Designing Recursive Algorithms

An algorithm that uses recursions usually has the following form:

1. **Test for base cases**
   It begins testing for a set of base cases (there should be at least one). These base cases should be defined so that every possible chain of recursive calls will eventually reach a base case.
2. **Recur**
   If not a base case, we perform one or more recursive calls.

To design a recursive algorithm for a given problem, it is useful to think of the different ways we might define subproblems that have the same general structure as the original problem. If you are having a problem in determining the repetitive structure, it is sometimes useful to work out the problem on a few concrete examples to see how the subproblems are defined.

Source: **Problem Solving and Algorithms in Python** (http://interactivepython.org/runestone/static/pythonds/index.html#) from Bradley Miller on **www.interactivepython.org** (http://interactivepython.org/runestone/static/pythonds/index.html#).