CSE 341, Autumn 2008, Assignment 7
Ruby Warmup
Due: Friday November 21, 10:00pm

18 points total (4 points each for Questions 1, 2, and 4; 2 points for Question 3)

Revised: Nov 19

You can use up to 2 late days for this assignment.

This assignment involves defining some simple classes in Ruby for binary and n-ary trees, and writing unit tests for them. Put your tree classes in one file (say `trees.rb`) and your unit tests in another (say `treetests.rb`).

1. Define 2 classes `Leaf` and `BinaryNode` with the methods described below. Put these definitions in a file `trees.rb`. An instance of either class is a "tree of strings" — a `Leaf` has one string and a `BinaryNode` has no string itself but has two smaller "trees of strings."

   (a) `Leaf`'s `initialize` takes one argument (assumed to be a string, no need to check).

   (b) `BinaryNode`'s `initialize` takes two arguments, both assumed to be "trees of strings." These are the node's children.

   (c) `iterate` takes one argument of class `Proc` (i.e., something produced by `lambda {|x| ...}`) and calls its argument with each string in the tree.

   (d) `min` should return the minimum element in the tree (as determined by the `<` method).

   (e) `max` should return the maximum element in the tree (as determined by the `>` method).

   (f) In `BinaryNode`, define a *class method* `self.concatAll` that takes one argument, a "tree of strings," and returns all its strings concatenated together. `self.concatAll` should use `iterate`.

2. Define a class `NaryNode` that is like `BinaryNode` except it can have any positive number of children. Note:

   - `initialize` should take an array of trees. It should raise an error if the array's length is 0. Else it should store a copy of the array in a field. Each tree in the array is one of the node's children.

3. In a comment in your code, answer each of these questions in a few English sentences:

   (a) If you built a tree using just the `Leaf` and `BinaryNode` classes but you put integers at each leaf instead of strings, what would happen if you called the tree's `concatAll` method? Why?

   (b) If you used integers as in the previous problem but part of your tree was built with `NaryNode`, what would happen if you called the tree's `concatAll` method? Why?

4. Finally, define a suitable set of unit tests for your binary and n-ary trees. Your tests should test all of the methods for `Leaf`, `BinaryNode`, and `NaryNode`. (In particular, it should test that the `intialize` method for `NaryNode` raises an exception if the array's length is 0.)

Turnin: Turn in your two files, one with the tree definitions and the other with your unit tests. You don't need to turn in a script showing your program running — the TA's can just run the unit tests for that.