

CSE 3302 Lab Assignment 1

Due September 18, 2013

Goal:

Elementary understanding of Pascal, JavaScript, and Scheme.

Requirements:

1. Write and test a program in each of the three languages indicated above to count the rational numbers in reduced form that are strictly between two other rational numbers ($\frac{lowNum}{lowDen}$ and $\frac{highNum}{highDen}$) such that the sum of the numerator and denominator of each included value does not exceed a threshold value. For the three languages, the input should be handled in the following ways:
 - a. Pascal - the five non-negative integers are provided in the order: *threshold lowNum lowDen highNum highDen*
 - b. Scheme - a function of five arguments (`count threshold lowNum lowDen highNum highDen`)
 - c. JavaScript - a simple html interface is available on the course webpage
2. Email your programs (and .html if you change it) as a .zip file to `sourabh.bose@mavs.uta.edu` by 12:45 p.m. on September 18, 2013.

Getting Started:

1. Suppose that the Pascal input is: 5 1 2 2 1

The result is 3 since there are three fractions ($\frac{2}{3}, \frac{1}{1}, \frac{3}{2}$) in reduced form between $\frac{1}{2}$ and $\frac{2}{1}$.

If the threshold were increased to 10, there would be nine results ($\frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1}, \frac{5}{4}, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}$) to count.

2. You may approach this problem in any way you wish, but *simple* recursive code based on the *Stern-Brocot tree* may be developed. Continued fraction concepts are not needed, but the notion of a *mediant* is useful. This approach also avoids the need for computing GCDs.
3. You do not need to output the fractions in your submitted versions.
4. Example values will be posted to the lab directory.
5. Use `longint` for Pascal integers.