Aidan Stoner and Hannah Fitzgerald

Data Structures and Algorithms

Adrian Rusu

4/17/2024

Lab 3

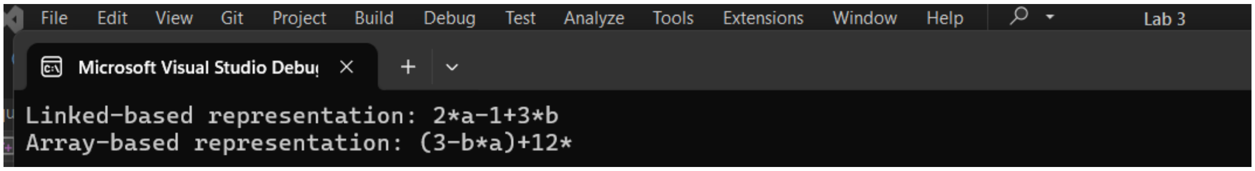
Introduction

Lab 3 centers the focus around Binary Trees, General Trees, and HeapSort. This lab will help reinforce what the students have previously learned through the lectures. Within the lab, there are various prompts including printing and evaluating arithmetic expressions, implementing the preorder traversal, writing a tic-tac-toe game efficiently, and implementing an array-based HeapSort. These prompts required the use of linked-representations, array-representations, Binary Trees, and General Trees. A linked representation utilizes a doubly linked list to represent the nodes of the tree, while the array representation is using an array to represent the nodes of the tree. The difference between a tree and binary tree is that a general tree can have multiple children or zero children, while a binary tree can have a maximum of two children. This lab gives students a technical view of what working with specific data structures can look like while trying to accomplish a certain goal.

Print of Arithmetic Expressions

The goal of this program was to print the equation ((2x(a-1)+(3xb)) in an array representation and a linked-based representation. When using the different representations, the equation prints differently due to the different watts that they are represented on the tree.

However, the results for this lab align with the representation they are programmed with. Below, the output shows the results of the compiled code.

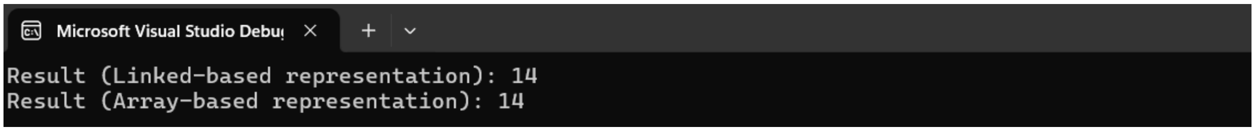


```
Linked-based representation: 2*a-1+3*b
Array-based representation: (3-b*a)+12*
```

Figure 1: Results of Array-based and Linked-Based Representation Print

Evaluation of Arithmetic Expressions

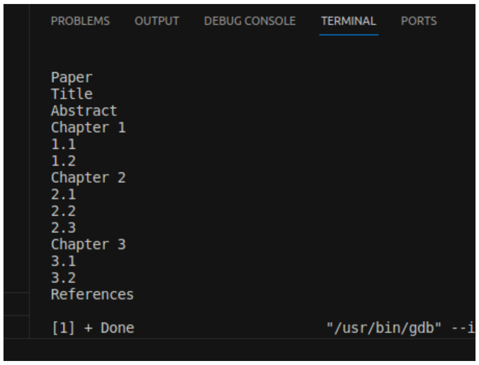
The objective of this problem was to then take the representations from the previous problem, and evaluate the expression ((2(5-1))+(3x2). Ultimately, the answer to this equation is 14. However, there is a difference on how each of these algorithms solves these problems. An array based tree assigns an index, and the way the problem is solved is depending on how the nodes are indexed. However, the linked-based representation involves recursive functions to solve the problems. Below, you can see that the code solves the equation for both respective prompts.



```
Result (Linked-based representation): 14
Result (Array-based representation): 14
```

Figure 2: Results of Array-based and Linked-Based Representation Evaluation

Preorder Traversal



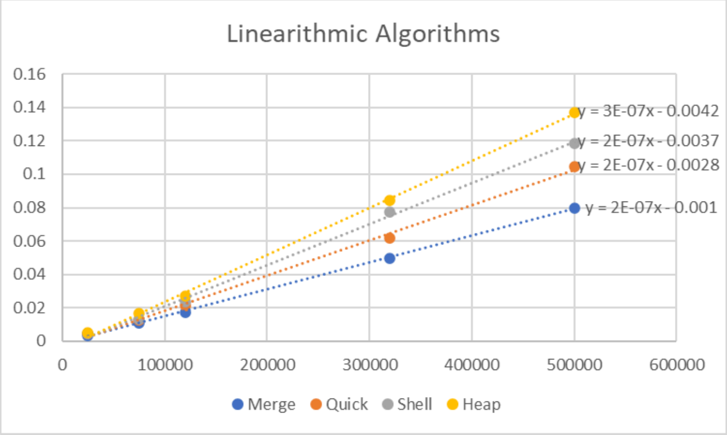
```
Paper
Title
Abstract
Chapter 1
1.1
1.2
Chapter 2
2.1
2.2
2.3
Chapter 3
3.1
3.2
References

[1] + Done                    "/usr/bin/gdb" --i
```

Tic-Tac-Toe

Video in zip file

Array-Based Heapsort

Heapsort is a sorting algorithm that takes an array, turns it into a heap data-structure. In the heap data structure the highest or lowest number, depending on the type of heap, in the array will be at the root. The root is then moved to the end of the array and the process repeats but now excludes the previous root. Once all the roots are added to the end of the array it is completely sorted. The heap sort algorithm was compared to multiple other sorting algorithms. It was found to be the fastest, it was able to sort 500,000 distinct numbers in less than .14s, as seen in the graph below.



Conclusion

To conclude, lab 3 involved implementing trees, binary trees, heapsort, and different representations in order to complete this assignment. This helped our team strengthen our coding skills when dealing with these algorithms. The prompts were straightforward, and bore a similarity to a few of the previous labs we have completed. From printing and evaluating arithmetic expressions, to creating a more efficient Tic-Tac-Toe, it allowed for our team to gain knowledge on these different problems that require solutions.