remove, Iterators, and Running times

4/10/26

Administrivia

- HW 3 (testing and implementing list methods) due Wednesday (4/15)

- Exam 1 on Friday (4/17)
  - In-class, open-notes, on-paper exam
  - Codingbat-like problems, testing, array-based list methods, linked list methods, running times
  - Sample exam out soon (hopefully today)

remove(T) method

Traversing a list

Array-based

```
for(int i=0; i < num; i++) {
        operation on vals[i]
}
```

Linked list

```
Node curr = head;
while(curr != null) {
        operation on curr.val
        curr = curr.next;
}
```

Iterator ADT

- boolean hasNext()
- T next()   //returns and advances
- void remove()   //optional: removes most recent

- ListIterator: hasPrevious, previous, nextIndex, previousIndex, add(T), set(T) (optional)

Common pattern with an iterator

```
List<T> stuff = ...;   // or other Collection

Iterator<T> it = stuff.iterator();
while(it.hasNext()) {
  T val = it.next();
  //do something with val
}
```

Common pattern(s) with an iterator

```
List<T> stuff = ...;   // or other Collection

Iterator<T> it = stuff.iterator();
while(it.hasNext()) {
  T val = it.next();
  //do something with val
}
```

```
for(T val : stuff) {
  //do something with val
}
```