EASTERN WASHINGTON UNIVERSITY
start something big

Circular and Doubly Linked List

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

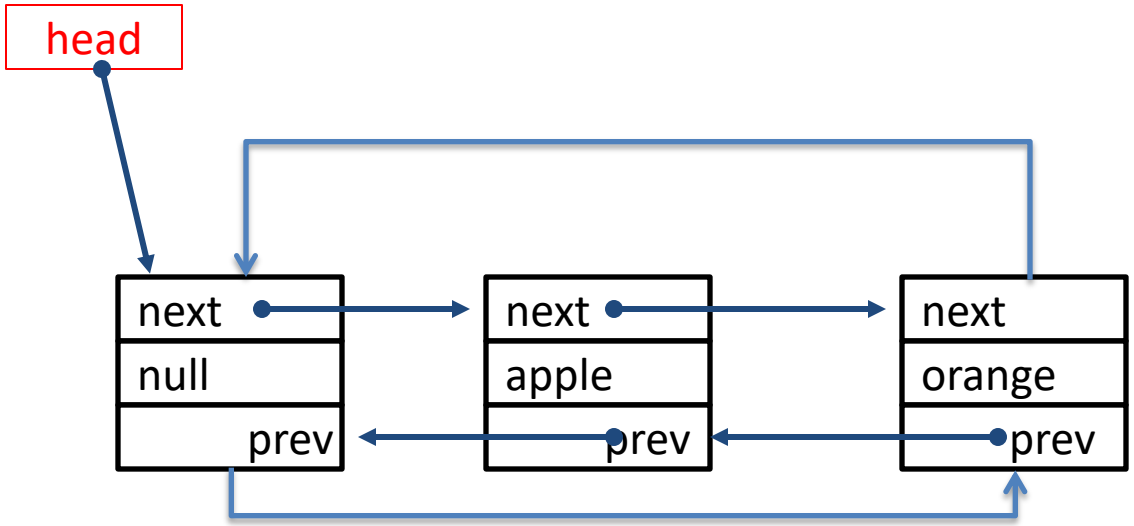
Review

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

Today

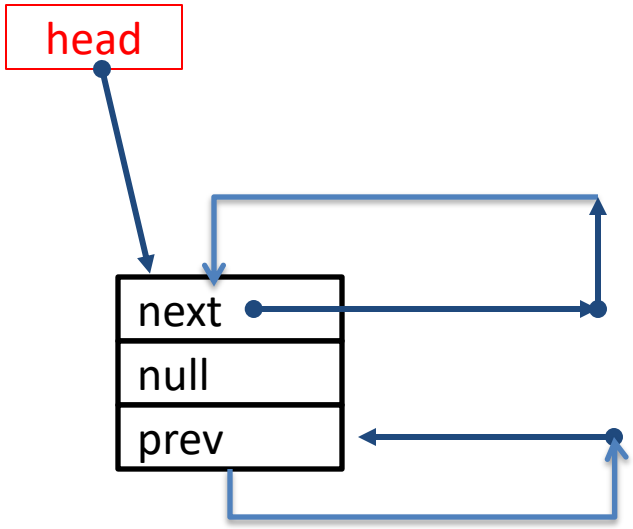
- Circular Doubly Linked List

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

Circular Doubly Linked List

EASTERN WASHINGTON UNIVERSITY
start something big

```
public class CDoublyLinkedList {
        private class Node {
                private Object data;   //comparable
                private Node next, prev;
                private Node(Object data, Node pref, Node next){
                        this.data = data;
                        this.prev = pref;
                        this.next = next;
                }
        }
        private Node head;
        private int size;
        public CDoublyLinkedList() {
                this.head = new Node(null, null, null );
                this.head.next = this.head;
                this.head.prev=this.head;
                this.size = 0;
        }
        //……
}
```

Add Method

EASTERN WASHINGTON UNIVERSITY
start something big

```
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
}
//
//1) First assign values to the links in the newNode.
//2) You change  the link that is far away from cur. (links that is not in cur).
//3) change the link in cur.
```

Remove Method

EASTERN WASHINGTON UNIVERSITY
start something big

```
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(  i < index ) {
                cur =  cur.next;
                i ++;
        }
        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;
}
```

Remove Method

EASTERN WASHINGTON UNIVERSITY
start something big

```
public boolean remove( Object data ) { //Assumes no-null data element in this list, which differ
                                       //from homework requirements.

        Node cur = this.head.next;
        while( cur!=this.head ){
            if( cur.data.equals(data) ) {
                cur.prev.next = cur.next;
                cur.next.prev = cur.prev;
                this.size --;
                return true;
            }
            cur =  cur.next;
        }
        return false;
}//end metho
```

Summary

- Circular Doubly Linked List