Programming and Data Structures
Test #2

1. Define the following generic methods inside the provided test class:

   a.
```
public static <E> void combineNoDuplicates(
                                        Collection<E> c1,
                                        Collection<E> c2)
```

   The method adds the elements from **c2** to **c1** only if they do not exist in **c1**.

   b.
```
public static <E> void recursiveCombineNoDuplicates(
                                        Collection<E> c1,
                                        Collection<E> c2)
```

   This method is a recursive implementation of **combineNoDuplicates**.

   c.
```
public static <E> void subCollection(Collection<E> c,
                                     int start,
                                     int end,
                                     Collection<E> subC)
```

   This method adds the elements from index **start** to index **end-1** from collection **c** to **subC**. If **start** is equal to **end**, **subC** should be empty. If **start** is greater than **end**, the method throws an exception of type **ArrayIndexOutOfBoundsException**.

   d.
```
public static <E> void splitCollection(
                                       Collection<E> c,
                                       E item,
                                       Collection<E> c1,
                                       Collection<E> c2)
```

   This method splits the collection **c** using the element **item** as the delimiter. **c1** should contain the elements from the first element to **item** (inclusive) and **c2** should contain the remaining elements after **item**. If **item** is not found in **c**, **c1** should contain all the elements from **c** and **c2** should be empty.

The methods listed below are the only methods you are allowed to use to implement the four methods above.

```
boolean add(E item)
int size()
boolean contains(E item)
Iterator<E> iterator()
boolean hasNext()
E next()
```

These methods are common to all the data structures that implement the interfaces **`Collection<E>`** and **`Iterator<E>`**, which are `ArrayList`, `LinkedList`, `Stack`, and `PriorityQueue`.

2. Determine the time complexity of the four methods and write the big-O notation as a comment before the methods' headers.

3. Test your methods using the provided main method.

4. Submit the file **`Test.java`** on course site. The file should include the definitions of the four methods described above and any other methods you may need to add.

5. **Javadoc comments are not required**.

========== Sample Program Output ==========

```
Combined Collection with no duplicates (iterative):
---------------------------------------------------
"Ohio" "Kansas" "California" "Washington" "Pennsylvania" "New Jersey"
"Vermont" "New Mexico" "Florida" "North Carolina" "New York" "Montana"
"Texas" "Connecticut" "Delaware" "New Hampshire" "Utah" "Wyoming"


Combined Collection with no duplicates (recursive):
---------------------------------------------------
"Ohio" "Kansas" "California" "Washington" "Pennsylvania" "New Jersey"
"Vermont" "New Mexico" "Florida" "North Carolina" "New York" "Montana"
"Texas" "Connecticut" "Delaware" "New Hampshire" "Utah" "Wyoming"


Sub List from the collection (index 5 to 5):
--------------------------------------------
Empty

Sub List from the collection (index 5 to 8):
--------------------------------------------
"New Jersey" "Vermont" "New Mexico"

Sub List from the collection (index 6 to 4):
--------------------------------------------
Invalid start and end indices.

Split the collection (delimiter = "Florida"):
---------------------------------------------
First part of the collection:
"Ohio" "Kansas" "California" "Washington" "Pennsylvania" "New Jersey"
"Vermont" "New Mexico" "Florida"

Second part the collection:
"North Carolina" "New York" "Montana" "Texas" "Connecticut" "Delaware"
"New Hampshire" "Utah" "Wyoming"

Split the collection (delimiter = "Arkansas"):
----------------------------------------------
First part of the collection:
"Ohio" "Kansas" "California" "Washington" "Pennsylvania" "New Jersey"
"Vermont" "New Mexico" "Florida" "North Carolina" "New York" "Montana"
"Texas" "Connecticut" "Delaware" "New Hampshire" "Utah" "Wyoming"


Second part of the collection:
Empty
```