阅读：10,663　　作者：解学武

折半插入排序算法（折半排序算法）

上一节　　下一节

上一节介绍了直接插入排序算法的理论实现和具体的代码实现，如果你善于思考就会发现该算法在查找插入位置时，采用的是顺序查找的方式，而在查找表中数据本身有序的前提下，可以使用折半查找来代替顺序查找，这种排序的算法就是折半插入排序算法。

该算法的具体代码实现为：

```
#include <stdio.h>
void print(int a[], int n ,int i){
    printf("%d:",i);
    for(int j=0; j<8; j++){
        printf("%d",a[j]);
    }
    printf("\n");
}

void BInsertSort(int a[],int size){
    int i,j,low = 0,high = 0,mid;
    int temp = 0;
    for (i=1; i<size; i++) {
        low=0;
        high=i-1;
        temp=a[i];
        //采用折半查找法判断插入位置，最终变量 low 表示插入位置
        while (low<=high) {
            mid=(low+high)/2;
            if (a[mid]>temp) {
                high=mid-1;
            }else{
                low=mid+1;
            }
        }
        //有序表中插入位置后的元素统一后移
        for (j=i; j>low; j--) {
            a[j]=a[j-1];
        }
        a[low]=temp;//插入元素
        print(a, 8, i);
```

```
32.         }
33.
34.     }
35.     int main(){
36.         int a[8] = {3,1,7,5,2,4,9,6};
37.         BInsertSort(a, 8);
38.         return 0;
39.     }
```

运行结果为：

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

折半插入排序算法相比较于直接插入排序算法，只是减少了关键字间的比较次数，而记录的移动次数没有进行优化，所以该算法的时间复杂度仍是 $O(n^2)$ 。

上一节 下一节