Optimised Searching and Sorting:Modified Binary Search and Merge Sort Hybrid

Luke Louise Cobrado

University of Science and Technology of Southern Philippines, Cagayan de Oro City, Philippines

Email: Cobrado.lukelouise123@gmail.com

Kenjie Budlong

University of Science and Technology of Southern Philippines, Cagayan de Oro City, Philippines

Email: budlongkenjie06@gmail.com

Junar Landicho

University of Science and Technology of Southern Philippines, Cagayan de Oro City, Philippines

Email: junarlandicho@ustp.edu.ph

Abstract - Searching and sorting algorithms are an efficient tool in computer science that performs an important task in locating data in a vast data set or arranging data in a large data set. Binary search is popular searching algorithm for its efficiency and speed when searching data; however, data must first be sorted in order for the algorithm to work as intended. Merge sort is another popular algorithm when it comes to sorting due to its fast sorting speed when it comes to sorting larger data sets. This paper aims to combine both the searching and sorting algorithms, specifically the modified binary search that Ankit R. Chadha proposed in 2014, with the popular merge sort algorithm to eliminate the drawback of the modified binary search, which only works on sorted data. The researchers studied the execution time of the modified binary search with merge sort hybrid against the standard binary search with merge sort hybrid and found that the modified binary search with merge sort hybrid outperformed the standard binary search with merge sort hybrid.

Index terms - Binary search, modified binary search, merge sort, Modified binary search with merge sort hybrid.

I. INTRODUCTION

Given the increasing amount of vast data in the modern world, there is a need for the most efficient way to sort or search that data [1]. The searching and sorting algorithms play an important role in those parts; sorting involves arranging elements in a specific order, while searching aims to find a particular element within a collection of data. Merge sort and binary search are two classic algorithms used for sorting and searching, respectively [5].

The researchers specifically selected the binary search and the merge sort because of their high efficiency and suitability for sorting and searching large data sets. For the searching algorithms, binary search is recommended because it follows the 'divide and conquer' strategy. One of its biggest weaknesses is that the collection of data needs to be sorted in order for the algorithm to work as intended [2]. The merge sort is ideal for sorting large data because, similar to the binary search, the merge sort also uses the divide and conquer algorithm, which, among the other divide and conquer algorithms, is easier to understand [6].

One of the more commonly used sorting algorithms is the merge sort. Its efficiency and stability are part of the reason why it is popular. The idea of this algorithm is to divide the array into halves, repeating the process until every half has only one single element left; then, merge the halves back together, sorting the array in the process [2].

Binary search is one of the popular algorithms in today's generation. The technique it uses is the divide and conquer strategy, which is effective in locating its targeted element in a sorted array. The researchers used the study made by Ankit R. Chadha in 2014, The Modified Binary Search. The idea behind this algorithm is advantageous because it not only checks the middle index it also checks the first and last index of the array at every iteration [1].

The two main steps of the modified binary search and merge sort hybrid are sorting and searching. The first step is sorting. It sorts all the data from the unsorted data with the use of merge sort. The next step is searching with the use of the modified binary search, searching the targeted element in the array.

The study aims to create an effective algorithm using the combined effectiveness of merge sort and the efficiency of the modified binary search. The code application the researcher used was the C++ programming language. The study only focuses on sorting and searching in a one-dimensional array. The data is limited to numeric data only.

The study may contribute to the improvement of the sorting and searching algorithms by uniting the two efficient techniques. It may also show the potential for optimising sorting and searching operations in different applications. The combined approach offers improved efficiency and performance compared to individual algorithms.

II. RELATED LITERATURE REVIEW

The idea of a hybrid sorting and searching algorithm combines the strengths of multiple sorting and searching techniques to achieve desirable performance in various scenarios. Existing articles show the effectiveness of combining binary search and merge sort. A study conducted shows that the combined binary search and merge sort algorithm outperforms sequential and multithreaded approaches [2].

Binary search works on sorted arrays; it's considered a fast searching algorithm because the average case time complexity is O (log n). According to the study conducted by Parmar, when searching sorted data, binary search is recommended [9]. The divide-and-conquer method is used as the basis of the formula [4]. An effort to enhance the efficiency of the binary search has been conducted in an article [1]. To optimise the binary search algorithm's worst-case time complexity, the modified binary search is used by comparing the target element with the first and last element of the data set alongside the middle element at each recursion, thus optimising the binary search's worst-case scenario [1].

The study shows that the merge sort algorithm is better for a larger scale of data sets, as used in their experiment [5]. In mishal's research, merge sort also works better in larger data sets [7] [5]. There was a test conducted that observed the performance of multiple sorting algorithms, and found that merge sort has the smallest run time when the data set is less than 5000, and beyond that, merge sort has the smallest sorting time [8] [10].

III. EXPERIMENTAL AND EVALUATION RESULTS

The researchers performs a series of test comparisons to visualise the result of the proposed algorithm. There'll be a comparison between standard and modified binary search without the merge sort algorithm. And another comparison is made between the proposed modified binary search with the merge sort and the standard binary search with the merge sort. The comparisons are tested using the measure command to measure the speed of the execution time between the algorithms in random, unsorted data sizes. The algorithms without the presence of the merge sort would be tested similarly using the measure command, but on sorted random data sizes.

A.Measure-Command

One of the many PowerShell commands, the measure command, is used to calculate and return the execution time of the algorithm.

B.Data Generation

The data are generated randomly in an array of varying sizes of 100, 1000, 2500, 5000, and 10000 random integers without redundancy.

C.Algorithm Design

Below is the step by step implementation for the proposed algorithm that would be used to compare with the other algorithm:

Step 1: Declare functions and set up the merge sort, the modified binary search, and search-and-sort functions.

Step 2: Initialise the array and define the array of data.

Step 3: Define the target and set the target value to search.

Step 4: Divide the array and sort the array using merge sort.

Step 5: Merge the subarrays and combine the sorted subarrays.

Step 6: Find the target value using the modified binary search.

Step 7: Print the result.

Step 8: End the program.

D. Performance Evaluation and Results

The data shown in Table 1 indicates that the proposed algorithm, the modified binary search and merge sort hybrid, outperforms the traditional standard binary search with merge sort. This reveals that with increasing data size, the proposed algorithm performs more efficiently and scales better than the traditional approach, highlighting the potential benefits of using the algorithm in real world applications.

	Modified Binary Search	Standard Binary Search
Data Sizes	Run time (ms)	Run time (ms)
100	300	303
1000	304	306
2500	309	311
5000	325	317
10000	333	336

Table 1: Comparison between the modified binary search and the standard binary search, both with merge sort.

In Table 2, to test the run time of both algorithms without the merge sort, the researchers use a sorted data set to retrieve accurate results. The result shows that, for the algorithm alone, the proposed algorithm still outperforms the standard approach. This demonstrates that, similar to Table 1, the proposed algorithm performs efficiently even with increasing data size, showing the possible relevance and application of the algorithm, assuming that the data is sorted.

	Modified Binary Search	Standard Binary Search
Data Sizes	Run time (ms)	Run time (ms)
100	300	304
1000	302	301
2500	300	304
5000	311	313
10000	326	328

Table 2: Comparison between the modified binary search and standard binary search without the presence of the merge sort.

Based on the data shown in Table 3, the modified binary search outperforms the standard approach in both cases. The researchers conclude that the modified binary search is better than the standard approach.

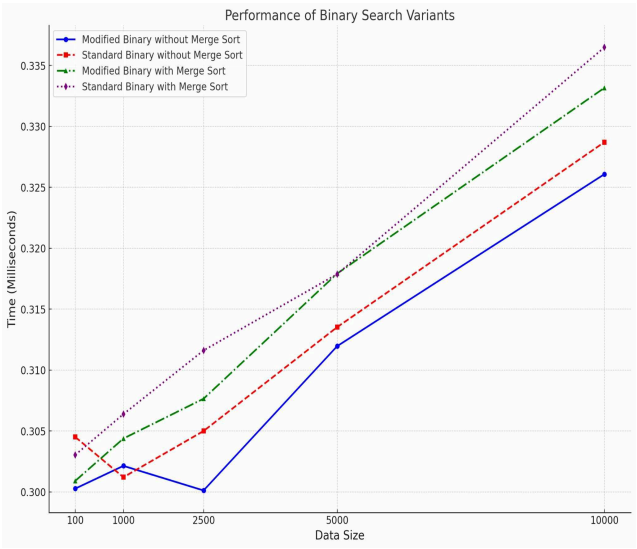

Table 3: Line graph comparison between the algorithms.

IV. CONCLUSIONS AND FUTURE WORK

The researchers concluded that the proposed algorithm, the modified binary search with merge sort hybrid, successfully demonstrated its efficiency and scalability over the traditional approach combined with merge sort. The rising execution time, along with the increasing data size, is normal, as larger data takes longer to sort. The proposed algorithm displays that it is viable in a real-world application, especially with large data sets, as it scales well with larger data. Further analysis and refinement are recommended to further solidify the efficiency, applicability, and performance of the algorithm.

In summary, while the modified binary search combined with merge sort successfully demonstrated its desired result, the concept of combining binary search and merge sort still showed a positive result as it eliminated the binary search's weakness of being that the data must be sorted first in order for the algorithm to work. Future work should focus on improving the speed and scalability of the modified binary search with merge sort to truly demonstrate the potential advantage of this approach.

References

[1] Chadha, A., Misal, R., & Mokashi, T. (2014). ISSN: 2249-0868 Foundation of Computer Science FCS. *International Journal of Applied Information Systems (IJAIS)*, *7*(2).

[2] Hijazi, S. W., & Qatawneh, M. (2017). Study of Performance Evaluation of Binary Search on Merge Sorted Array Using Different Strategies. *International Journal of Modern Education and Computer Science*, *9*(12), 1.

[3] Rahim, R., Nurarif, S., Ramadhan, M., Aisyah, S., & Purba, W. (2017). Comparison Searching Process of Linear, Binary, and Interpolation Algorithms. *Journal of Physics: Conference Series*, *930*, 012007.

[4] Harun, M., & Imtiaz, A. (2023). Paw Search: A Searching Approach for Unsorted Data Combining with Binary Search and Merge Sort Algorithm. *International Journal of Advanced Computer Science and Applications*, *14*(2).

[5] Rabiu, A. M., Garba, E. J., Baha, B. Y., & Mukhtar, M. I. (2022). Comparative Analysis between Selection Sort and Merge Sort Algorithms. *Nigerian Journal of Basic and Applied Sciences*, *29*(1), 43–48.

[6] Lobo, J., & Kuwelkar, S. (2020, July 1). *Performance Analysis of Merge Sort Algorithms*. IEEE Xplore.

[7] Mishal, I., AL-Khatib, R., & Hiasat, R. (2021). Comparative Study of Two Divide and Conquer Sorting Algorithms: Modified Quick Sort and Merge Sort. *International Journal of Computer Applications*, *183*(31), 28–33.

[8] Olabiyisi S.O, Adetunji A.B, & Oyeyinka F.I. (2024). An Evaluation of the Critical Factors Affecting the Efficiency of Some Sorting Techniques. *International Journal of Modern Education and Computer Science*, *5*(2), 25.

[9] P. Parmar, V., & Kumbharana, C. (2015). Comparing Linear Search and Binary Search Algorithms to Search an Element from a Linear List Implemented through Static Array, Dynamic Array and Linked List. *International Journal of Computer Applications*, *121*(3), 13–17.

[10] Taiwo, O. E., Christianah, A. O., Oluwatobi, A. N., Aderonke, K. A., & kehinde, A. J. (2020). COMPARATIVE STUDY OF TWO DIVIDE AND CONQUER SORTING ALGORITHMS: QUICKSORT AND MERGESORT. Procedia Computer Science, 171, 2532–2540.

Author's Profiles

Luke Louise Cobrado, August 26, 2003, in Cagayan de Oro City. Is a second-year student pursuing a Bachelor of Science in Computer Science in the Philippines at the University of Science and Technology of Southern Philippines in Cagayan de Oro City.

Kenjie Budlong, October 06, 2002, in Cagayan de Oro City. Is a second-year student pursuing a Bachelor of Science in Computer Science in the Philippines at the University of Science and Technology of Southern Philippines in Cagayan de Oro City.

Junar Arciete Landicho, received PhD in Information Management from the Asian Institute of Technology, Thailand. His research interest is in the field of database system, e-commerce, mobile application and healthcare informatics.