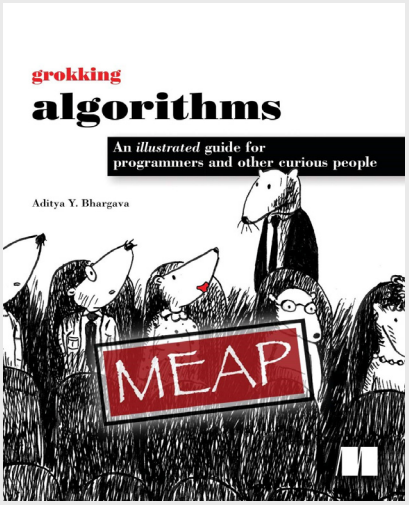

Understanding the concept of recursion

By Aditya Y. Bhargava

There are many important algorithms that use recursion, so it is important to understand the concept. In this article, we explore the concept of recursion.

This article is an excerpt from Grokking Algorithms by Aditya Y. Bhargava. Save 39% on Grokking Algorithms *with code 15dzamia at manning.com.*

Suppose you're digging through your grandma's attic and come across a mysterious locked suitcase:

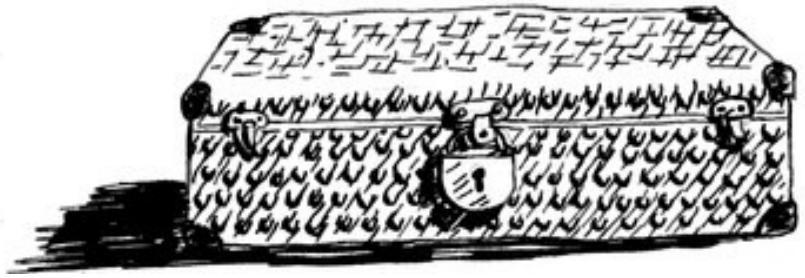

Figure 1

She tells you that the key for the suitcase is probably in this other box:

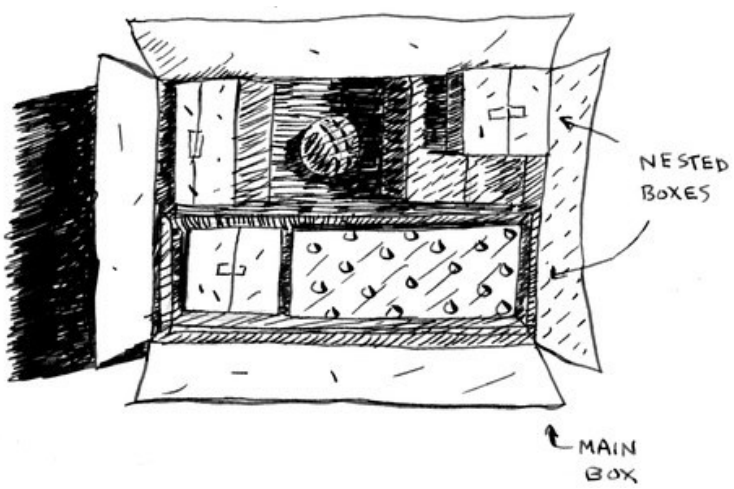

Figure 2

This box has more boxes, with more boxes inside those boxes. The key is in a box somewhere. What's your algorithm to search for this key? Think of an algorithm before you read on.

Here's one approach:

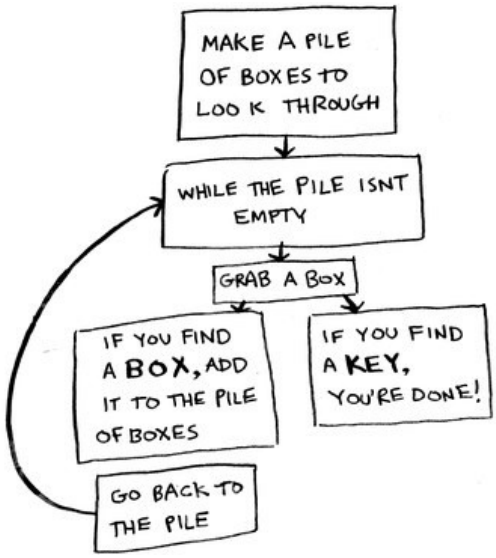

Figure 3

1. Make a pile of boxes to look through
2. Grab a box and look through it
3. If you find a box, add it to the pile to look through later
4. If you find a key, you're done!
5. Repeat!

Here's an alternate approach:

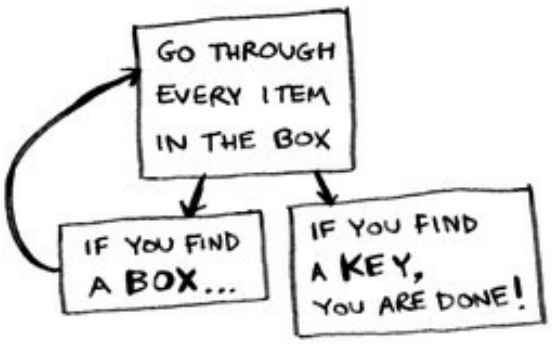

Figure 4

1. Look through the box
2. If you find a box, go to step 1.
3. If you find a key, you're done!

Which approach seems easier to you? The first approach uses a while loop. While the pile isn't empty, grab a box and look through it:

```
def look_for_key(main_box):
  pile = main_box.make_a_pile_to_look_through()
  while pile is not empty:
    box = pile.grab_a_box()
    for item in box:
      if item.is_a_box():
        pile.append(item)
      elif item.is_a_key():
        print "found the key!"
```

The second way uses recursion. Recursion is where a function calls itself.

Here's the second way in pseudocode:

```
def look_for_key(box):
  for item in box:
    if item.is_a_box():
      look_for_key(item) ❶ recursion!
    elif item.is_a_key():
      print "found the key!"
```

Both approaches accomplish the same thing, but the second approach is clearer to me. Recursion is used when it makes the solution clearer. There is no performance benefit to using recursion; in fact, loops are sometimes better for performance. I like this quote by Leigh Caldwell on StackOverflow[1]: "Loops may achieve a performance gain for your program. Recursion may achieve a performance gain for your programmer. Choose which is more important in your situation!"

There are many important algorithms that use recursion, so it is important to understand the concept.

Figure 5

[1] http://stackoverflow.com/a/72694/139117