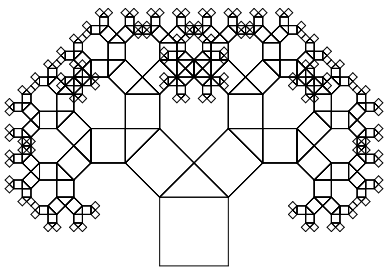
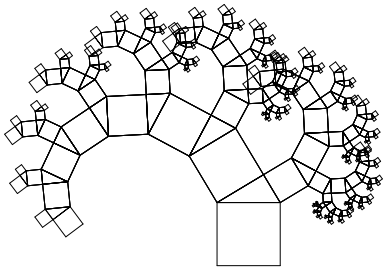
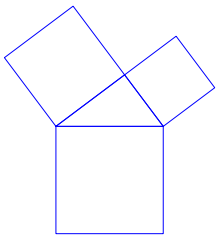
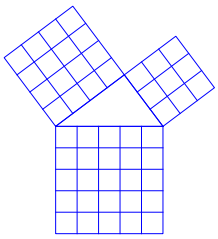
MCS 320 Project Two : Pythagorean Trees

The goal of the project is to use Sage to draw Pythagorean trees:

1. A Visual Proof of the Theorem of Pythagoras

We may formulate the theorem of Pythagoras as follows. Let a, b, and c denote the lengths of the edges of a triangle. If the edges with lengths b and c are perpendicular to each other, then $a^2 = b^2 + c^2$. To see why this is so, attach to each edge of the triangle a square with sides equal to the length of the corresponding edge. For $(a, b, c) = (5, 4, 3)$, we see

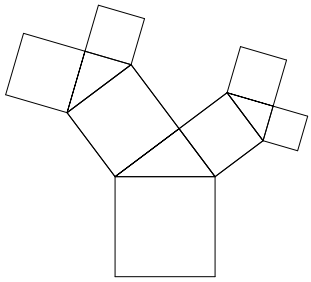



Assignment One. Use Sage worksheet to generate the pictures from above.
The requirements are:

1. The input parameters in the worksheet are b and c. Then a is computed: $a^2 = b^2 + c^2$.
2. New pictures are generated by rerunning commands in the worksheet for new values of b and c. Separate the computation of the plot structures from the rendering.
3. The largest edge is always in horizontal position. By a natural choice of coordinates this largest edge has end points $(0, 0)$ and $(a, 0)$.
4. Use `line` to draw the edges.
5. Provide justifications for the computations of the coordinates of the points. For example, the coordinates (x, y) of the top corner of the triangle where the two perpendicular edges meet satisfy a simple algebraic system. Formulate this system and solve it symbolically to find formulas in terms of (a, b, c) for the coordinates of the top corner. Do likewise for the corners of the squares.

2. A Trunk of a Pythagorean Tree

Imagine the visualization of the theorem as a tree trunk:



In the picture above two triangles of smaller size were created with dimensions to match the lengths of the sides of the larger triangle. The squares of the smaller triangles are slided above the squares of the larger triangle.

Assignment Two. Collect the commands you used in the first assignment into a Sage function. You can call it "`shiftpyt`". This function has the following input arguments:

1. Lengths of the sides `b` and `c`.
2. Coordinates for the leftmost corner of the triangle. With the natural choice of coordinates to display one triangle, this leftmost corner has coordinates `(0, 0)`.
3. A rotation angle `t`. The coordinates of every point `(x, y)` are multiplied with the rotation matrix:
$$\begin{bmatrix} \cos(\texttt{t}) & \sin(\texttt{t}) \\ -\sin(\texttt{t}) & \cos(\texttt{t}) \end{bmatrix}.$$

The output of the function is a plot structure which can be directly given to `show()` for rendering. Please note that the function `shiftpyt` does ***not*** do the plotting.

In the Sage notebook/worksheet for your answer, illustrate how to use your `shiftpyt` to display the trunk from above, made for `b = 4` and `c = 3`.

3. Drawing Pythagorean Trees recursively.

To draw Pythagorean trees, we apply `shiftpyt` from the previous assignment recursively.

Assignment Three. Write a recursive function to return the plot structures to display Pythagorean trees. The function contains the same arguments as `shiftpyt` from the previous assignment. One extra input parameter determines the depth of the recursion. Depth equal to zero just gives the plot structures for the first assignment. Depth equal to one returns the plot structures for the second assignment. The plots shown at the beginning of the project description were made with depth level equal to six.

The deadline is Friday 1 April 2016 at 11AM

Bring *your* solution to the project to class. The *your* is emphasized to stress that your solution is the result of an *individual* effort. Collaborations are **not** permitted.

The solution to this project consists in two parts:

1. A print out of the Sage notebook or SageMathCloud worksheet that you bring to class.

   Deliver a well written document, with grammatically correct and complete sentences, without spelling mistakes, appropriately structured into sections and subsections.

   The printed version must contain all relevant plots.

2. The Sage notebook/worksheet that you email as an attachment to me.

   The notebook/worksheet should run as a computer program, from top to bottom with consistent output and without errors.

If you have questions or difficulties, feel free to come to my office for help.

References

[1] Frank Drewes. Grammatical Picture Generation. Springer-Verlag, 2006.