PRINCETON
UNIVERSITY

COS 226–Algorithms and Data Structures
Week 2: *Generics, Iterators and Elementary Sorting (Algorithms §1.3 and 2.1)*

Version: February 13, 2018

Exercise 1 – Iterators (video §3.E)

Consider the partial code of LinkedBag.java below. The container data structure for the LinkedBag is a linked list. We need to develop a ListIterator that can traverse through the list and produce the elements in the list sequentially starting with the head of the list. (The line numbers match the downloadable version of the code on the precept page.)

```
15 public class LinkedBag<Item> implements Iterable<Item> {
16  private Node first;     // beginning of bag
17  private int n;          // number of elements in bag
18
19  // helper linked list class
20  private class Node {
21     private Item item;
22     private Node next;
23  }
```

A. The code for the add() method in the LinkedBag class is given below. When a new element is added to the LinkedBag, does it add the new element to the beginning or to the end of the Linked list?

```
53  public void add(Item item) {
54    Node oldfirst = first;
55    first = new Node();
56    first.item = item;
57    first.next = oldfirst;
58    n++;
59  }
```

Note that all code found in the algs4 library is listed on the booksite. The url is: https://algs4.cs.princeton.edu/code/

B. Complete the missing code below starting at line 66 in iterator(), the ListIterator class declaration (line 70), ListIterator(), hasNext() and next().

```
64   // Returns an instance of the iterator class (to be used for iterating)
65   public Iterator<Item> iterator()  {
66
67   }
68
69   //  traverse through the list sequentially starting at first; use Iterator and the g
70
71   private class ListIterator implements _____________________________________ {
72      private Node current;
73
74      public ListIterator(){
75
76      }
77
78      public boolean hasNext()  {                                          }
79      public void remove()      { throw new UnsupportedOperationException();  }
80      public Item next() {
81          if (!hasNext()) throw new NoSuchElementException();
82
83
84
85
86      }
87  }
```

C. The code below is a client of the LinkedBag class. What output is produced by this code?

```
LinkedBag<Integer> myBag = new LinkedBag<Integer>();
myBag.add(3);
myBag.add(1);
myBag.add(2);
for (int i : myBag){
     for (int j : myBag) {
            StdOut.println(i + "␣" + j);
}
 StdOut.println();
}
```

Exercise 2 – Elementary sorting

Give an example of the best and worst case for both insertion sort and selection sort (using an array of the integers between 0 and 4) in terms of number of compares and number of exchanges.

	best case	worst case
insertion sort		
selection sort		