

# 插入排序算法及C语言实现

**插入排序算法**是所有排序方法中最简单的一种算法，其主要的实现思想是将数据按照一定的顺序一个一个的插入到有序的表中，最终得到的序列就是已经排序好的数据。

**直接插入排序**是插入排序算法中的一种，采用的方法是：在添加新的记录时，使用**顺序查找**的方式找到其要插入的位置，然后将新记录插入。

很多初学者所说的插入排序，实际上指的就是**直接插入排序算法**，插入排序算法还包括**折半插入排序**、**2-路插入排序**，**表插入排序**和**希尔排序**等，后序文章都会一一讲到。

例如采用直接插入排序算法将无序表 {3,1,7,5,2,4,9,6} 进行升序排序的过程为：

- 首先考虑记录 3，由于插入排序刚开始，有序表中没有任何记录，所以 3 可以直接添加到有序表中，则有序表和无序表可以如图 1 所示：



图 1 直接插入排序 (1)

- 向有序表中插入记录 1 时，同有序表中记录 3 进行比较， $1 < 3$ ，所以插入到记录 3 的左侧，如图 2 所示：



图 2 直接插入排序 (2)

- 向有序表插入记录 7 时，同有序表中记录 3 进行比较， $3 < 7$ ，所以插入到记录 3 的右侧，如图 3 所示：



图 3 直接插入排序 (3)

- 向有序表中插入记录 5 时，同有序表中记录 7 进行比较， $5 < 7$ ，同时  $5 > 3$ ，所以插入到 3 和 7 中间，如图 4 所示：



图 4 直接插入排序 (4)

- 向有序表插入记录 2 时，同有序表中记录 7 进行比较， $2 < 7$ ，再同 5，3，1 分别进行比较，最终确定 2 位于 1 和 3 中间，如图 5 所示：



图 5 直接插入排序 (5)

- 照此规律，依次将无序表中的记录 4，9 和 6 插入到有序表中，如图 6 所示：



图 6 依次插入记录4， 9和6

直接插入排序的具体代码实现为：

```
01.  #include <stdio.h>
02.  //自定义的输出函数
03.  void print(int a[], int n ,int i){
04.      printf("%d:",i);
05.      for(int j=0; j<8; j++){
06.          printf("%d",a[j]);
07.      }
08.      printf("\n");
```

```

09.     }
10.     //直接插入排序函数
11.     void InsertSort(int a[], int n)
12.     {
13.         for(int i= 1; i<n; i++){
14.             if(a[i] < a[i-1]){ //若第 i 个元素大于 i-1 元素则直接插入；反之，需要找到适当的插入位置后在插入
15.                 int j= i-1;
16.                 int x = a[i];
17.                 while(j>-1 && x < a[j]){ //采用顺序查找方式找到插入的位置，在查找的同时，将数组中的元素后移一位
18.                     a[j+1] = a[j];
19.                     j--;
20.                 }
21.                 a[j+1] = x; //插入到正确位置
22.             }
23.             print(a,n,i); //打印每次排序后的结果
24.         }
25.     }
26.
27.     int main(){
28.         int a[8] = {3,1,7,5,2,4,9,6};
29.         InsertSort(a,8);
30.         return 0;
31.     }

```

运行结果为：

```

1:13752496
2:13752496
3:13572496
4:12357496
5:12345796
6:12345796
7:12345679

```

直接插入排序算法本身比较简洁，容易实现，该算法的[时间复杂度](#)为  $O(n^2)$ 。

插入排序的其它 4 种排序方法，在后序章节中有详细介绍。

[< 上一节](#)

[下一节 >](#)