Practice Test

August 29, 2016

1. (35 Pts) The class MyList represents a linear list. This class has a member class that implements the Java interface Iterator (recall that Iterator has the public methods hasNext, next, and remove, the remove method is not required for this problem). The class MyList has the public methods clear() that makes the list this empty, append(Object x) that appends x to the right end of the linear list this and iterator() that returns an iterator for the list.

   The public static method split(MyList a, MyList b, MyList c) is to be written. Following an invocation of split, b has elements 0, 2, 4, ... of a (in this order) and c has the remaining elements (in order). List a is unchanged. For example, if a is [a, b, c, d, e], then following a split, a, b and c are, respectively, [a, b, c, d, e], [a, c, e] and [b, d]. Notice that the split lists b and c may not be empty initially.

   (a) (25 Pts) Write code for the public static method split. Do not assume the existence of any methods other than those stated above.

   (b) (10 Pts) What is the time complexity of your code as a function of the size of list a? Explain how you arrived at this complexity. Assume that each of the methods given above takes O(1) time.

2. In the class MyArrayList a linear list is represented using a one-dimensional array element. The data member size is such that the list elements are in positions 0 through size-1 of the array. The method removeNull (which is a method of MyArrayList) removes every null element of the list. For example, suppose that the list x is [1, null, 3, null, null, 6] (list size is 6). Then following the invocation x.removeNull(), the list is [1, 3, 6] (the size is now 3).

   (a) Write code for the member method removeNull. Do not assume the existence of any methods for MyArrayList.

   (b) What is the time complexity of your code as a function of the initial list size?

3. Consider the class MyCHList, which is a singly-linked circular list with header node. The data members of this class are headerNode and size. The

data type of headerNode is MyChainNode. Objects of type MyChainNode have the data members element (which is an int) and next (whose data type is MyChainNode.

Nodes on the circular list are linked together using the field next. size is an integer whose value equals the number of elements in the chain (this count excludes the header node/element).

The value in the element field of a header node is INFINITY, where INFINITY is a constant that is larger than the value in any non-header node. Excluding the header, the elements are in ascending (or more accurately nondecreasing) order left to right. The method merge(MyCHList a, MyCHList b) merges the elements in a and b to yield a sorted circular list with header, which is pointed at by this.headerNode. Following the merge, the lists a and b are empty (i.e., they have only a header node and their size is 0).

If a and b are initially [INFINITY, 1, 4, 9] and [INFINITY, 2, 3], then following the merge, a and b are [INFINITY] and this is [INFINITY, 1, 2, 3, 4, 9] (note that INFINITY is the element in the header). The size of this is 5 and that of a and b is 0.

(a) Write code for the member method merge. You must not create new nodes. Reuse the nodes of a and b. Do not assume the existence of any methods for MyCHList. (Possible Hint: To keep the code short, use a single loop.)

(b) What is the time complexity of your code as a function of the length of the resulting chain? Explain how you arrived at this complexity.