Overview

Monday, September 16, 2019 12:34 PM

Internship fair tomorrow!

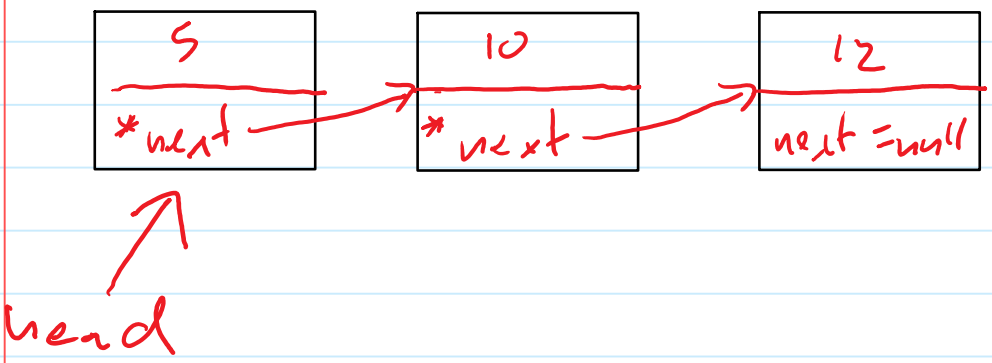
This Week:

- Assignment 3 - Linked List
- New Topics: Linked List implementation
- Reminder: Quiz due tonight

Today:

- Linked List:
  - Recap: linked list
  - the Abstract Data Type (ADT)
  - search method
  - insert method

Linked List recap:



- a data structure that allows individual elements to be added and deleted as needed.
- typically, allocated dynamically
- memory is allocated and deallocated for each individual element as it is added or removed
- each element:
  - called a node

    - node contains a key value(s) a pointer to next node
- access the LL via the head pointer
- the end of a LL is denoted by a next = nullptr

Abstract Data Type (ADT)

- A collection of member data and the allowed operations on that data.
- Abstract, because the user (i.e. the programmer) only has info about the input(s) the output(s) and a explanation of actions.
- Can think of it as a pseudo-code for a class defnition
- Not language specific.

Singly Linked List Generic ADT:

private:

head - *ptr to first element in list. Null means empty list.*

public:

intialize() - *set header to null*

nodePtr = search(value) - *find a value and return ptr to its node*

insertNode(previousPtr, newValue) - *given a ptr to a node, insert a new node right after it*

displayList() - *starting at head, traverse the list and display all keys*

deleteNode(nodePtr) - *given a ptr to a node, remove node from the LL*

deleteList() - *deallocate the entire least*

Rule to keep in mind:

Every method in the ADT should be designed such that once

it perfoms its task, the integrity of the data structure is preserved.

example violation: delete a node and forget to reconnect the pointers properly.

Implementation in C++

Where to start?

Constructor definition? - *gets called when the object is instantiated, so not bad place to start*

We want to start building the list, so displayList next? sure

1. constructor - *test: compile*
2. search - *test: compile; call search() from driver with some test string. Expect is a null ptr.*
3. display - test: *compile*
4. insert
    I. empty list case - *test: insert node, display*
    II. insert before head - *test: insert node, display*
    III. insert anywhere else - *test: insert node, display*