**Recursion**

Recursion is the method of solving a problem that depends on smaller versions of the same problem. An example of this would be Russian Nesting Dolls. Let’s say the goal is to find the smallest doll. You could solve this problem recursively by following the following rule.

* If the doll in front of you is openable, open it, put what’s inside in front of you, move the old doll to the right, and repeat this for the doll in front of you.

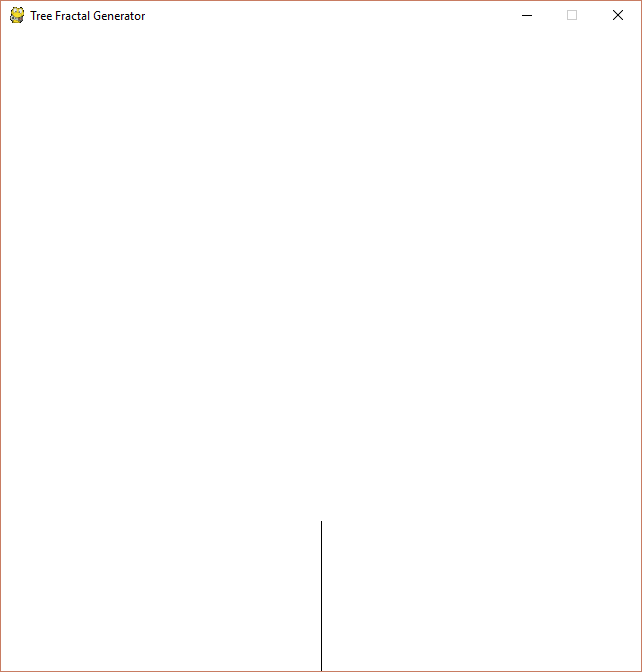
If you keep doing this, eventually the smallest doll will be in front of you, and all the bigger dolls will be to the right.



Divide and Conquer is a strategy that is often used in conjunction with recursion to solve various problems. It is the idea of taking a complicated problem, and dividing it into several smaller, simpler problems and solving those problems to get the solution to the larger problem. An example of this in our daily lives would be looking for something you’ve lost in your home. Let’s say you lost your keys somewhere in your house. Rather than looking in random places until you find it, you can use divide and conquer to solve the problem. First, split the house into the various rooms (bedroom, kitchen, hall, etc.). Then split each room into various places the keys could be (under the bed, in the fridge, on the study table, etc.). Finally, check each of these places one by one, and you will eventually find the keys. By using divide and conquer, we were able to simplify the complicated problem of finding the keys somewhere in the house to a simple problem of checking specific places for keys.



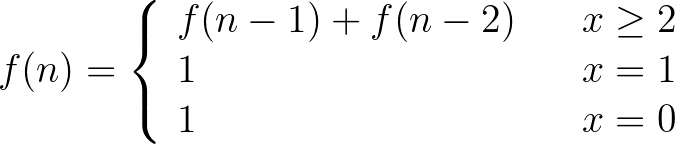
One great example of recursion is the Tree Fractal. The Tree Fractal starts off with one vertical line, and then branches off the vertical line with two lines at an angle from the first line. For each new line that’s created it is also branched off by two more lines that go in two different directions. This ends up in a tree like fractal, hence the name, Tree Fractal.



This tree fractal was made by me in Python. Here’s the code if you’re interested:

<https://github.com/mbhaskar1/PythonProjects/blob/master/TreeFractal/fractal.py>

An example of a problem you might see that would be solved using recursion is a piece wise function with references to the function in the definition. These kinds of problems are solved using the S-S-S strategy. The S-S-S strategy is to simplify the function down to the base case, substitute the base case into the last expression created, and then solve the expression until you have the answer. Here’s an example function:



This is actually the Fibonacci function, which is a great example of a recursive piece-wise function. Now let’s say we had to calculate f(4).

First, we would simplify this expression:









Next, we substitute the base case:



Finally, we solve the expression to get the answer:

