CSDS 233 Spring Session 2

SI Leader: Jakob Danninger

1/31/2023

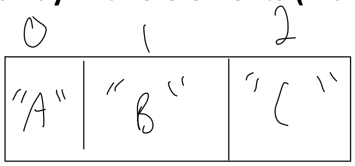
Disclosure: This is a supplement to class, not a replacement. This should not be your only study activity for exams, it should aid you in studying. I do not have the actual exam so questions here will differ from those on the exam.

Session Objectives:

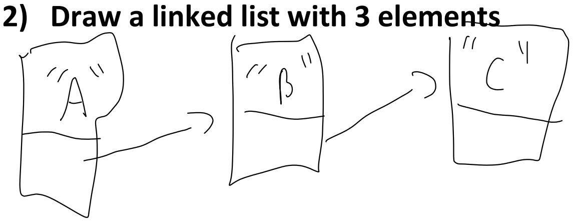
1) Explain what a linked list is (including how to add remove and search
2) Understand how computer memory works and how that related to array and linked list
3) Decide when to use a list vs array

Practice Problems:

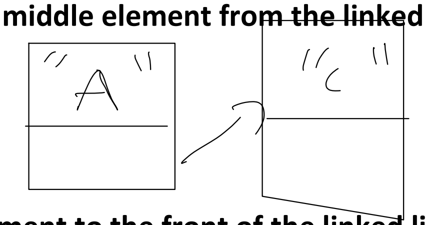
1) Draw an array with 3 elements (including index)



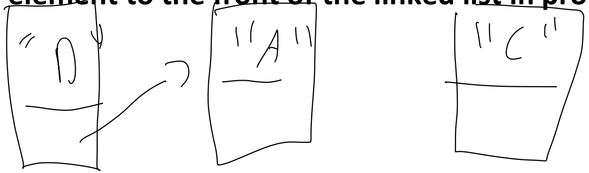
2) Draw a linked list with 3 elements



3) Delete the middle element from the linked list in problem 2



4) Add an element to the front of the linked list in problem 2



5) Complete the following definitions

a. Pointer

points to next
node (behind the
scenes it has data
adress)

b. Header

points to first node

6) What is the max size of a linked list? What about an array?

however large memory allows

whatever was predetermined when initializing array

7) What two pieces of information does every node contain?

data
pointer

8) Scenario: you are creating a searchable business receipts program, would you use a linked list or array ? Justify why (There is no right answer this question is about your justification)?

If you argued you need fast searching then organized array.

9) Scenario: you are creating an archive of customers in alphabetic order, would you use a linked list or array? Justify why (There is no right answer this question is about your justification)?

linked list since easy to add to middle of list

10) Complete the table

	Array	Linked List (single linked)
Access an item	O(1)	O(n)
Search for an item	O(n)	O(n)
Add item to front	O(n)	O(n)
Add item to back	O(1)	O(1)
Delete value from middle	O(n)	O(n)

11) What are the benefits of using a Linked List over an array? (there are 2 big ones)

↳ size isn't limited

↳ easy to delete or add middle node