Keith Beauvais

CS 202

Assignment 9 Comprehension Questions

Comprehension Questions:

1. The SwapCards() function is supposed to only swap the next pointers and not the card within the node. This may be annoying to code, but can you explain an example of where swapping the next pointers will be significantly better than swapping the data within the node?

In situations where a lot of data is handled, this would allow for faster swapping time. Instead of iterating through an extensive list and then sorting, this method enables two specific nodes to be swapped, saving time. Also, if you were to swap data instead of the node, there would be no benefit of using a linked list.

1. In the SortCards function description, I mention you can use a sorting method that doesn’t call swap cards at all. Can you describe a very simple sorting algorithm using linked lists that does not require SwapCards? Hint: You will need the unsorted linked list and an empty linked list.

If I were not to use the SwapCards function, I would probably use an insertion sort. This algorithm would allow me to use the unsorted list and place the data in the correct order in an empty linked list.

1. This card game implementation benefits heavily from using linked lists, as opposed to using dynamically allocated arrays using pointers. Why do you think that is?

Linked lists have their benefits over other dynamic lists. Using it in a card game is beneficial because there is a lot of insertion and deleting in the piles and hands. In a vector or array, we would have to resize the structure, making it inefficient. Also, a linked list does not waste as much memory as an array; it does take up more memory because each node has a pointer but could be offset with its efficiency.