CS112 Lab 03, Feb 4, 7 2010

http://cs-people.bu.edu/deht/cs112_spring11/lab03/

Diane H. Theriault

deht@cs.bu.edu

http://cs-people.bu.edu/deht/

Getting Comfortable with Linked Data Structures

- Last time, we talked about how to use Collections and Iterators

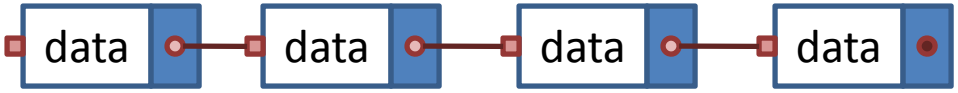
- Now, you will get some experience manipulating these types of data structures yourself

No Indexes!

- Indexes are not meaningful in linked data structures.
- You will need to use references to move around.
- The enqueue method provides some hints.

Linked List Concept

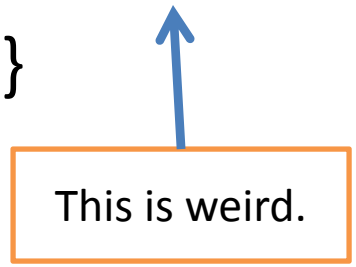
- You have a set of boxes with rope tied to them.
  - Your data is in the boxes.
  - Each rope is tied to the next box.
  - "Node" is the most common name for the boxes that make up a linked list. (But this name is not special)



- You need to keep a reference to the first box (head).
- Sometimes a reference to the last box (tail) is useful.

“Node” Implementation

```
class Node{
    int mData;
    Node mNext;
}
```

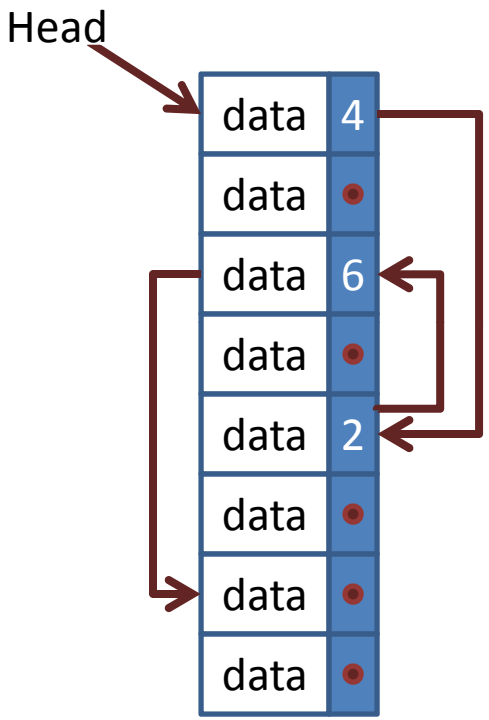


- Why does the “Node” class contain a member of type “Node”?
- It is just a reference.

Node Implementation

- Imagine if you had an array of nodes
- And mNext was an integer index into the array

```
class Node{
    int mData;
    int mNext;
}
```



To traverse the list, just start at the head and follow the indexes.

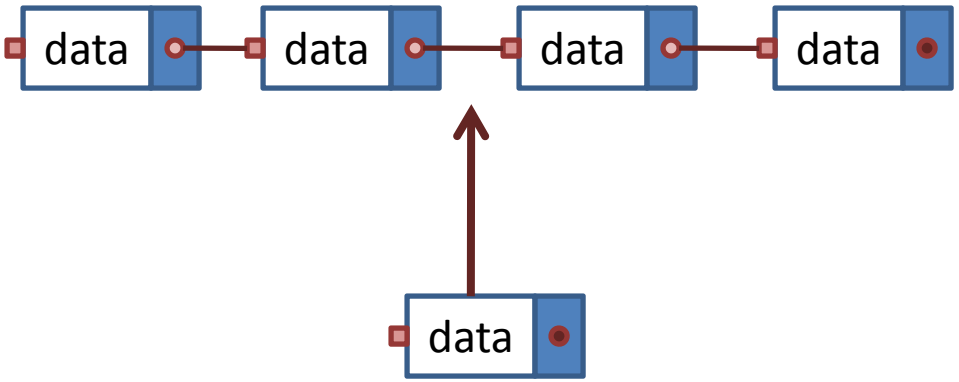
LinkedQueue Implementation

- I have provided a very simplistic implementation of a linked queue.

- You will write a print() method to move through the list and print all of the elements.

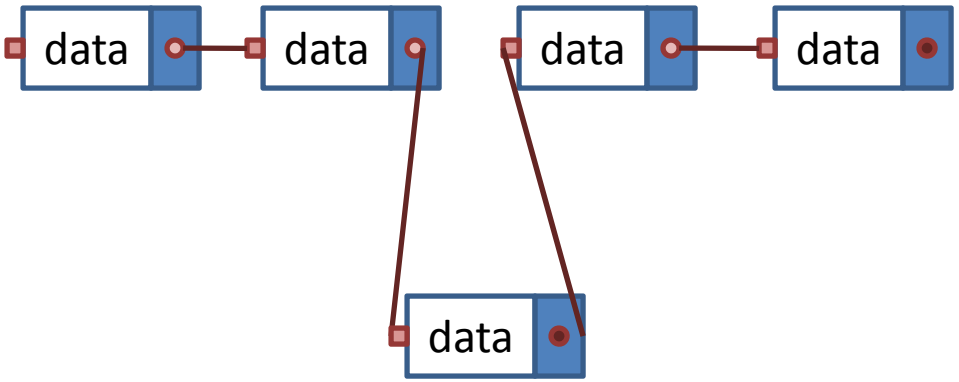
Manipulating a Linked List

- Need to change mNext references so that no one gets lost, and your state matches your picture.



Manipulating a Linked List

- Need to change mNext references so that no one gets lost, and your state matches your picture.



Priority Queue Implementation

- Now, you will modify the enqueue method to insert elements into the list in sorted order. (smallest to largest)

- Make sure you handle all of your edge cases.

- Use the print method to verify that the elements were inserted correctly.

Priority Queue Implementation

- 4 cases to handle when inserting:
  - The list is empty
  - New value is less than the head
  - New value is larger than the tail
  - Inserting into the middle of the list somewhere