# Exercise 2

CS 240

Ocean Lu

01/29/18

Exercise 2 concerned the implementation with linked data. There are many benefits into utilizing chains rather than common arrays. Such include adjustable sizes, ease of removal and addition of nodes, as well as recycling the entire chain. However, cons to chains is that it requires more memory than arrays of the same length. Another noticeable difference between the two is that chains only allow sequential access to its respective elements while arrays allow random access. Exercise 2 shows the flexibility and similarity to arrays in terms of applications.

My test code:

```
public class Test {

    public static void main(String[] args) {
        LinkedBag bag = new LinkedBag();
        System.out.println("Current Size: " + 
bag.getCurrentSize());
        bag.add(1);
        bag.add(1);
        bag.add(1);
        bag.add(13);
        bag.add(99);
        System.out.println("Added: 1 1 1 13 99");
        if (!bag.isEmpty()) {
            System.out.println("Not Empty.");
        }
        bag.remove();
        System.out.println("Removed randomly.");
        bag.remove(1);
        System.out.println("Removed one 1.");
        System.out.print("In the bag: ");
        for (int i = 0; i < bag.toArray().length; i++) {
            System.out.print(bag.toArray()[i] + " ");
        }
        System.out.println("\nFrequency of 1: " + 
bag.getFrequencyOf(1));
        System.out.println("Contain 5: " + bag.contains(5));
        bag.clear();
        System.out.println("Nullified.");
        System.out.println("Current Size: " + 
bag.getCurrentSize());
    }
}
```

My outcome:

```
run:

Current Size: 0

Added: 1 1 1 13 99

Not Empty.
```

```
Removed randomly.

Removed one 1.

In the bag: 13 1 1

Frequency of 1: 2

Contain 5: false

Nullified.

Current Size: 0

BUILD SUCCESSFUL (total time: 0 seconds)
```

Here, my first step was to create a LinkedBag, the class name of Exercise 2, and see if the constructor was suitable. Having created one successfully, I went to test around the methods in the class. The first I tried out was the getCurrentSize(), where I put it in a print statement, showing that the bag was empty. Next, I filled up my bag with five integers, which were implemented with the add(T newEntry) function. Once filled to my satisfaction, I tested to see if my bag was empty with the isEmpty() method, where I placed it in an if statement, as it returned a Boolean value. Next, I tested the remove() method—of a random nature. I also tested the remove(T anEntry) method directly after. Not knowing what was in my bag, I used my toArray() method to see what was in the bag. Because I added the same integer into my bag multiple times, I utilized the getFrequencyOf(T anEntry) method to see how many I had, where I compared to the above to see if it was indeed a correct ratio. To check if I had a value in my bag without knowing, I selected a random integer and used the contains(T anEntry) method to see. Finally, I cleared my list, and had a getCurrentSize() check. In between all functions, I had statements that were pointers on where I was.

This project was especially simple to implement. The only difficulty was learning a new form of “array”. With the reference of the textbook and the professor’s slide notes, it was especially easy to understand.