[循环不变式（Loop invariant）](http://blog.csdn.net/zj060607/article/details/7533119)

分类： [数据结构](http://blog.csdn.net/zj060607/article/category/561970) [算法](http://blog.csdn.net/zj060607/article/category/1138961)2012-05-03 22:32 359人阅读 [评论](http://blog.csdn.net/zj060607/article/details/7533119#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/zj060607/article/details/7533119#report)

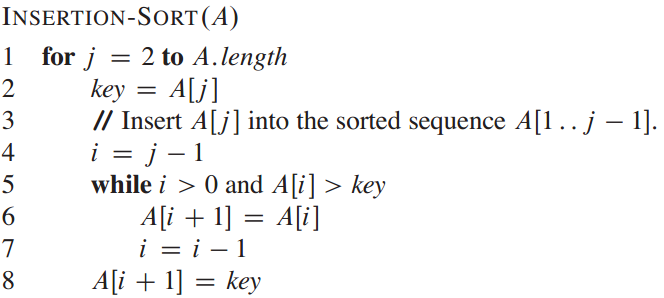
[算法](http://blog.csdn.net/tag/details.html?tag=%e7%ae%97%e6%b3%95)[initialization](http://blog.csdn.net/tag/details.html?tag=initialization)[properties](http://blog.csdn.net/tag/details.html?tag=properties)[algorithm](http://blog.csdn.net/tag/details.html?tag=algorithm)[c](http://blog.csdn.net/tag/details.html?tag=c)

循环不变式（Loop invariant）  
个人理解，欢迎讨论  
**背景知识：**  
Hoare 逻辑(也叫做Floyd–Hoare 逻辑)是英国计算机科学家 东尼·霍尔 开发的形式系统，随后为 Hoare 和其他研究者所精制。它发表于东尼·霍尔 1969年的论文"计算机程序的公理基础"中。这个系统的用途是为了使用严格的数理逻辑推理计算机程序的正确性提供一组逻辑规则。  
Hoare 逻辑的中心特征是Hoare 三元组。这种三元组描述一段代码的执行如何改变计算的状态。Hoare 三元组有如下形式  
http://my.csdn.net/uploads/201205/03/1336055726_7497.png  
这里的 P 和 Q 是断言而 C 是命令。P 叫做前条件而 Q 叫做后条件。断言是谓词逻辑的公式。这个三元组在直觉上读做: 只要 P 在 C 执行前的状态下成立，则在执行之后 Q 也成立。  
在loop中的应用，横线上下表示同一个意思:  
http://my.csdn.net/uploads/201205/03/1336055323_4849.png

横线上面一行表示,只要{P∧B}在 S 执行前的状态下成立，则在执行之后 P 也成立;  
横线下面一行表示，只要{P}在 while B DO S done 执行前的状态下成立，则在执行之后 {ùB∧P} 也成立;  
eg.  
http://my.csdn.net/uploads/201205/03/1336055381_7429.png  
————————————————————————————————  
http://my.csdn.net/uploads/201205/03/1336055444_5812.png  
  
化简得：  
http://my.csdn.net/uploads/201205/03/1336055641_3518.png

**定义：**

In computer science, a loop invariant is an invariant used to prove properties of loops.

循环不变量，是一个用于证明循环属性的不变量。（直白一点，不管循环多少次，这个属性都不变。eg.在插入排序中，不管循环多少次A[1..j-1]都是有序的）  
    1. Initialization: It is true prior to the first iteration of the loop.  
    2. Maintenance: If it is true before an iteration of the loop, it remains true before the next iteration.  
    3. Termination: When the loop terminates, the invariant gives us a useful property that helps show that the algorithm is correct.  
      
　　1、初始：保证在初始的时候不变式为真。  
　　2、保持：如果在循环的某一次迭代开始之前它是正确的，那么，在下一次迭代开始之前，它也应该保持正确。  
　　3、终止：当循环结束时，不变式给我们一个有用的性质，他有助于表明算法算法是正确的。   
  
**作用：**循环不变式，主要帮助我们理解算法的正确性。  
eg.  
《算法导论》第二章，证明插入算法的正确性。  
1. 初始化（循环第一次迭代之前）的时候，j=2,A[1 ‥ j -1](即A[1])的“有序性”是成立的；  
2. 在循环的每次迭代过程中，A[1 ‥ j -1]的“有序性”仍然保持；  
3. 循环结束的时候，A[1 ‥ j -1]的“有序性”仍然成立。  
  
此处的P，即 A[1 ‥ j -1]的“有序性”保持成立。每一迭代，前j-1个元素的有序性都成立，那么，插入排序当然可以保证执行之后数组中元素的有序性了。  
**尚未明白：**  
     Partial and total correctness  
个人感觉，很多跟遍历有关的算法（二分查找能用吗？），都用可以用循环不变量的思想来验证～～  
**参考文章：**  
http://en.wikipedia.org/wiki/Hoare\_logic  
http://en.wikipedia.org/wiki/Loop\_invariant  
《算法导论》  
http://bbs.chinaunix.net/thread-750480-1-1.html