Announcements/Reminders:

- P2 and HW3 assigned
- P1 late days

Last class:

- ListADT with linked lists (cont'd)
  - \texttt{tail} reference and dummy header
- Linked List variations

Today:

- Comparing Complexities
- Shadow array improvement
- Adding iterators to Linked Lists
Linked List Variations

- Singly-linked
  - with tail
  - with (dummy) header node

- Doubly-linked

- Circular
  - Singly-linked
  - Doubly-linked
• Comparing Complexities of Array-Based and Chain-Based List ADT Implementations

**Space requirements:** (what is problem size?)

Array:

Singly-linked chain:

Circular singly-linked chain:

Doubly-linked chain:

Circular doubly-linked chain:
Comparing Complexities of Array-Based and Chain-Based List ADT Implementations (cont'd)

**Time requirements:**

<table>
<thead>
<tr>
<th></th>
<th>Constructor</th>
<th>add(E)</th>
<th>add(int,E)</th>
<th>contains(E)</th>
<th>size</th>
<th>isEmpty</th>
<th>get(int)</th>
<th>remove(int)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Array</strong></td>
<td>O(1)</td>
<td></td>
<td></td>
<td></td>
<td>O(1)</td>
<td>O(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Singly-LL</strong></td>
<td>O(1)</td>
<td></td>
<td></td>
<td></td>
<td>O(1)</td>
<td>O(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Circular SLL</strong></td>
<td>O(1)</td>
<td></td>
<td></td>
<td></td>
<td>O(1)</td>
<td>O(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Doubly-LL</strong></td>
<td>O(1)</td>
<td></td>
<td></td>
<td></td>
<td>O(1)</td>
<td>O(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Circular DLL</strong></td>
<td>O(1)</td>
<td></td>
<td></td>
<td></td>
<td>O(1)</td>
<td>O(1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Ease of implementation:

Array:

Singly-linked chain:

Circular singly-linked chain:

Doubly-linked chain:

Circular doubly-linked chain:
**add(E) when array is full**

Naive approach:

Shadow array approach:

Amortized analysis
Adding iterators to Linked Lists

Recall Iterator interface (3 operations):

```java
public class LinkedListIterator<E> implements ____________ {
    ____________ LinkedListIterator( ) {
    }
    public boolean hasNext() {
    }
    public E next() {
    }
}
```

Meanwhile, over in the LinkedList class:

```java
public class LinkedList<E> implements ListADT<E> {
    ... 
    public LinkedListIterator iterator() {
    }
}
```