CS 2230
CS II: Data structures
Meeting 26: the Set ADT
Brandon Myers
University of Iowa
Today’s learning objectives

• Interpret code that uses the Set interface

• Describe how to use a Set for a simple application

• Finish code for a Set implemented using an unsorted List (ListSet)

• Analyze the running time of ListSet methods
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();
}
Interpret code that uses the Set interface

Set<String> s = /* call a constructor */;
s.add("eight");
s.add("ten");
s.add("eight");
System.out.println(s.contains("eight"));
System.out.println(s.remove("eight"));
System.out.println(s.contains("eight"));
Iterator<String> i = s.iterator();
while (i.hasNext()) System.out.println(i.next());

What will this program do?

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+
What data structure should we use to keep track of ticket #’s?
Describe how to use a **Set** for a simple application

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 ticket 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();
}
```

https://b.socrative.com/login/student/room CS2230X ids 1000-2999
room CS2230Y ids 3000+
ListSet: Implementing Set using a List

Diagram:
- ListSet
  - elements
  - count: 3
- ArrayList
  - elements: 301, 118, 220
ListSet: Implementing Set using a List

add(96): iterate through elements, don’t see 96 before the end, so...

```
ListSet:
  elements
  count

ArrayList:
  elements
  count

301 118 220
```
ListSet: Implementing Set using a List

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

add(220): iterate through elements, see 220, stop
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
Finish code for a Set implemented using an unsorted List (ListSet)

```java
public class ListSet<T> implements Set<T> {
    private final List<T> elements;
    public ListSet() {
        this.elements = new ArrayList<>();
    }

    // return the index of e in the elements
    // or -1 if it is not in the elements
    private int find(T e) {
        Iterator<T> it = this.elements.iterator();
        int i = 0;
        while (it.hasNext()) {
            if (this.elements.get(i) == e) {
                return i;
            }
            i++
        }
        return -1;
    }
}
```

This code has a bug. What is it?
Finish code for a Set implemented using an unsorted List (ListSet)

Write the add, contains, remove methods (use find in each)

```
public class ListSet<T> implements Set<T> {
    private final List<T> elements;

    // return the index of e in the elements
    // or -1 if it is not in the elements
    private int find(T e) {...}

    public void add(T e) { ??? }
    public boolean contains(T e) { ??? }
    public boolean remove(T e) {???}
}
```

1. go to gist.github.com
2. type in your code
3. select Create public gist
4. submit your url on socrative

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+
Analyze the running time of ListSet methods

Given that ListSet uses an ArrayList to hold its elements, give the tightest upper bound for the best case/worst case running time of ListSet methods.

Let n be the number of items in the ListSet

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

<table>
<thead>
<tr>
<th></th>
<th>add</th>
<th>contains</th>
<th>remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>best case</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>worse case</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

in your notes: justify each answer

https://b.socrative.com/login/student/
room CS2230X ids 1000-2999
room CS2230Y ids 3000+
Analyze the running time of ListSet methods

Given that ListSet uses a LinkedList to hold its elements, give the tightest upper bound for the best case/worst case running time of ListSet methods.

Let n be the number of items in the ListSet

<table>
<thead>
<tr>
<th></th>
<th>add</th>
<th>contains</th>
<th>remove</th>
</tr>
</thead>
<tbody>
<tr>
<td>best case</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>worse case</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

in your notes: justify each answer

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

exercise for you @home

https://b.socrative.com/login/student/
room-CS2230X-ids-1000-2999
room-CS2230Y-ids-3000+
Administrivia

Midterm in-class Thursday 10/26, 75min, two 8.5”x11” sheets of double sided notes, no devices

• “Practice problems for study” on Piazza
• Practice midterm2 on ICON | Files

Go to section on Thursday

• your pre-lab is to finish your HW6 PROGRESS_REPORT.txt so that you can discuss it with your TA.
• more practice testing/fixing tree code in lab
Implementing a Set using a BinarySearchTree...

This structure works for integers, what about any type T?
Today’s learning objectives

• Interpret code that uses the Set interface

• Describe how to use a Set for a simple application

• Finish code for a Set implemented using an unsorted List (ListSet)

• Analyze the running time of ListSet methods
Acknowledgements

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