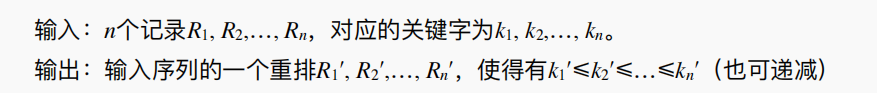
排序

一、基本概念

**排序**（Sort），就是重新排列表中的元素，使表中的元素满⾜**按关键字有序**的过程。

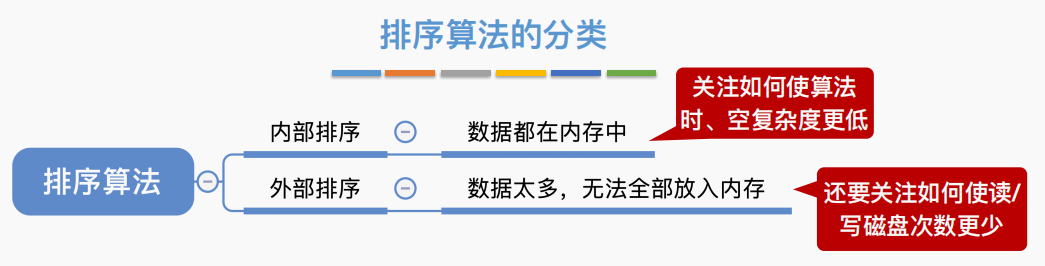


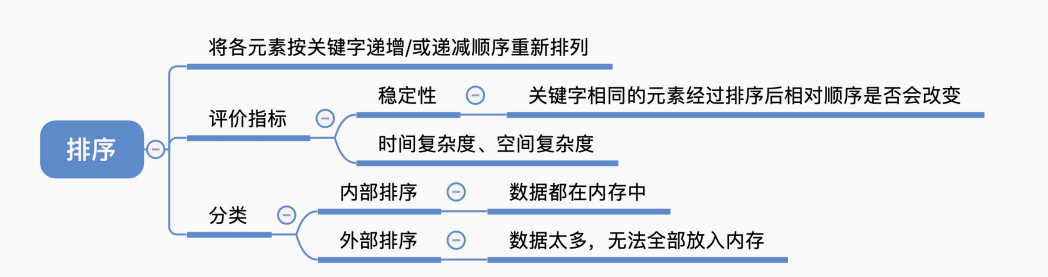
排序算法的评价指标：

**算法的稳定性**

若待排序表中有两个元素Ri和Rj，其对应的关键字相同即keyi = keyj，且在排序前Ri在Rj的前⾯，若使⽤某⼀排序算法排序后，Ri仍然在Rj的前⾯，则称这个排序算法是稳定的，否则称排序算法是不稳定的。

排序算法的分类：





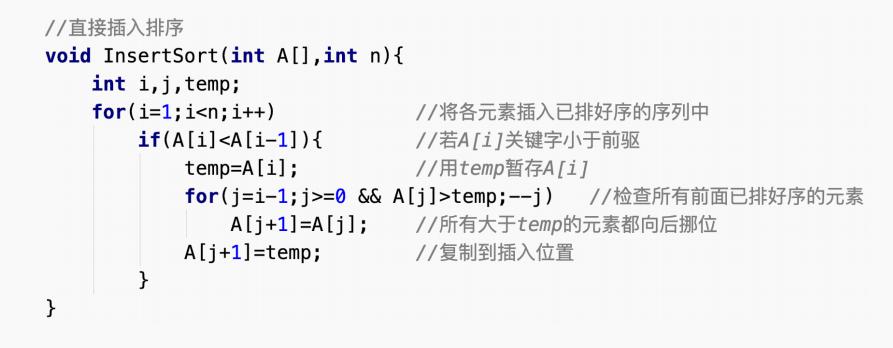
1. 插入排序

算法思想

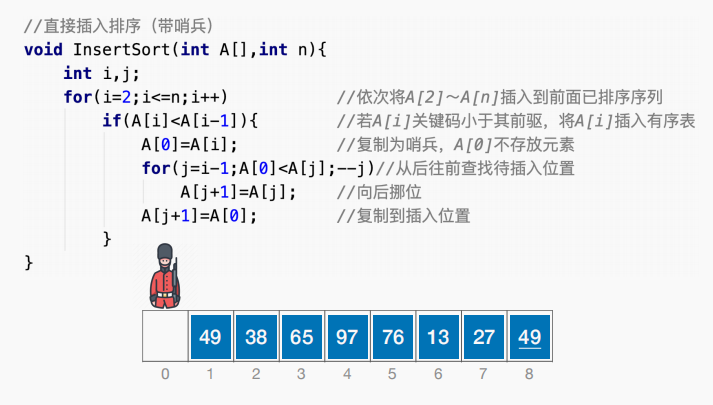
每次将⼀个**待排序的记录**按其关键字⼤⼩**插⼊到前⾯已排好序的⼦序列**中，

直到全部记录插⼊完成。

算法实现：



算法实现（带哨兵）：



算法效率分析

空间复杂度：O(1)

时间复杂度：主要来⾃**对⽐关键字**、**移动元素**若有 n 个元素，则需要 n-1 趟处理

**最好**情况：原本就**有序**

共n-1趟处理，每⼀趟只需要对⽐关键字1次，不⽤移动元素

最好时间复杂度—— O(n)

**最坏**情况：原本为**逆序**

第 i 趟：对⽐关键字 i+1次，移动元素 i+2 次

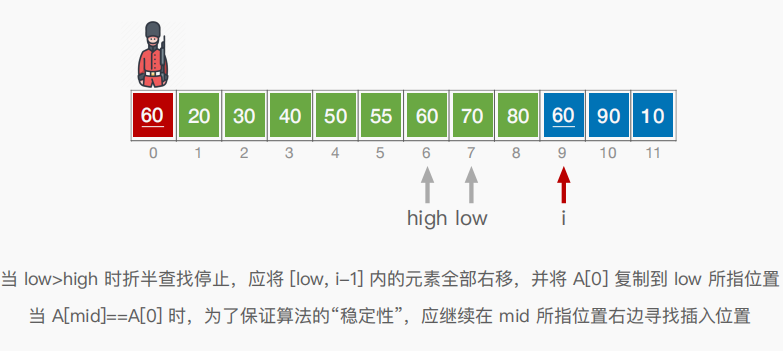
最坏时间复杂度——O(n2)

平均时间复杂度：O(n2)

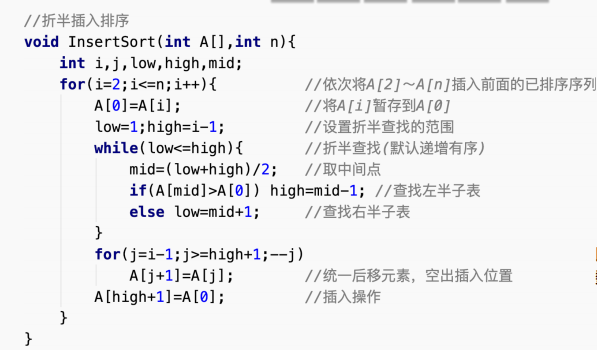
算法稳定性：**稳定**

优化——折半插⼊排序

思路：先⽤**折半查找**找到应该**插⼊的位置**，再移动元素



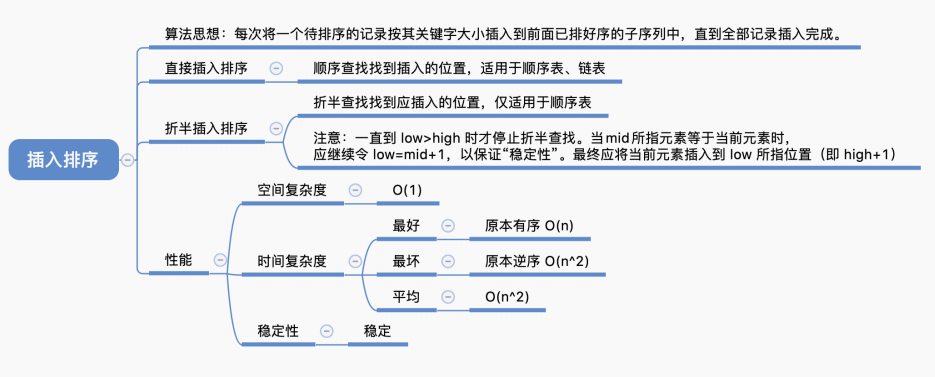
算法实现



效率分析：⽐起“直接插⼊排序”，**⽐较关键字的次数减少**了，但是移**动元素的次数没变**，整体来看时间复杂度依然是**O(n2)**

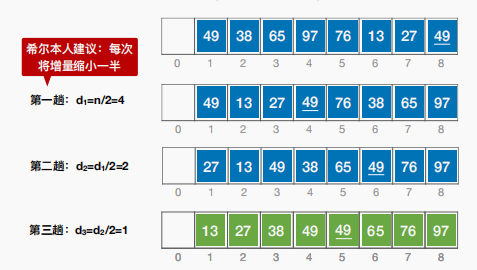
对**链表**进⾏**插⼊**排序

**移动元素的次数变少**了，但是**关键字对⽐的次数依然是O(n2)** 数量级，整体来看时间复杂度依然是**O(n2)**

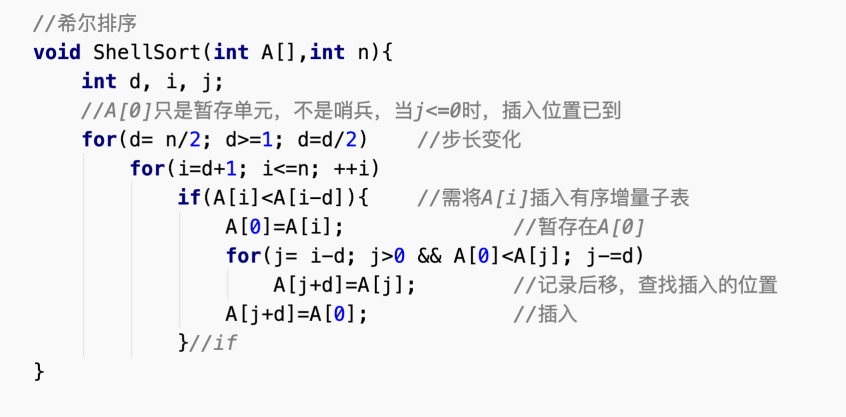


1. 希尔排序

思路：先将待排序表**分割**成若⼲形如 L[i, i + d, i + 2d,…, i + kd] 的“特殊”⼦表，对**各个⼦表**分别进⾏**直接插⼊排序**。**缩⼩增量d**，重复上述过程，**直到d=1为⽌**。



算法实现:



算法性能分析:

空间复杂度：O(1)

时间复杂度：

和增量序列 d1, d2, d3… 的选择有关，⽬前⽆法⽤数学⼿段证明确切的时间复杂度

最坏时间复杂度为 O(n2)，当n在某个范围内时，可达O(n1.3)

算法稳定性：**不稳定**

适用性：仅适⽤于**顺序表**，不适⽤于链表

