PA5 Report

By

William Box

CSI 281
Data Structures & Algorithms
Fall 2022-23

1. Introduction

For this assignment, I tested an implementation of binary search on a doubly linked list, and compared it against a standard search algorithm.

2. Background

Provide information on each of the algorithms on their characteristics, advantages and disadvantages. Also, provide information on their performance analysis.

The binary search algorithm is supposed to speed up search performance by reducing the possible location of the search key by half every. However, this algorithm was written with the assumption that one can access an arbitrary location within an array, something that is impossible with a doubly linked list.

The standard search algorithm in comparison, simply iterates through the list until it either finds the target or it hits the end of the doubly linked list.

3. Implementation Detail

The binary search algorithm was a fairly standard version of the algorithm, with the smallest possible modifications necessary to make it work with doubly-linked-lists.

The standard search algorithm simply goes through the list until it hits the target or hits the end of the list.

4. Experimentation Detail

The experiment was ran on a Windows Surface Book 3 with the following specs

a. 32 GB Total physical memory, 19.9 GB Available physical memory
b. Intel Core i7-1065G7, 4 core processor
c. 1.30 GHz
d. System type: 64 bits

Summary Data

In order to help your reader digest information, you must provide the following summary data in this section using the following table format:

Algorithm: Standard	
Target Location	**Average Time**
0 to 9	8.00E-08
990 to 999	9.22E-06
495 to 504	3.33E-06
Out of Bounds	2.91E-04

Algorithm: Binary Search	
Target Location	**Average Time**
0 to 9	8.24E-04
990 to 999	2.21E-04
495 to 504	2.25E-04
Out of Bounds	1.46E-04

5. Discussion and Conclusion

With the data I have collected, it would appear that the binary search algorithm was several orders of a magnitude slower than the standard search algorithm in almost every case, with the only exception being when looking for objects that were not in the list. This is expected, as the Binary Search algorithm essentially has to iterate through the list multiple times in order to perform its search, while the standard algorithm only has to iterate through once. As for personal preference, I like the standard algorithm more here, because it’s both faster and easier to code.

6. References

Provide the detailed references you used for this report. You must cite all references you used throughout your report using the IEEE standard.

7. Appendix

You must have one table for each algorithm to report on the different run-time requirements.

Algorithm: Binary Search										Average
0 to 9	8.24E-04	8.24E-04	8.24E-04	8.24E-04	8.24E-04	8.24E-04	8.24E-04	8.24E-04	8.24E-04	8.24E-04
990 to 999	2.21E-04	2.21E-04	2.21E-04	2.21E-04	2.21E-04	2.21E-04	2.21E-04	2.21E-04	2.21E-04	2.21E-04
495 to 504	2.25E-04	2.25E-04	2.25E-04	2.25E-04	2.25E-04	2.25E-04	2.25E-04	2.25E-04	2.25E-04	2.25E-04
Out of Bounds	1.46E-04	1.46E-04	1.46E-04	1.46E-04	1.46E-04	1.46E-04	1.46E-04	1.46E-04	1.46E-04	1.46E-04

Algorithm: Standard Search										Average
0 to 9	2.00E-07	2.00E-07	2.00E-07	2.00E-07	2.00E-07	2.00E-07	2.00E-07	2.00E-07	2.00E-07	8.00E-08
990 to 999	9.10E-06	9.10E-06	9.10E-06	9.10E-06	9.10E-06	9.10E-06	9.10E-06	9.10E-06	9.10E-06	9.22E-06
495 to 504	3.40E-06	3.40E-06	3.40E-06	3.40E-06	3.40E-06	3.40E-06	3.40E-06	3.40E-06	3.40E-06	3.33E-06
Out of Bounds	8.96E-04	8.96E-04	8.96E-04	8.96E-04	8.96E-04	8.96E-04	8.96E-04	8.96E-04	8.96E-04	2.91E-04