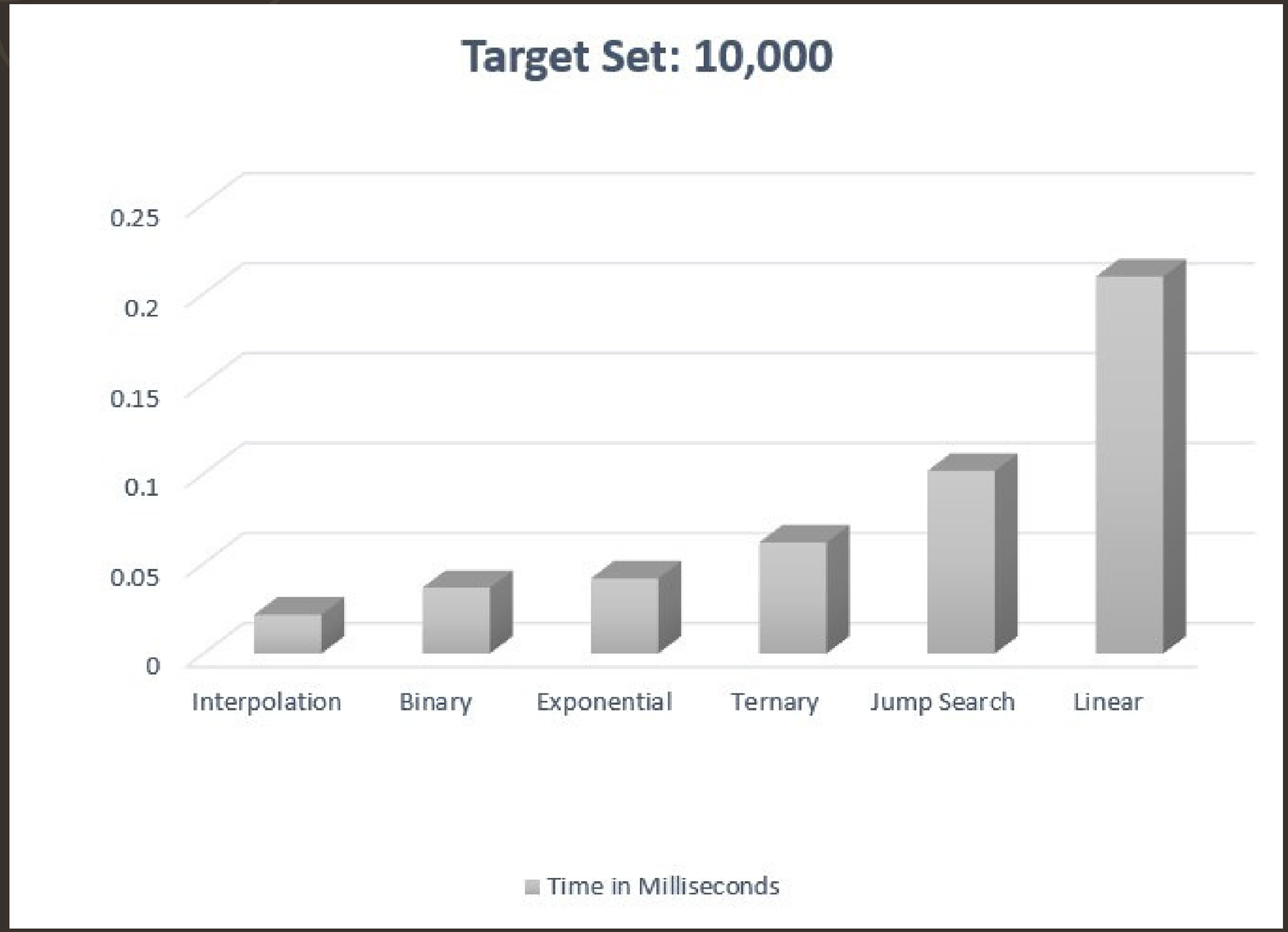

Figure 4

LAB 5 CODE

TO ACCESS THE CODES:

https://github.com/BillCipher22/Group_5_Lab_5.git

RESULTS

A. Which search algorithm performed the best overall?

Each algorithm has its strengths and weaknesses, and the best-performing algorithm can vary depending on the context. In this activity, looking at the graphs and table, it is clear that interpolation performed the best. In target set 100, linear and interpolation are tied for being the fastest two times each. In target set 1000, interpolation is the fastest with a 4/5 score, and in target set 10000, it earns 3/5 for the fastest. Interpolation Search has not only lived up to but exceeded expectations, earning its title as the fastest search algorithm. Its stunning performance, retrieving 9 out of 15 target data points with ease, solidifies its position as a standout solution for those seeking unparalleled speed and efficiency in their search operations.

RESULTS

B.

Did any search algorithms perform better on specific data sets?

Interpolation Search Algorithm performs the best in the 1-1000 dataset, where it attains a dazzling 4 out of 5 target ratio, showcasing its unparalleled speed in navigating the best performer among the 6 search algorithms. In the thrilling race of search algorithms, the Interpolation search algorithm emerges as the fastest among different search algorithms, consistently outpacing its rivals across 3 diverse datasets. With a stunning display of speed, it performed a 9 out of 15 target data, establishing itself as the fastest search algorithm.

RESULTS

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

The size of the data set can have a significant impact on the performance of search algorithms. Different search algorithms may exhibit varying degrees of efficiency, depending on the characteristics of the dataset. Linear search has a linear time complexity, so its performance is directly proportional to the size of the dataset. For the dataset of 100, it performed reasonably well. For larger datasets (1000, 10000), it becomes less efficient compared to other algorithms. As the size of the dataset increases, binary search remains relatively efficient, as it divides the search space in half with each comparison.

RESULTS

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

Jump search, on the other hand, is effective for larger datasets. With its square root time complexity, it strikes a balance between linear and binary search. It performed reasonably well for 100, became increasingly efficient for 1000, and remains efficient for 10000. Similar to binary search, exponential search is efficient for larger datasets. Its logarithmic time complexity makes it well-suited for all three dataset sizes (100, 1000, and 10000). Meanwhile, interpolation search performs well for uniformly distributed datasets. Its logarithmic log-log time complexity makes it efficient for all three dataset sizes. And lastly, ternary search divides the search space into three parts, which can be advantageous for larger datasets. Its logarithmic base-3 time complexity makes it efficient for all three dataset sizes.

CONCLUSION

D.

A brief conclusion summarizing your findings

The efficiency of various search algorithms was evaluated across three target sets: 1-100, 1-1,000, and 1-10,000. In the case of the 1-100 target set, the Interpolation algorithm was the fastest, having a speed of 0.02 milliseconds. Notably, Linear, Binary, Exponential, and Ternary algorithms exhibited marginal differences in their performance, while Jump Search algorithm lagged behind, nearly doubling the time required by Interpolation. Advancing to the 1-1,000 target set, Interpolation maintained its lead, completing the search in less than 0.01 milliseconds.

CONCLUSION

D.

A brief conclusion summarizing your findings

Exponential, Binary, Jump Search, and Ternary followed in descending order, with Linear search algorithm proving to be the least efficient. This finding persisted in the larger dataset of 1-10,000, where Interpolation continued to be the fastest with a speed of 0.02 milliseconds, outpacing Binary, Exponential, Ternary, and Jump Search. Linear search, once again, proved to be the slowest among the algorithms. In conclusion, the data analysis underscores the superior performance of the Interpolation algorithm across varying target sets, with consistent efficiency even in larger datasets.

Data Structures
and Algorithms

THANK YOU

SEARCH ALGORITHMS
BSCPE 2-6 | Group 5