John Hopkins University

605.202 Data Structures

Obadiah K Njoroge

Lab 3 Analysis

11/16/2021

Introduction

The aim of his lab is to build an advanced form of Solitaire game, called Scorpion. The game starts with 49 cards that are delt in 7 columns on each of the tableau. There is a mixture of cards that are dealt face up and other dealt face down. Each face card is available to play and the aim is to build card set down the suit from king to ace. If a card is moved to a different column, the cards on top are also moved as well. The last card should be the ace to complete the suit and move it to the foundation.

A linked list was used in the implementation of the scorpion Solitaire game. A linked list is a linear data structure that has interconnected nodes. Like most data structures, linked list needs to have a starting point and a termination point. The starting point or node of a linked list is referred to as a head node, while the ending node is known as a tail node.

Justification of using linked list

Linked list made for a good choice for this lab because of their inherent benefit of breaking and rejoining between nodes. Unlike arrays that store their elements contiguously, and therefore insertions and deletion require shifting of the elements, Linked list as a data structure improves on arrays by shift the current pointer during insertions and deletions. I took advantage of this to shift cards around from one column to another based on the scorpion rules.

Time complexity

Time complexity is the total amount of time that an algorithm or logical statement takes to run through a series of logical steps. Time complexity measures the efficiency of the code by assessing its performance. Big O looks at the relationship between the inputs that are presented to an algorithm and the subsequent outputs. In our case, a file containing a series of card sets that we required to use the algorithm to play a game of solitaire scorpion.

Linked list use a sequential access to find elements, therefore to access an element needs a traversal of the linked list which has time complexity of O(n) unlike that of an array which is constant time because it supports random access.

Design Justification

The implementation of this lab was not easy because of the number of moving elements that needed to be accounted for at any given time. Linked list and the custom functions that were created to support it made it easier to manage the tasks needed to play the game.

The choice of a linked list was better than using an array because it enabled an easier shifting of elements when needed. When looping through the array, an iterative solution was used, however a recursive solution would have been optimal because of the repeated nature of the task, however that approach created some undo complexity in this lab. Given more time, I would switch from an iterative solution to a recursive one.

What I might do different next time

There is no doubt that this was a challenging lab to get through. Some of the challenges were to create an algorithm that would take a string of cards, split them into a tableau set, foundation set and waste set. Then used a linked list to hold the pairs of cards with custom functions to switch the cards around.

The one element that I would change is time management. I was late submitting this lab because of lab was more complicated that previously anticipated and therefore I was unable to complete the lab on time.

Given more time I would have created a better implementation that would play much smarter by adding more functions that check if the moves were correct. Also I would create more test cases and better error handling.

There are more functionality that I would have liked to implement but time was limited. I will however continue with the implementation for my records.

Improvements/Enhancements

Added an array implementation of a linked list to hold the tableau seven column cards and output them as such.

References

“Difference between Array and Linked List.” *Studytonight.com*, https://www.studytonight.com/data-structures/linked-list-vs-array.