

Lecture 6: LinkedLists

07/06/22



Announcements

- Checkpoint 2 due Wednesday, July 6th @ 11:59pm
- A2: GuitarHero due Thursday, July 7th @ 11:59pm

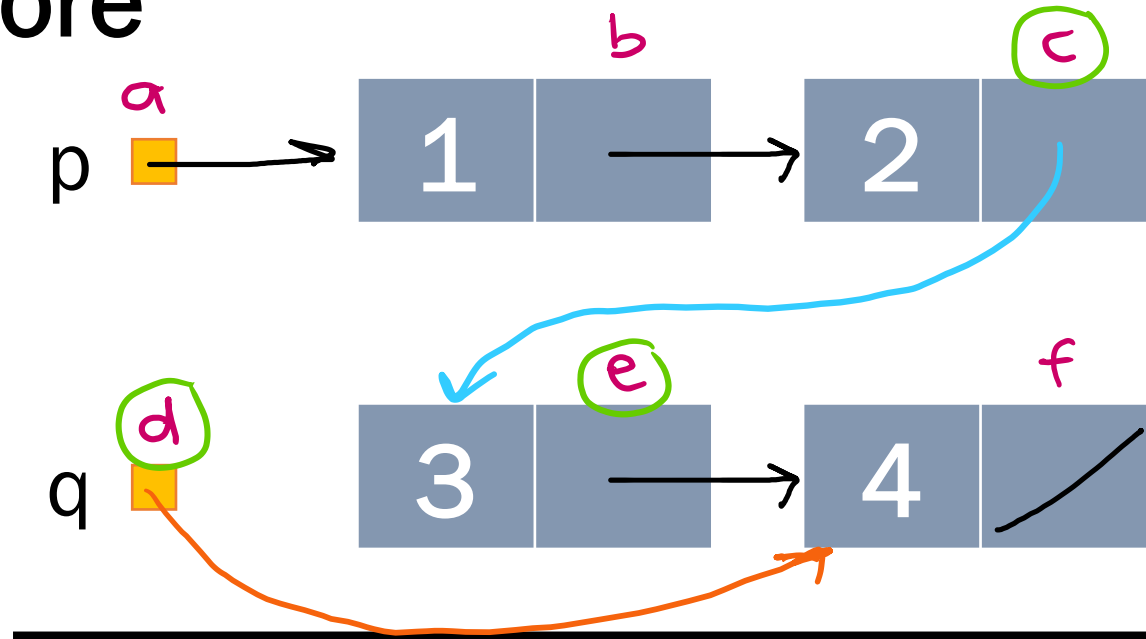
```
// ListNode is a class for storing a single node of a linked list
// storing integer values.
public class ListNode {
    public int data;          // data stored in this node
    public ListNode next;     // link to next node in the list

    // post: constructs a node with data 0 and null link
    public ListNode() {
        this(0, null);
    }

    // post: constructs a node with given data and null link
    public ListNode(int data) {
        this(data, null);
    }

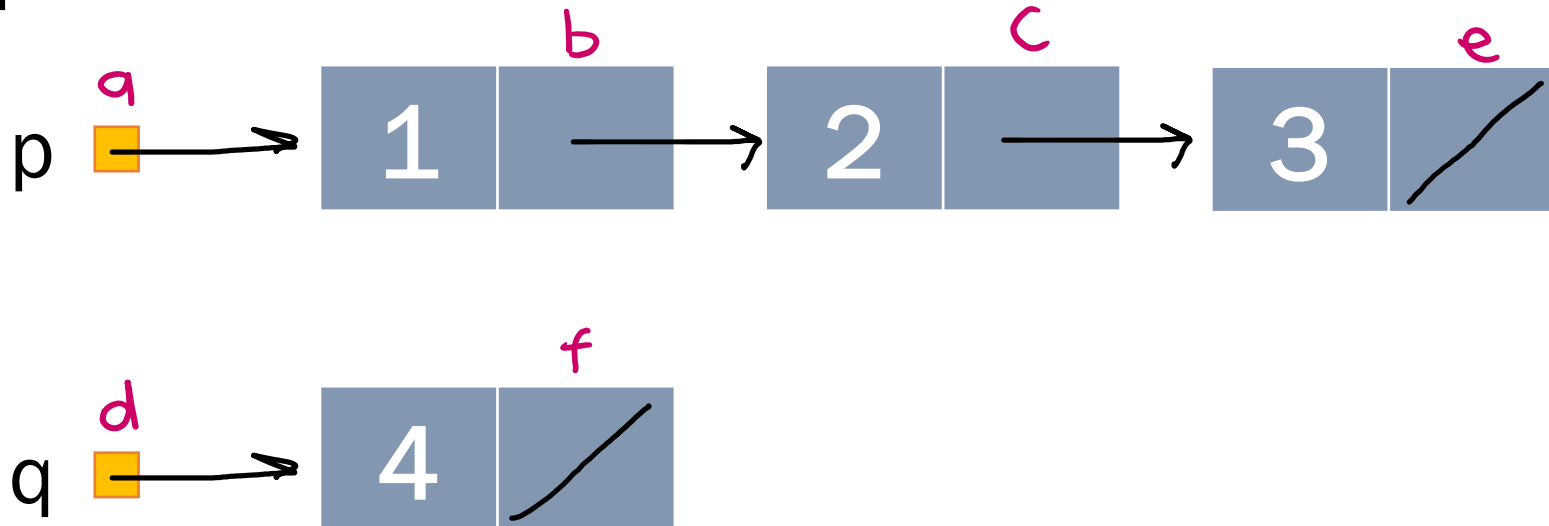
    // post: constructs a node with given data and given link
    public ListNode(int data, ListNode next) {
        this.data = data;
        this.next = next;
    }
}
```

Before

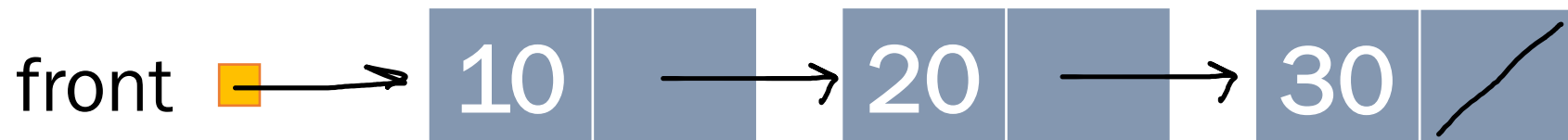


$p.\text{next}.\text{next} = q;$
 $q = q.\text{next};$
 $p.\text{next}.\text{next}.\text{next} = \text{null};$

After



```
public class ListNodeClient {  
    public static void main(String[] args) {  
        ListNode front = new ListNode(10);  
        front.next = new ListNode(20);  
        front.next.next = new ListNode(30);  
  
        System.out.println(front.data);  
        System.out.println(front.next.data);  
        System.out.println(front.next.next.data);  
    }  
}
```



Print Pseudocode

Start at the **front** of the list.

```
while (there are more nodes to print):
```

```
    Print the current node's data.
```

```
    Go to the next node.
```

- How do we walk through the nodes of the list?

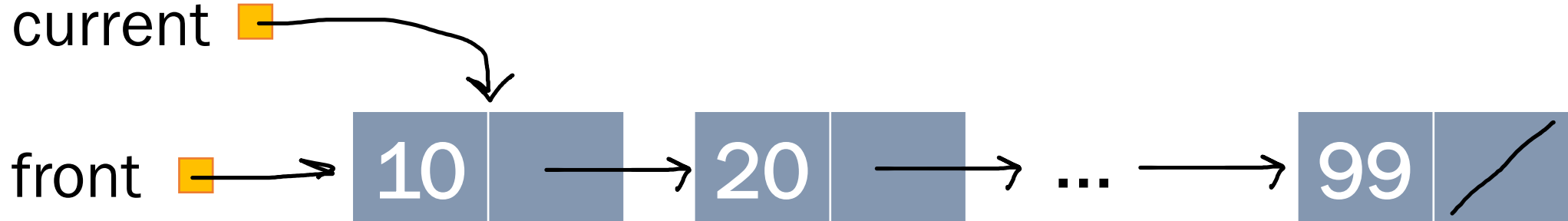
```
front = front.next;    // is this a good idea?
```



A current reference

- Don't change `front`. Make another variable, and change it.
 - A `ListNode` variable is NOT a `ListNode` object

```
ListNode current = front;
```



- What happens to the picture above when we write:

```
current = current.next;
```

Linked List vs. Array

- Print list values:

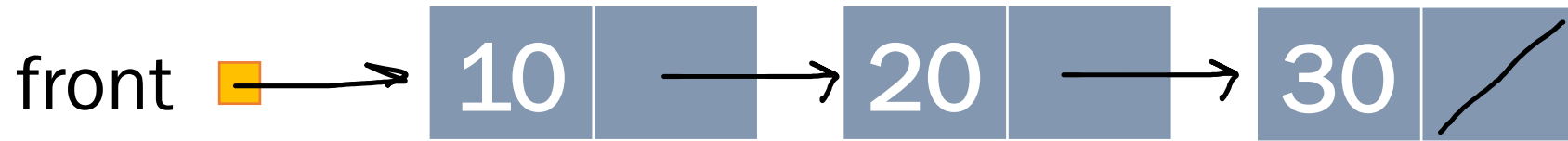
```
ListNode front = ...;
ListNode current = front;
while (current != null) {
    System.out.println(current.data);
    current = current.next;
}
```

- Similar to array code:

```
int[] a = ...;
int i = 0;
while (i < a.length) {
    System.out.println(a[i]);
    i = i + 1;
}
```

Description	Linked List Code	Array Code
Go to front of list	<code>ListNode current = front;</code>	<code>int i = 0;</code>
Test for more elements	<code>current != null</code>	<code>i < size</code>
Current value	<code>current.data</code>	<code>elementData[i]</code>
Go to next element	<code>current = current.next;</code>	<code>i = i + 1;</code>

- Suppose our list had the contents



- Practice simulating the code we wrote and tell us what the result will look like when we call `list.add(40)` ;

```
public void add(int value) {  
    ListNode current = front;  
    while (current != null) {  
        current = current.next;  
    }  
    current = new ListNode(value);  
}
```