EASTERN WASHINGTON UNIVERSITY
start something big

Circular and Doubly Linked List

Computer Science Department
Eastern Washington University
Yun Tian (Tony) Ph.D.

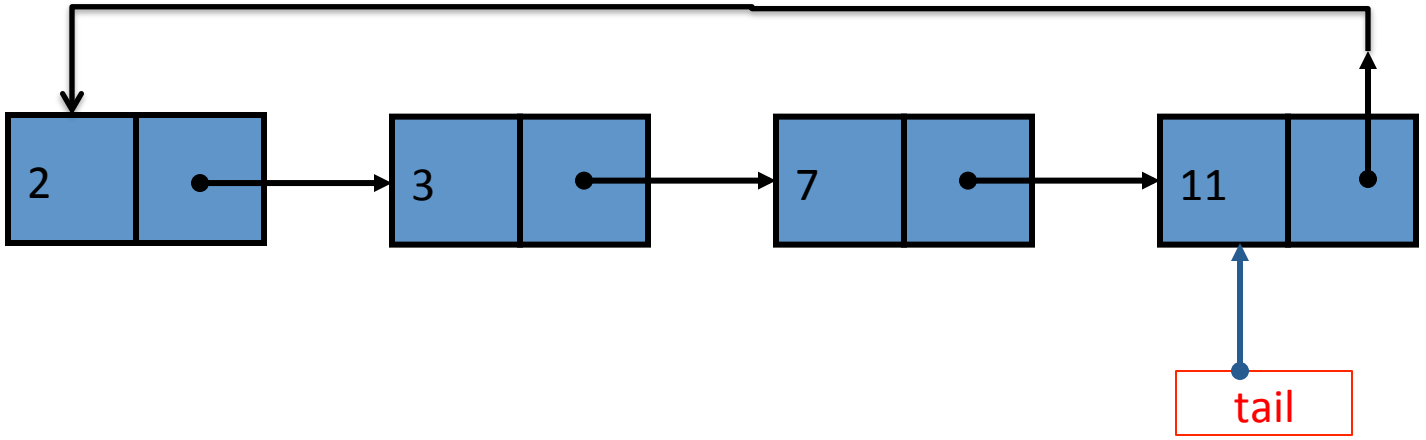
Review

- addSorted() method on Linked List
- Review Selection Sort on Array
- Selection Sort on Linked List with Dummy Node

Today

- Circular Singly Linked List
- Circular Doubly Linked List

Circular Singly Linked List



1) addFirst() and addLast() takes constant time O(1), which is very convenient.
2) The Circular Linked List above in diagram has no dummy node.

addFirst() on Circular Linked List


```
public void addFirst(E value)
{
    Node<E> temp = new Node<E>(value);
    if (this.tail == null) { // first value added, edge case
       this.tail = temp;
       this.tail.next = tail;
    } else { //  list is not empty before
       temp.next = tail.next;   //step 1
       this.tail.next = temp;    //step 2
    }
    this.size++;
}
```

//when rewire the links, the order of the these step1 and step2 matters?

//what happen if we do step2 then step1 in method above?

addLast() on Circular Linked List


```
public void addLast(E value)
{
    // new entry
    addFirst(value);
    tail = tail.next();
}
```

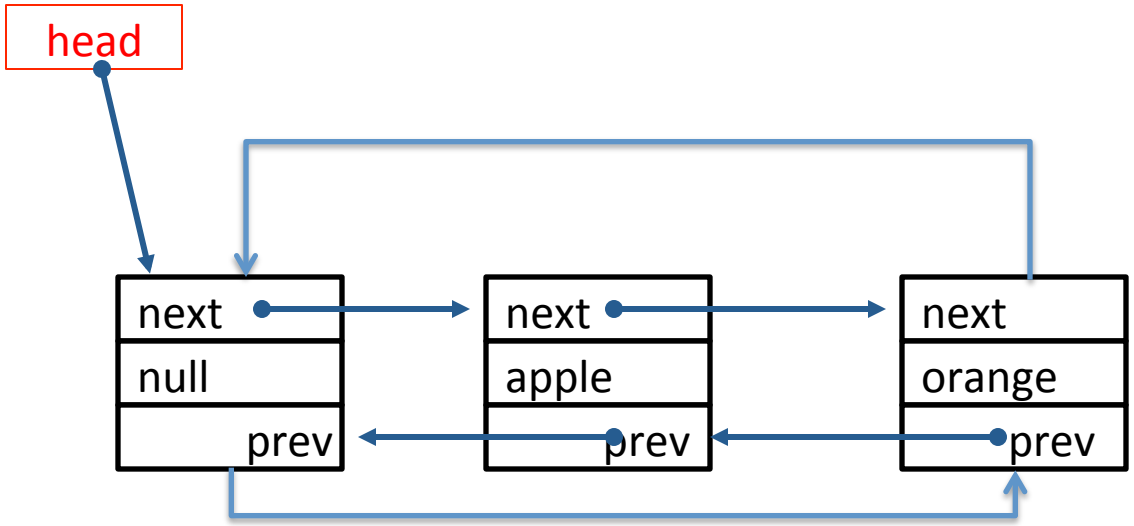
how do we know we reaching the end of a circular list, when Traversing all the nodes in the list?

toString() on Circular Linked List

EASTERN WASHINGTON UNIVERSITY
start something big

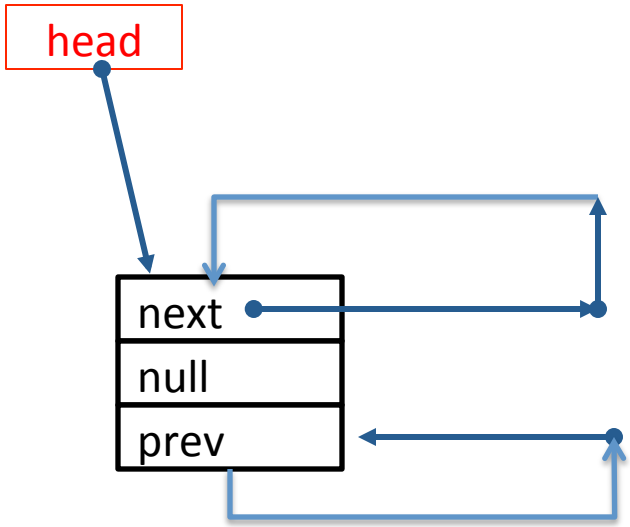
```
public String toString() {

        String result = "";
        if this.size <= 0
                return result;
        Node cur = this.tail.next;
        while( cur!= tail ) {
                result += cur.data + "\n",
                cur=cur.next;
        }
        result += cur.data + "\n" ;
}
```

Circular Doubly Linked List



1) In each node, we have three fields. **next** points the successor of current node. **prev** points to the predecessor of the current node. The last field is the data field.
2) This circular doubly linked list has a dummy node.
3) Could you imagine what an empty circular doubly linked list with a dummy node looks like?

Circular Doubly Linked List



Empty Doubly Linked List

Add Method

```
public void add( Object data, int index) {
        if(index < 0 || index > this.size || data ==null)
                throw new IllegalArgumentException("Message error");
        Node cur;
        int i;
        for( i = 0, cur = this.head; i < index; i ++ ) {
                cur = cur.next;
        }
        Node newNode = new Node(data, ____cur____, ____cur.next____); //step1
        ____cur.next.prev____= newNode;    //step2
        ____cur.next_____ = newNode; //step3
        this.size ++;
        //the order of step1 and step2, step3 matters,
        // we can not switch with the following command
}
//principle of rewire the links:
//1) First assign values to the links in the newNode.
//2) then assign newNode to its successor's prev link
//3) then assign newNode to its predecessor's next link
```

Remove Method

```
public boolean remove( int index ) {

        if(index < 0 || index >= this.size)
                throw new IllegalArgumentException("The index parameter is out of bound!");
        Node cur = this.head.next;
        int i = 0;
        while( cur != this.head && i < index ) {
                cur =  cur.next;
                i ++;
        }
        if( cur == head )  //not found data in list
                return false;
        __cur.prev.next__ = cur.next;
        __cur.next.prev__ = cur.prev;
        this.size --;
        return true;
}
```

Remove Method

```
public boolean remove( Object data ) {

        Node cur = this.head.next;
        while( cur!=this.head && ! cur.data.equals(data) ) {
                cur =  cur.next;
        }
        if( cur == head )  //not found data in list
                return false;
        cur.prev.next = cur.next;
        cur.next.prev = cur.prev;
        this.size --;
        return true;
}
```

Summary

- Circular Singly Linked List
- Circular Doubly Linked List

Next class

- addOrdered() on circular doubly linked list
- Selection sort on linked list