CSCI 2110 Data Structures and Algorithms
Fall 2022
Assignment No. 4

Date Given: Monday, October 31st, 2022
Due: Monday, November 14th, 2022, 11.59 PM

This assignment on the Ordered List Data Structure. You will write a program to implement the two-finger walking algorithm discussed in the lectures, and small extensions on the algorithm to perform various merging operations on ordered lists.

You will need the following files to complete your work:

OrderedList.java (Generic Ordered List Class)
list1.txt (first sample input text file)
list2.txt (second sample input text file)

Exercise: Write a program called OrderedListDemo.java that does the following:

1. Prompt the user to enter two text files, each consisting of words (names) and create two ordered lists. For example, if one of the input text files contains:

```
Shai
Tom
Jim
Aaron
Barbara
Beth
Fred
Jack
Jarred
Jill
Amar
Ralph
Jill
Hillary
```

your program should create the following ordered list:

```
[Aaron, Amar, Barbara, Beth, Fred, Hillary, Jack, Jarred, Jill, Jim,
Ralph, Shai, Tom]
```

Note: Although Jill is repeated in the input list, the ordered list does not have repeated items.

2. Write a method called merge that accepts two ordered lists, list1 and list2, as input parameters and creates and returns a third list that is a merger of the two ordered lists. Use the two-finger walking algorithm discussed in class.

Your method's header should be written as follows:

```
public static <T extends Comparable<T>> OrderedList<T> merge(OrderedList<T> list1,
OrderedList<T> list2){

  //TODO

}
```

For example, if list1 is {Amar, Boris, Charlie, Dan, Fujian, Inder, Travis} and list2 is {Alex, Ben, Betty, Charlie, Dan, Pei, Travis, Zola, Zulu}

then the method should return the new ordered list

{Alex, Amar, Ben, Betty, Boris, Charlie, Dan, Fujian, Inder, Pei, Travis, Zola, Zulu}

2. Write another method called difference that accepts two ordered lists, list1 and list2, as input parameters and creates and returns a third list that is an ordered list with the items in list1 that are not in list2. Make a small modification to the two-finger walking algorithm to implement this method.

Your method's header should be written as follows:

```
public static <T extends Comparable<T>> OrderedList<T> difference(OrderedList<T> list1, OrderedList<T> list2){

  //TODO

}
```

For example, if list1 is {Amar, Boris, Charlie, Dan, Fujian, Inder, Travis} and

list2 is {Alex, Ben, Betty, Charlie, Dan, Pei, Travis, Zola, Zulu}

then the method should return the new ordered list

{Amar, Boris, Fujian, Inder}

3. Write another method called common that accepts two ordered lists, list1 and list2, as input parameters and creates and returns a third list that is an ordered list with the items that are common in list1 and list2. Make a small modification to the two-finger walking algorithm to implement this method.

Your method's header should be written as follows:

```
public static <T extends Comparable<T>> OrderedList<T> common(OrderedList<T> list1, OrderedList<T> list2){

  //TODO

}
```

For example, if list1 is {Amar, Boris, Charlie, Dan, Fujian, Inder, Travis} and

list2 is {Alex, Ben, Betty, Charlie, Dan, Pei, Travis, Zola, Zulu}

then the method should return the new ordered list

{Charlie, Dan, Travis}

4. Write the newly created ordered lists into three text files, namely, merged.txt, diff.txt and common.txt.

Here's the overall structure of your code and input dialog:

```
public class OrderedListDemo{

        public static void main(String[] args){

                //TODO

        }
        public  static  <T  extends  Comparable<T>>  OrderedList<T>  merge(OrderedList<T>
list1, OrderedList<T> list2){

        //TODO

        }

        public static <T extends Comparable<T>> OrderedList<T> difference(OrderedList<T>
list1, OrderedList<T> list2){

        //TODO

        }

        public  static  <T  extends  Comparable<T>>  OrderedList<T>  common(OrderedList<T>
list1, OrderedList<T> list2){

        //TODO

        }

}
```

```
Enter the first filename to read from: list1.txt
Enter the second filename to read from: list2.txt
The merge operations are complete and the results are in merged.txt,
difference.txt and common.txt
```

Note: Test your program on small input files to ensure that it works correctly and then run it on list1.txt and list2.txt.

What to submit:
You are to submit just one .java file, namely, OrderedListDemo.java with the three methods. The markers will run your code with the input files and test them.

You MUST SUBMIT .java file that are readable by your TAs. If you submit files that are unreadable such as .class, you will lose points. Please additionally comment out package specifiers.

Late Submission Penalty: The assignment is due on Monday at 11.59 PM. Late submissions up to 5 hours (4.59 AM on Tuesday) will be accepted without penalty. After that, there will be a 10% late penalty per day on the mark obtained. For example, if you submit the assignment on Tuesday at 12 noon and your score is 8/10, it will be reduced to 7.2/10. Submissions past five days after the grace submission time will not be accepted.