

## 2 Insert Sort

### 插入排序

问题:

将无序序列  $s$  进行排序。

解法:

本问题对无序序列  $s$  进行升序排序，排序后  $s$  是从小到大的。

将序列  $s$  分为左右两个部分  $left$  和  $right$ ， $left$  是有序的， $right$  是无序的。对  $right$  中最左边的元素  $x$ ，在  $left$  中找到一个位置（下标） $i$ ，满足  $s[i-1] \leq x \leq s[i]$ ，也就是说  $x$  可以夹在  $s[i-1]$  和  $s[i]$  之间。由于  $left$  是有序的，插入  $x$  之后仍然有序，只需要将  $left$  中包括  $i$  之后的元素向右移动一个位置，将  $x$  放置在原  $s[i]$  的位置即可。

对于下面这个长度为 10 的无序序列  $s$  进行升序排序。初始时将  $s[0]$  单独作为有序的  $left$ ， $s[1-9]$  作为无序的  $right$ 。

0	1	2	3	4	5	6	7	8	9
45	23	1	90	57	33	81	12	74	16

(1) 将  $right$  左边第一个元素  $s[1]$  插入  $left$ ，满足  $s[i-1] \leq s[1] \leq s[i]$  的  $i = 0$ ，但因为数组中没有  $s[-1]$ ，可以假设  $s[-1] = -\infty$ ，这样  $s[-1]$  必然小于  $s$  中的所有元素（如果是降序排序，即从大到小，则需要假设  $s[-1] = +\infty$ ）；

0	1	2	3	4	5	6	7	8	9
23	45	1	90	57	33	81	12	74	16

(2) 重复这个过程直到  $right$  中所有元素都被插入  $left$  中，算法结束；

0	1	2	3	4	5	6	7	8	9
1	12	16	23	33	45	57	74	81	90

对长度为  $n$  的序列  $s$ ，每一轮将  $right$  中一个元素插入  $left$  中的时间为  $O(n)$ ，总共需要  $n$  轮操作，该算法的时间复杂度为  $O(n^2)$ 。