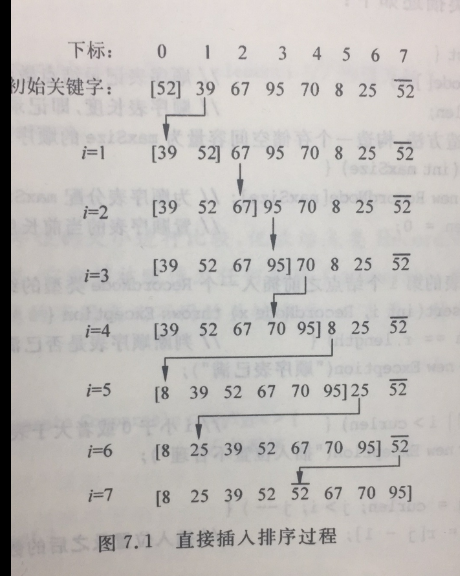
**直接插入排序**是一种简单的排序方法，其基本思想就是每趟一条待排序的记录，按照其关键字的大小插入到前面已经排好的记录序列中的适当位置，直到全部记录插入完成为止。

**基本思路**

对于n个带排列的记录序列，在插入第i个记录时，0到i-1已经时排好序的记录的有序序列。当我们插入的时候，要用i的关键字key从后向前进行比较，如果大于key就向后移动一位，直到遇到小于或等于的关键字停止，该位置为j，便找到了要插入的位置j+1。然后将第i个记录的关键字插入到j+1位置。



**步骤**

1.将nums[i]保存在临时变量temp中

因为在遇到大于nums[i]的值的关键字后移过程中可能覆盖nums[i]

2.将temp分别与nums[j]（j=i-1，i-2，… 0）依次比较，若对应关键字大于temp则后移动一位，小于或等于退出。

因为是把大于temp的关键字后移，所以必须后从向前比较。

如果是把小于temp的关键字前移，才可以从前往后遍历，同样保存排好序的数列也要从最后开始，不应该从头。

3.将temp插入到j+1.

到j的位置是小于或等于temp的，所以temp应该插入到j+1的位置。

4.重复，循环插入完所有的关键字。**注意，这里i是从1开始的，因为表头的第一个关键字不用比较就直接入排好序的子序列，直接跳过。**

在每个位置关键字插入过程中，需要比较两次

1.比较关键字位置是否有效 j>=0 ?

2.比较关键字和带插入排序的关键字的大小

带监视哨方法

在每次插入比较的时候，令nums[0]=nums[i]，同样还是遇到小于等于nums[i]停止，这样我们到nums[0]的时候就会自动停下，我们就不用比较关键字位置是否有效.但是这种方法要牺牲一个位置来保存监视哨，不能保存数据。

**代码**

不带监视哨：

public static int[] insertSort(int[] nums){

for(int i=1;i<nums.length;i++){ //第一个关键字组成排好序的序列，所以从第二个开始，也就是i=1.

int temp = nums[i];

int j=i-1;

while(j>=0&&nums[j]>temp){ //需要比较不成超出下届和nums[j]和temp的大小

//比nums[i]大的后移一位，当遇到小于等于temp的关键字停下

nums[j+1]=nums[j]; //不用担心覆盖什么数据，最次全部后移一位

// 最多只能覆盖最后一位的下一位是nums[i]，还保存在了temp中

j--; //向前遍历

}

//当全部关键字都比temp大，一直j--，一直到j=-1<0结束，要插入的位置是表头nums[j+1=0]

//当nums[j]小于等于temp才退出，所以nums[j+1]才是要插入的位置。

nums[j+1]=temp;

}

return nums;

}

带监视哨：

public static int[] insertSortWithGuard(int nums[]){

for(int i=1;i<nums.length;i++){

nums[0]=nums[i];//用nums[0]当作哨站并保存当前要排序的关键字，当遍历 到nums[0]的时候 相等，退出循环

int j=i-1; //省了一次比较j位置是否越界

//但是需要节省出一个表头空间不能保存数据

while(nums[j]>nums[0]){ //当nums[i]大于关键字的时候

nums[j+1]=nums[j];//向后移动一个位置

j--;//向前遍历

}

nums[j+1]=nums[0];

}

return nums;

}

**算法的性能分析**

1.空间复杂度

仅使用了一个辅助单元nums[0]，**空间复杂度O(1)**

2.时间复杂度

有序表中逐个插入记录的操作进行了n-1趟，每趟排序的操作分为比较关键字和移动记录，而比较的次数和移动记录的次数取决与待排序列关键字的初始状态。

最好的情况是当带排序列已经按照关键字有序的时候，每趟排序只需与记录关键字进行比较1次，两次移动。总的比较次数达到最小值，即n-1次，总的移动次数2\*(n-1)次。

最坏情况是当待排序列逆序的时候，每趟排序都要将待插入的记录插入到记录表的最前边。第i趟排序，需要进行关键字比较的次数为i，移动记录的次数为i+2.总的比较次数为 , 总的移动次数为 .

当待排序记录是随机序列的情况下，即待排序记录出现的概率相同，第i趟排序所需要比较和移动次数可取最好情况和最坏情况的平均值，约为。总的平均比较次数和移动次数约为2.

因此，**直接插入排序的时间复杂度为O(2)**

**算法稳定性：稳定**