一、首先说一下插入排序的整体思路

通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。

数组分为两部分：有序部分和无序部分。取出无序部分的第一个元素，与有序区部分进行比较，并插入到合适的位置。

二、详细图解

通过下图来了解一下整个排序过程吧，现在排序一个乱序数组，如下



1、首选取出第一个元素 10，认为10已经是有序的了，进行下一步

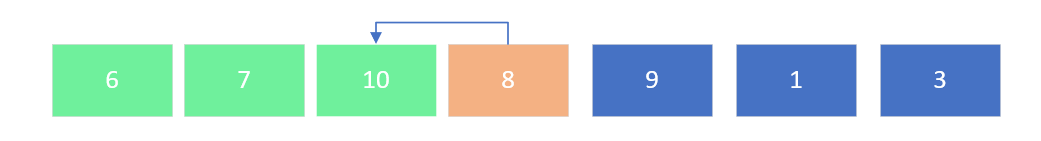
2、取出元素7，与10进行比较，比10小，交换10和7的位置，进行下一步



3、取出元素6，与10进行比较，比10小，交换位置，然后和7进行比较，比7小交换位置，进行下一步



4、取出元素8，与10比较，比10小，交换位置，与7比较，比7大，不动，进行下一步



………………………………………………………….

一直顺序重复操作，完成排序



三、代码实现

**public** **void** sort(**int**[] arr)

{

**int** length = arr.length;

**for**(**int** i=0;i<length-1;i++)

{

**if**(arr[i]>arr[i+1])

{

**for**(**int** j=i;j>=0&&arr[j]>arr[j+1];j--)

{

**int** tmp = arr[j];

arr[j]=arr[j+1];

arr[j+1] = tmp;

}

}

}

}

四、时间和空间复杂度

1、时间复杂度

最好情况，是有序数组，比较次数是n-1,不需要移动

最坏情况，需要进行n-1次排序，比较次数（n+2）(n-1)/2，移动次数（n+4）(n-1)/2

平均比较次数 （n+2）(n-1)/4,移动次数 （n+6）(n-1)/6

所以时间复杂度是O(n2)

2、空间复杂度

整个过程只有一个tmp变量，所以空间复杂度是O（1）