

CMSC 132: Group Project #2

Due on Tuesday, June 30, 2015

1:00pm

Anwar Mamat

Problem 1

Fixed Size Bag LoopBag (100%) Do you know how the Flight Recorders work? They always store the last 2 hours of sound. They use a continuous loop of tape that completes a cycle every 2 hours. As new material is recorded, the oldest material is replaced. We talked about the resizable bag class. In this group project your group (1-3 students) will implement a variant of the Bag class: LoopBag. LoopBag is a fixed size bag and works like the flight recorder. The size of the bag is given the when the LoopBag object is created. After the LoopBag is full, if a new item is inserted, the oldest item in the LoopBag is replaced. You will implement the following interface:

```

import java.util.Iterator;
public interface LoopBag<E> extends Iterable {
    /**
     * Adds the given item to bad LoopBag.
     * @param item the item to add
     */
    void insert(E item);

    /**
     * Returns the number of items in this LoopBag.
     * @return the number of items in this LoopBag
     */
    int size();

    /**
     * Returns true if this LoopBag is empty and false otherwise.
     * @return true if this LoopBag is empty; false otherwise.
     */
    boolean isEmpty();

    /**
     * if the bag contains a given item?
     * @return true if bag contains the item. false otherwise
     */
    boolean contains(E item);

    /**
     * creates the union with the given LoopBag
     * if the Bag capacity is 5:
     *   1,2,3 union with 2,3,4,5 is 1,2,3,4,5
     *   1,2,3,4 union with 3,4,5,6 is 2,3,4,5,6
     */
    void union(LoopBag lb);

    /**
     * Returns an iterator for this LoopBag. Iterator iterates from oldest to newest.
     * @return an iterator for this LoopBag
     */
    Iterator<E> iterator();
}

```

Listing 1: Test Example

```
LoopBag<Integer> bag = new LoopBagImpl(5);
```

```
bag.insert(1);
bag.insert(2);
bag.insert(3);
5 bag.insert(4);
bag.insert(5);
bag.insert(6);
bag.insert(7);
10 for (Integer i: bag) {
    System.out.print(i+", ");
}

Output:
3, 4, 5, 6, 7
```

1. You are not allowed to use Java ArrayList, LinkedList, Queue, or Stack. You have to implement the LoopBag using basic arrays or linked lists.
2. Iterator iterates from the oldest to the newest.
3. DO NOT change the given JUnit test cases