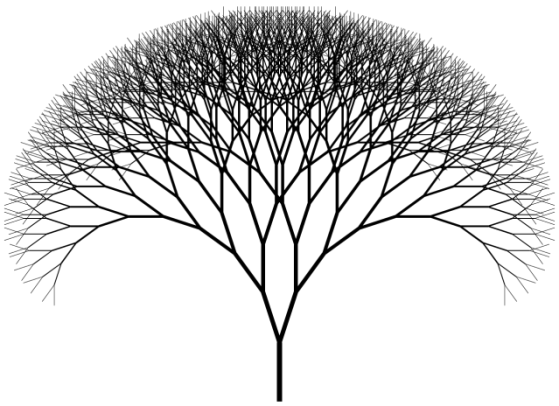
ETEC2101	LabB01: **QuadTrees**	Points: **up to…**
Assigned: **11/28/2016**	Due: **12/9/2016 at 5:00pm SHARP**	**(35 quiz or hw points, your choice)**
		both are ~3% increase in overall

Tasks:



1. Spawn n circles of varying sizes and store them in a container (or array) in your MainClass (which should use the Slick2D template, or similar).
2. Create a CircleSolver interface class which includes at least these methods:
    a. **void draw(Graphics g)**: draws all the circles in this solver and any extra "stuff" [the BruteForce solver won't have any extra stuff; the QuadTree solver will].
    b. **double findOverlaps(HashSet<Circle> set)**: this method should add all circles that overlap with each other to the set. I'm using a set so we don't double-count circles. The double returned should be the time (in seconds) to find these overlaps.
    c. Any other methods you deem necessary / useful.
3. Create two classes which both implement the CircleSolver interface: BruteForceSolver and QuadTreeSolver
4. BruteForceSolver should find all overlaps using a $O(n^2)$ algorithm (basically, check each circle against all others…but only check each pair of circles once).
5. QuadTreeSolver should implement a quadtree (as dicussed in lecture). The findOverlaps method should be $O(m * \log_4 n)$ where m is the bucket size.
6. Make your application class capable of switching (at run-time) between the two types of solvers. Use a variable of type CircleSolver to store the current solver.
7. Make the circles move / pause at runtime.
8. Mark overlaps visually (e.g. make them red)
9. If using a quadtree solver, show the grid.