CS 2230
CS II: Data structures
Meeting 26: the Set ADT
Brandon Myers
University of Iowa
Question about AVL trees

after an insert imbalances an AVL tree of height $H$, how many tri-node restructurings do we have to perform to balance the whole tree?
Today’s big ideas

• A **Set** keeps track of whether an element is a in the set or not

• We can implement a set in a variety of ways, including with a list and with a tree

• Next we’ll think about a Set of **key-value pairs**, which is called a **Map**
Airline tickets

Airline knows the ticket # of every person on the flight

When a person boards at the gate, the ticket is checked

The airline is able to detect if a 2nd person tries to hand in a copy of the ticket with a # already scanned
What data structure should we use to keep track of ticket #'s?
The Set ADT

/* a collection that contains no duplicates */
public interface Set<T> {
    /* add e to the set if it is not already present */
    void add(T e);

    /* return true if e is in the set */
    boolean contains(T e);

    /* remove e from the set. Return true if e
    was in the set and false otherwise */
    boolean remove(T e);

    /* return an iterator over the elements in the set.
    The iterator does not need to return elements
    in a particular order */
    Iterator<T> iterator();
}
Peer instruction

How can we use a single Set to solve the ticket problem?

a) Set.add ticket number when reserving a ticket, Set.remove when a passenger tries to board

b) Set.add ticker number when reserving a ticket, use the iterator returned by Set.iterator and call next() each time a passenger tries to board

c) Set.add ticket number when reserving a ticket, Set.contains when a passenger tries to board

d) Set.add ticket number when a passenger tries to board

```java
public interface Set<T> {
    void add(T e);
    boolean contains(T e);
    boolean remove(T e);
    Iterator<T> iterator();
}
```
Implementing Set using a List, call it ListSet
Implementing Set using a List, call it ListSet

add(96): iterate through elements, don’t see 96 before the end, elements.append(96)
Implementing Set using a List, call it ListSet

add(220): iterate through elements, see 220, stop

[Diagram showing a ListSet with elements and a count of 3, connected to an ArrayList with elements 301, 118, 220, 96]
Implementing Set using a List, call it ListSet

contains(118): iterate through elements, see 118, return true
Implementing Set using a List, call it ListSet

contains(100): iterate through elements, get to the end without finding 100, return false
Given that ListSet uses an ArrayList to hold its elements, what is the running time of ListSet.contains?

a) \(O(1)\)
b) \(O(\log n)\)
c) \(O(n)\)
d) \(O(n \log n)\)
e) \(O(n^2)\)

if you finish early, fill out the rest of the table

<table>
<thead>
<tr>
<th></th>
<th>add</th>
<th>contains</th>
<th>insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListSet</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Peer instruction

Given that SortedListSet uses a **LinkedList** to hold its elements **in sorted order**, what is the goal for the running time of SortedListSet.add?

a) $O(1)$
b) $O(\log n)$
c) $O(n)$
d) $O(n \log n)$
e) $O(n^2)$

if you finish early, fill out the rest of the table

<table>
<thead>
<tr>
<th></th>
<th>add</th>
<th>contains</th>
<th>insert</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListSet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SortedListSet</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Admininstrivia

Midterm in-class 3/31, 50min, 2 sheets of double sided notes, no devices

Next week discussion will be a practice exam, available early during the weekend
Implementing a Set using a BinarySearchTree

This structure works for integers, what about any type T?
Today’s big ideas

• A **Set** keeps track of whether an element is a in the set or not

• We can implement a set in a variety of ways, including with a list and with a tree

• Next we’ll think about a Set of **key-value pairs**, which is called a **Map**
Acknowledgements

- Ticket Purchase by Andrew Jones from the Noun Project
- Airplane Ticket by Unlimiticon from the Noun Project