Programming Assignment 1

Darin Goldstein

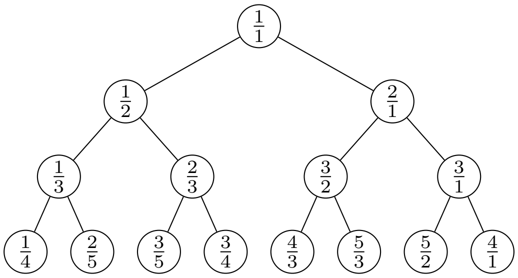
1 Deadline

9/10/2020 5 PM

2 The Fraction Tree

It is possible to build a fraction tree in the following way. Starting with $L = 1$, on the Lth level of the tree, perform the following steps.

1. Place the fraction $\frac{0}{1}$ on the left and $\frac{1}{0}$ on the right.
2. Perform the following action L times, between each pair of fractions $\frac{a_1}{b_1}$ and $\frac{a_2}{b_2}$ on this level, insert the fraction $\frac{a_1+a_2}{b_1+b_2}$.
3. Index the fractions on this level by starting $\frac{0}{1}$ with index 0 and going up to 2^L.
4. For each *odd* index $2i+1$, your parent index on the line above is $2\lfloor \frac{i}{2} \rfloor + 1$. All even indices are discarded from the tree.



It can be proved that every possible fraction eventually appears once and only once in this tree. Input to your program will be a "large" fraction M/N where M and N are large. (You may use the java.BigInteger package if you want.) Output from your program will be the first fraction $\frac{a}{b}$ in the tree such that $\frac{a}{b} \approx \sqrt{\frac{M}{N}}$ in the following sense: $\frac{a}{b}$ must be the first fraction in the tree such that

$$|Na^2 - Mb^2| < b$$

The output will be two lines, a and then b.