**冒泡排序算法的正确性**

2008-04-24 10:08:28|  分类： [程序设计资料](http://ufownl.blog.163.com/blog/#m=0&t=1&c=fks_081064083095084069082087074066086083081066086084)|字号 订阅

伪代码：

BUBBLE-SORT(A)

1 for i<--1 to Length(A)

2   do for j<--Length(A) downto i+1

3      do if A[j]<A[j-1]

4         then Swap(A[j], A[j-1])

假设Length(A)=n。

第一部分：

对于2~4行的for循环，有循环不变式：每次迭代之前，A[j]=min{A[k] | j<=k<=n}，并且子数组A[j..n]中的元素还是最初在A[j..n]中的元素。

初始化：第一次迭代之前j=n，子数组A[j..n]只包含一个元素A[n]，循环不变式显然成立。

保持：迭代一个给定值的j。首先假设此次迭代前循环不变式成立，那么根据循环不变式，A[j]是A[j..n]中最小的元素。第3~4行代码表示如果A[j]<A[j-1]就交换A[j]和A[j-1]，显然这使得A[j-1]成为A[j-1..n]中最小的元素。由于唯一改变子数组A[j-1..n]的操作仅仅是那次可能发生的交换操作，且在迭代开始时，A[j..n]中的元素最初都是在A[j..n]中，所以在迭代开始时A[j-1..n]中的元素最初都是在A[j-1..n]中。j自减，准备开始进入下一轮迭代。

终止：循环终止时j=i。根据循环不变式，A[j]=A[i]=min{A[k] | i<=k<=n}并且A[i..n]中的元素最初都在A[i..n]中。

所以在2~4行的for循环执行结束过后，A[i]将是A[i..n]中最小的元素。

第二部分：

对于1~4行的for循环，有循环不变式：每次迭代之前，子数组A[1..i-1]包含了A[1..n]中前i-1小的所有元素，并且它们是已排好序的。A[i..n]中包含了A[1..n]中余暇的元素。

初始化：第一次迭代之前i=1。子数组A[1..i-1]为空，循环不变式显然成立。

保持：迭代一个给定值的i。首先假设此次迭代前循环不变式成立，那么根据循环不变式，A[1..i-1]包含了A[1..n]中前i-1小的所有元素，并且它们是已排好序的。第一部分已经证明：在执行2~4行的for循环后A[i]是A[i..n]中最小的元素。所以在执行了2~4行的for循环后A[1..i]中就包含了A[1..n]中前i小的所有元素，并且它们已经排好序。子数组A[i+1..n]就包含了n-i个A[1..n]中余下的元素。

终止：循环终止时i=n+1 => i-1=n。所以根据循环不变式，A[1..i-1]óA[1..n]中包含了A[1..n]中前i-1小的元素（即A[1..n]的全部元素），并且它们是排好序的。

所以在1~4行的for循环执行结束过后，A[1..n]将是有序的。

根据第二部分的结论，外层for循环结束后，数组A[1..n]中的元素将是排好序的。故冒泡排序算法是正确的。