



插入排序 Insertion Sort

在日常生活中，排序是一个十分常见的数学问题。以最简单的例子而言，当你的面前摆有一堆无序的数字需要被排序时，此时的你一定会根据自己的经验想到最适合自己的方法。在众多方法中，插入排序无疑是最简单的排序方法之一。

插入排序 Insertion Sort

1. 初始化 $p = 1$
2. 对前 p 个元素进行排序。
3. 在列表中正确插入第 $(p + 1)$ 个元素，使现在的 $p + 1$ 个元素排序。
4. 增加 p 并转到步骤 3

插入排序的原理

- 向左扫描，将每一个较大的元素向右移动一个位置，从而为新元素腾出空间
- 停止条件
 1. 当找到较小或相等的元素
 2. 到达左边界
- 将新元素移入

代码实现

```
1 // 伪代码实现
2 INSERTION-SORT(A)
3   FOR p = 1 TO n-1
4       key = A[p]
5       i = p - 1
6       WHILE i ≥ 0 AND A[i] > key
7           A[i+1] = A[i]
8           i = i - 1
9       A[i+1] = key
```

```
1 // java实现
2 public static void insertionSort(int[] A) {
3     for(int p=0; p<A.length; p++) {
4         int key = A[p];
5         int i = p-1;
6         while(i≥0 && A[i]>key) {
7             A[i+1] = A[i];
8             i--;
9         }
```

```
10         A[i+1] = key;
11     }
12 }
```

复杂度分析

运行情况	复杂度
最差情况	$O(n^2)$
最好情况	$O(n)$
平均情况	$O(n^2)$

插入排序的准确运行时间无法提前预测，而运行时间主要取决于输入。因此根据其平均运行时间，它被认为是一种 $O(n^2)$ 算法。