编程界有一个非常经典的话”迭代是人,递归是神”来从宏观上对二者进行把握.

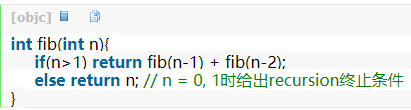
从概念上来讲,递归就是指程序调用自身的编程思想,即一个函数调用本身;迭代是利用已知的变量值,根据递推公式不断演进得到变量新值的编程思想.

从直观上讲,递归是将大问题化为相同结构的小问题,从待求解的问题出发,一直分解到已经已知答案的最小问题为止,然后再逐级返回,从而得到大问题的解;

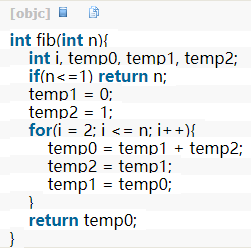
而迭代则是从已知值出发,通过递推式,不断更新变量新值,一直到能够解决要求的问题为止.

以斐波那契数列的求解为例进行两种思想的对比:

递归的实现:



迭代的实现：



下面就对队规和迭代进行比较：

递归实际上不断地深层次调用函数，直到函数有返回才会逐层的返回，因此，递归涉及到**运行时的堆栈开销**（参数必须压入堆栈保存，直到该层函数调用返回为止），所以有可能导致堆栈溢出的错误；但是递归编程**所体现的思想**正是人们追求节俭、将问题交给计算机，以及将大问题分解成相同小问题从而解决大问题的动机。

迭代大部分时候需要人为的对问题进行剖析，将问题转变成一次次的迭代来逼近答案。迭代不像递归一样对堆栈有一定的要求，另外一旦问题剖析完毕，就可以很容易的通过循环加以实现。迭代的效率高，但不太容易理解，当遇到数据结构的设计时，比如“二叉树，图”等问题时，使用就比较困难。用递归就能省掉人工思考解法的过程，只需要不断地将问题进行分解直到返回就可以了。

所以从执行效率来说，底层（迭代）往往比高层（递归）来的高，但高层却更加简洁。