**插入排序的原理**

插入排序（Insertion Sort）是一种简单直观的排序算法。它的工作原理是通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。插入排序在实现上，通常采用in-place排序（即只需用到*O*(1)的额外空间的排序），因而在从后向前扫描过程中，找到排序位置后，需要将已排序元素逐步向后挪位，为最新元素提供插入空间。

**具体步骤如下**：

1. 从第一个元素开始，该元素可以认为已经被排序。
2. 取出下一个元素，在已经排序的元素序列中从后向前扫描。
3. 如果该元素（已排序）大于新元素，将该元素移到下一位置。
4. 重复步骤3，直到找到已排序的元素小于或者等于新元素的位置。
5. 将新元素插入到该位置后。
6. 重复步骤2~5，直到所有元素均排序完毕。

**复杂度分析**

**时间复杂度**：

* **最好情况**：当输入数组已经是排序状态时，插入排序的时间复杂度为*O*(*n*)。因为每个元素都只需要比较一次（与已经排序好的最后一个元素比较），并直接插入到最后位置。
* **最坏情况**：当输入数组是逆序排列时，插入排序的时间复杂度为*O*(*n*2)。因为每个元素都需要与前面所有元素进行比较，并且移动到最前面位置。
* **平均情况**：时间复杂度同样是*O*(*n*2)。尽管在某些情况下（如部分排序的数组）可能接近*O*(*n*)，但总体上平均性能与最坏情况相似。

**空间复杂度**：

* 插入排序是in-place排序，它只需要*O*(1)的额外空间来存储临时变量（如当前需要插入的元素）。这意味着它不需要额外的存储空间来保存数据副本。

**总结**

插入排序因其实现简单、在小规模数据集上效率高而受到青睐。然而，由于其平均和最坏情况下的时间复杂度都是*O*(*n*2)，它在大规模数据集上表现不佳。因此，在实际应用中，对于大规模数据集，通常会选择更高效的排序算法，如快速排序、归并排序等。