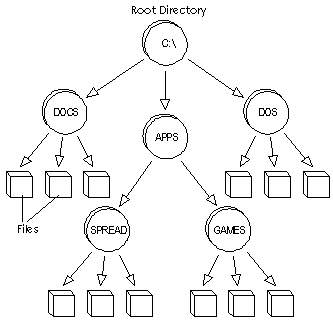
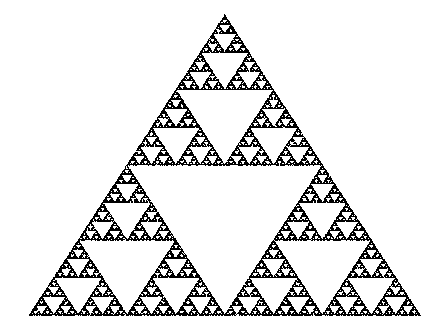
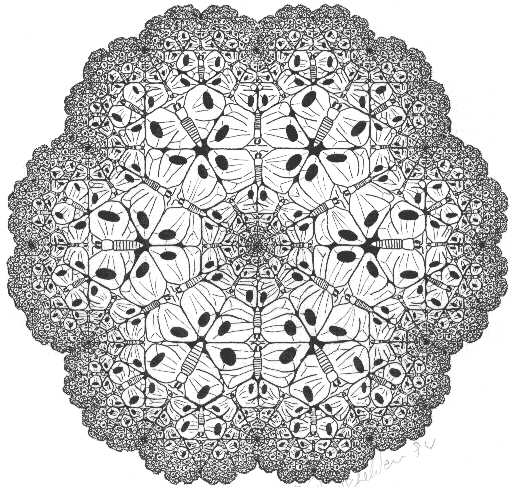
**Recursion**

Recursion is an algorithmic way to solve problems. Where iteration solves a series of problems in sequence, recursion takes more of a “divide and conquer” approach.

This “divide and conquer” approach is what makes recursion so intriguing. It continues to simplify a problem more and more into simpler parts until it cannot be simplified anymore. It does this by calling itself as a function and storing previous instances of the call in stack.



A good way to picture this is with your computer’s file directory system. You start at the base directory and then that directory could contain more directories, files, or nothing. If it does contain another directory, the function is called again, but this time it is the new base directory. This would continue until there are no further directories. This is how recursion works.



Fractals are another example of recursion. They take a large picture and make the same picture over and over again until it cannot be simplified anymore.

And now for an example of solving a piecewise function using recursion:

f(x) { f(x-3) + 4 if x > 5 }

{8 if x <= 5}

f(20) = f(17) + 4

f(17) = f(14) + 4

f(14) = f(11) + 4

f(11) = f(8) + 4

f(8) = f(5) + 4

f(5) = 8

f(20) = 8 + 4 + 4 + 4 + 4 + 4 = 28