Farey sequences

Dr. Sam Dolan (s.dolan@sheffield.ac.uk)

This is a warm-up exercise for MAS212 Assignment #1 (2018).

(1) Mathematical background. The **Farey sequence** F_n is the set of all irreducible rational numbers $\frac{a}{b}$ in the unit interval with denominator b less than or equal to n, arranged in increasing order. For example,

$$F_5 = \{\tfrac{0}{1}, \tfrac{1}{5}, \tfrac{1}{4}, \tfrac{1}{3}, \tfrac{2}{5}, \tfrac{1}{2}, \tfrac{3}{5}, \tfrac{2}{3}, \tfrac{3}{4}, \tfrac{4}{5}, \tfrac{1}{1}\}. \tag{1}$$

(*Irreducible* means that the numerator and denominator are coprime, that is, they have no prime factors in common.)

(2) The task. We want to write a Python function to compute the Farey sequence F_n.

How could you do this? One way would be by inspecting all the fractions i/j, with $1 \leq i \leq j$ and $1 \leq j \leq n$; keeping only those where i and j are coprime; and then sorting the list. But there is a better way ...

(3) An algorithm. The terms of a Farey sequence may be generated, in order, by a simple recurrence relation which builds up the sequence $F_n = \left\{ \frac{a_0}{b_0}, \frac{a_1}{b_1}, \ldots, \frac{a_N}{b_N} \right\}$.

1. Let $a_0 = 0$, $b_0 = 1$, $a_1 = 1$ and $b_1 = n$.
2. Find
$$a_{i+1} = \lambda_i a_i - a_{i-1}, \qquad b_{i+1} = \lambda_i b_i - b_{i-1}, \qquad \lambda_i = \left[\frac{b_{i-1} + n}{b_i} \right],$$
where $[x]$ is the integer part of x. (*Hint:* use `math.floor()`).
3. If $a_{i+1} = 1$ and $b_{i+1} = 1$, then the sequence is complete.
Otherwise, increment i by 1 and return to step 2.

(4) Python code. Write Python code to implement the algorithm. More specifically, write a function `farey(n)` that returns lists of numerators and denominators.

```
def farey(n):
    ...
    return a, b
```

Test your code by computing F_5 and F_{10}.

(5) *Extension.* Calculate $|F_n|$ for $n = 1, \ldots, 100$. What is $|F_n|$?

Now read the brief for Assignment #1.

Appendix: Example implementation

```
import math

def farey(n):
    a = [0,1]
    b = [1,n]
    nmax = math.factorial(n)
    for k in range(1, nmax):
        lam = math.floor((b[k-1] + n)/b[k])
        a.append(lam*a[k] - a[k-1])
        b.append(lam*b[k] - b[k-1])
        if (a[k+1] == 1) and (b[k+1] == 1):
            break
    return a, b

a,b = farey(7)

c = ["%d/%d" % t for t in zip(a,b)]
print(", ".join(c))
```