

CSCI-UA-102-011-Spring-2025

Recitation - 7

Today's Agenda

- Section 10.5.1
- Problem Statements
- Quiz (Last 20 mins)

10.5.1 The Set ADT

The Java Collections Framework defines the `java.util.Set` interface, which includes the following fundamental methods:

`add(e)`: Adds the element *e* to *S* (if not already present).

`remove(e)`: Removes the element *e* from *S* (if it is present).

`contains(e)`: Returns whether *e* is an element of *S*.

`iterator()`: Returns an iterator of the elements of *S*.

There is also support for the traditional mathematical set operations of ***union***, ***intersection***, and ***subtraction*** of two sets *S* and *T*:

$$S \cup T = \{e: e \text{ is in } S \text{ or } e \text{ is in } T\},$$

$$S \cap T = \{e: e \text{ is in } S \text{ and } e \text{ is in } T\},$$

$$S - T = \{e: e \text{ is in } S \text{ and } e \text{ is not in } T\}.$$

In the `java.util.Set` interface, these operations are provided through the following methods, if executed on a set *S*:

`addAll(T)`: Updates *S* to also include all elements of set *T*, effectively replacing *S* by $S \cup T$.

`retainAll(T)`: Updates *S* so that it only keeps those elements that are also elements of set *T*, effectively replacing *S* by $S \cap T$.

`removeAll(T)`: Updates *S* by removing any of its elements that also occur in set *T*, effectively replacing *S* by $S - T$.

```
public void removeAll(SetADT<K> other) {
    for (K element : other.elements()) {
        this.remove(element);
    }
}

public void retainAll(SetADT<K> other) {
    PositionList<K> keys = this.elements();
    for (K element : keys) {
        if (!other.contains(element)) {
            this.remove(element);
        }
    }
}

public void addAll(SetADT<K> other) {
    for (K element : other.elements()) {
        this.add(element);
    }
}
```

Problem Statements

- Q10.33
- Q10.25
- Q10.26
- Q10.27
- Q10.29

(Assume the set isn't implemented by hashing but rather by UnsortedMap)