

对插入排序的改进计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/645/2021\\_2022\\_\\_E5\\_AF\\_B9\\_E6\\_8F\\_92\\_E5\\_85\\_A5\\_E6\\_c97\\_645743.htm](https://www.100test.com/kao_ti2020/645/2021_2022__E5_AF_B9_E6_8F_92_E5_85_A5_E6_c97_645743.htm) 插入排序的性能大家都了解了，时间复杂度是 $O(n^2)$ ，有没有办法提升他的效率呢？这里有一个方法，在宏观上可以将插入排序的时间复杂度降低到 $n\log n$ 。其思想如下，插入排序中每次将本次取到的数据插入到已排序序列的时候，都会将有序序列中大于这个数据的元素依次向后移动一个单元，这个过程的时间复杂度就是 $n$ ，有没有办法简化这个过程呢，其实有一种方法：因为待插入的序列是有序的，所以我们可以使用一个二分查询定位出新元素要插入的位置(插入到这个位置后，这个序列依然保持有序，这个操作的时间复杂度为 $\log n$ )，知道这个位置之后就好办了，我们可以将这个位置后面的序列整块的向后移动一个位置，这个操作称为memmove整块的移动内存单元(这样，移动元素的时间复杂度就变成1了)，因为这时候数组的位置已经腾出来了，我们可以将新的元素插入到指定位置(这个操作复杂度为1)。因此，整个插入的操作时间复杂度为 $\log n^2$ ，常数项可以忽略，因此复杂度为 $\log n$ ，因为整个排序需要 $n$ 次插入操作，那么整个算法的复杂度就为 $n\log n$ 从宏观上看，这个算法的时间复杂度为 $n\log n$ ，当然具体的性能还要看memmove这个操作的效率。一个C语言的实现如下 1

#include 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)