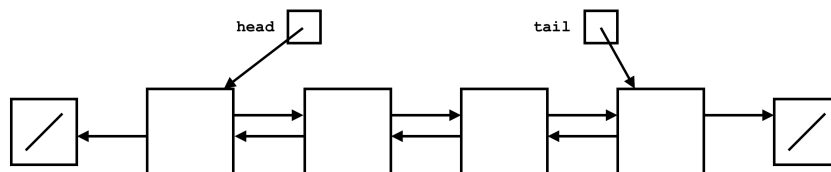


CS 314 FINAL REVIEW — REVERSE DOUBLY LINKED LIST — Solution

Linked Lists

Write an instance method for a doubly linked list which reverses the order of the elements in the list. The first element will become the last element, the second becomes the second-to-last, etc.

The doubly linked list class provided, `DoubleLinkedList`, has doubly linked nodes and the list is null terminated at both ends. There are no special header nodes, all nodes contain elements of the list. The list object holds references to the first and last elements in the list. The following is a visual digram of this list class:



Examples of calls to `reverseList()`. The result shown is the new state of `this`.

- `[0, 1, 2, 3].reverseList() => [3, 2, 1, 0]`
- `[3, 1, 4].reverseList() => [4, 1, 3]`
- `[5, 5, 5].reverseList() => [5, 5, 5]`
- `[7].reverseList => [7]`
- `[] .reverseList => []`

Use the following `DoubleLinkedList` implementation.

```
public class DoubleLinkedList<E>{

    private DoubleListNode<E> head, tail;
    private int size;

    // Inner node class
    private static class DoubleListNode<E>{
        DoubleListNode<E> prev, next;
        E data;
    }
}
```

You may not create any new data structures.

No methods are provided. Do not use any other Java classes.

```

/*
 * Pre:  none
 * Post: oldSize == newSize, 'this' list is reversed
 */
public void reverseList(){
    DoubleNode<E> currNode = head;

    while(currNode != null){
        //Swap this node's next and prev references
        DoubleNode<E> temp = currNode.next;
        currNode.next = currNode.prev;
        currNode.prev = temp;

        //The next node is now the prev node because of swap
        currNode = currNode.prev;
    }

    //Swap the head and tail references
    DoubleNode<E> oldHead = head;
    head = tail;
    tail = oldHead;
}

```