CMPS 303 Data structures
Case study: The Farey Fractions Sequence

Farey fractions of level one are defined as sequence $(\frac{0}{1}, \frac{1}{1})$. This sequence is extended in level two to form a sequence $(\frac{0}{1}, \frac{1}{2}, \frac{1}{1})$. Sequence $(\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1})$ at level three, sequence $(\frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1})$ at level four, so that at each level n, a new fraction $\frac{a+b}{c+d}$ is created between two neighbor fractions $\frac{a}{c}$ and $\frac{b}{d}$ only if $c + d \leq n$. Write a program, which for a number n entered by the user, creates –by constantly extending it—a linked list of fractions at level n and then displays them.

SOLUTION

```
import java.util.*;

// Class Farey Seq extends the Java LinkedList class
class FareySeq extends LinkedList<Fraction>
{
        // added for serialisation
        private static final long serialVersionUID = 1L;

        public FareySeq(){
                super();
        }

        public void display()
        {
                for (int i = 0; i < size(); i++) System.out.print(get(i));
                System.out.println();
        }
}

// The class Fraction
class Fraction
{
        int Num, Den ;

        public Fraction(int a, int b)
        {
                Num = a;
                Den = b;
        }

        public String toString()
        {
                return Num + "/" + Den + ", ";
        }
}


// The test class containing the main() function
class FareySeqApp
{
        public static void main(String[] args)
        {
                FareySeq fList = new FareySeq();
```

```
                Fraction f1 = new Fraction(0,1);
                Fraction f2 = new Fraction(1,1);

                try{
                        fList.add(f1);
                        fList.add(f2);
                }catch(IndexOutOfBoundsException ex){}

                Scanner ab = new Scanner(System.in);
                int userInput;
                System.out.print("The current sequence is ");
                fList.display();
                System.out.println("Please enter number of terms to expand the sequence:");
                userInput = ab.nextInt();

                // Generate new sequences and add them to the list. The new list is returned
                FareySeq newList = fList;
                for(int i = 1; i <= userInput;i++)
                {
                        newList = insertFraction(newList,i);
                }
                System.out.println("The list after expanding is:");
                newList.display();
        }

        public static FareySeq insertFraction(FareySeq fL, int n)
        {
                Fraction fNode1,fNode2;
                int newSize = fL.size() - 1;
                for ( int i = 0; i < newSize; i++)
                {
                        fNode1 = (Fraction) fL.get(i);
                        fNode2 = (Fraction) fL.get(i+1);

                        if (fNode1.Den + fNode2.Den <= n)
                        {
                                int newNum = fNode1.Num + fNode2.Num;
                                int newDen = fNode1.Den + fNode2.Den;
                                Fraction newNode = new Fraction(newNum, newDen);
                                fL.add(i+1,newNode);
                                newSize++;  // increment newsize after adding terms
                        }
                }
                return fL;
        }
}
```