

CS 0445 Spring 2024

Recitation Exercise 3

Introduction:

In Recitation Exercise 2 you added 3 methods to the `BagInterface<T>` and implemented this modified interface with the `ResizableArrayBag<T>` class. In this exercise you will implement the same modified interface, but this time by modifying the `LinkedBag<T>` class.

Before continuing with this exercise, **look again at Recitation Exercise 2**. In particular, be sure you fully understand the specifications of the `union()`, `intersection()` and `difference()` methods.

The required methods are repeated below. However, for more detailed descriptions of their functionalities, refer back to Recitation Exercise 2.

```
// Creates a new bag that combines the contents of this bag
// and anotherBag.
//   @param anotherBag The bag that is to be added.
//   @return A combined bag.
public BagInterface<T> union(BagInterface<T> anotherBag);

// Creates a new bag that contains those objects that occur
// in both this bag and anotherBag.
//   @param anotherBag The bag that is to be compared.
//   @return A combined bag.
public BagInterface<T> intersection(BagInterface<T> anotherBag);

// Creates a new bag of objects that would be left in this bag
// after removing those that also occur in anotherBag.
//   @param anotherBag The bag that is to be removed.
//   @return A combined bag.
public BagInterface<T> difference(BagInterface<T> anotherBag);
```

For this exercise your task is to implement the three new methods in the `LinkedBag<T>` class so that they work as described. The interface is available in file [BagInterface.java](#). The `LinkedBag<T>` class (without the new methods) is available in file [LinkedBag.java](#). Test your implementation with the main program [CS445Rec3.java](#) (note: this file is the same as `CS445Rec2.java` except for the bag object types). The output contents should match that shown in [Rec3Out.txt](#). Note however, that within any bag the order of the data does not matter – so if you bags show the contents in a different order that does not necessarily mean that they are incorrect.

Important Implementation Restriction: Since the primary point of this exercise is to give you experience programming with linked-lists, it is counter-productive to access the data in your bags via arrays. Thus, **you should not use the `toArray()` method anywhere in this exercise.**

You are, however, allowed to use any of the other `BagInterface<T>` methods that are previously implemented in the `LinkedBag<T>` class.

Hints:

You will need to iterate through the contents of one or perhaps both of your `LinkedBag<T>` objects to implement these methods. This can be done by accessing the underlying linked lists in a loop.

The argument bag for each of these methods will be passed in via a `BagInterface<T>` parameter. This parameter type will restrict access to the methods in `BagInterface` and will not allow direct access of the underlying linked list. To get this access (if you need it) you will need to cast the parameter to type `(LinkedBag<T>)`.