Reasons to Re-write your Recursive Functions using Loops

This document gives a better understanding that recursion is not always the better option.

Recursion is one of the coolest ideas in the introduction to computer science course. It may be one of the coolest ideas in computer science generally. Because of this, students tend to write a lot of programs using recursion, and these programs often work perfectly. However there are a number of very important reasons to get used to re-writing these programs using loops rather than recursion.

1. Practice

The most straightforward of the reasons is to get practice reading and writing the same code in both a recursive and a looping form. First, this builds your abilities as a programmer. Second, some languages work better with one format or the other, and it is very important, in the long run, to be able to fluently use both.

2. Performance

In the case of Python, and in fact many of the common languages used today, recursive solutions perform much more slowly than looping solutions to the same problem. This is because of the way function calls are stored in a stack in these languages and because of the number of hidden steps required to call a function within anothe function.

3. Debugging

Because of the complicated steps required to call each function, when a bug shows up many layers down in the recursion, it is often difficult to read and understand the stack trace which leads you to where the bug occured. By rewriting the program using loops, you simplify the debugging process for yourself and for anyone else who might use the code in the future.

If you like this example, and are interested in seeing the principle discussed in greater depth and with other problems, Polya's book is an excellent place to look.