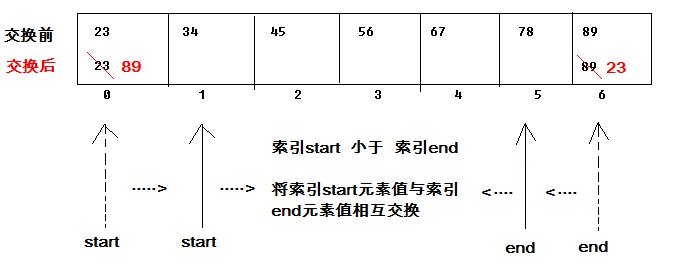
# 第N篇 数组排序

##### 1 数组元素逆序

图形分析：



代码如下：

//数组元素逆序

**public** **static** **void** receive(**int**[] arr){

**for** (**int** start = 0, end = arr.length-1; start < end; start++,end--) {

**int** temp = arr[start];

arr[start] = arr[end];

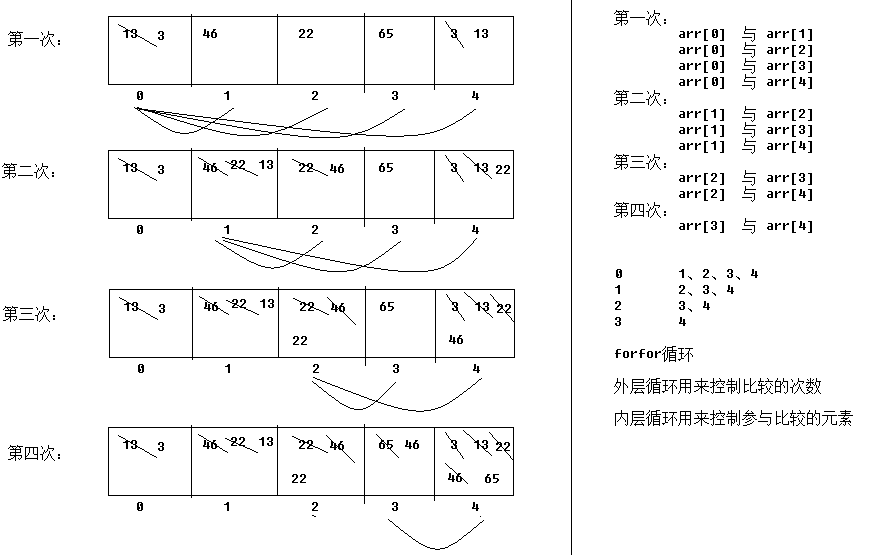
arr[end] = temp;

}

}

##### 数组元素选择排序

图解：



代码如下：

//选择排序

**public** **static** **void** selectSort(**int**[] arr) {

//功能

//外层循环用来控制数组循环的圈数

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

//内层循环用来完成元素值比较，把小的元素值互换到要比较的第一个元素中

**for** (**int** j = i+1; j < arr.length; j++) {

**if** (arr[i] > arr[j]) {

**int** temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

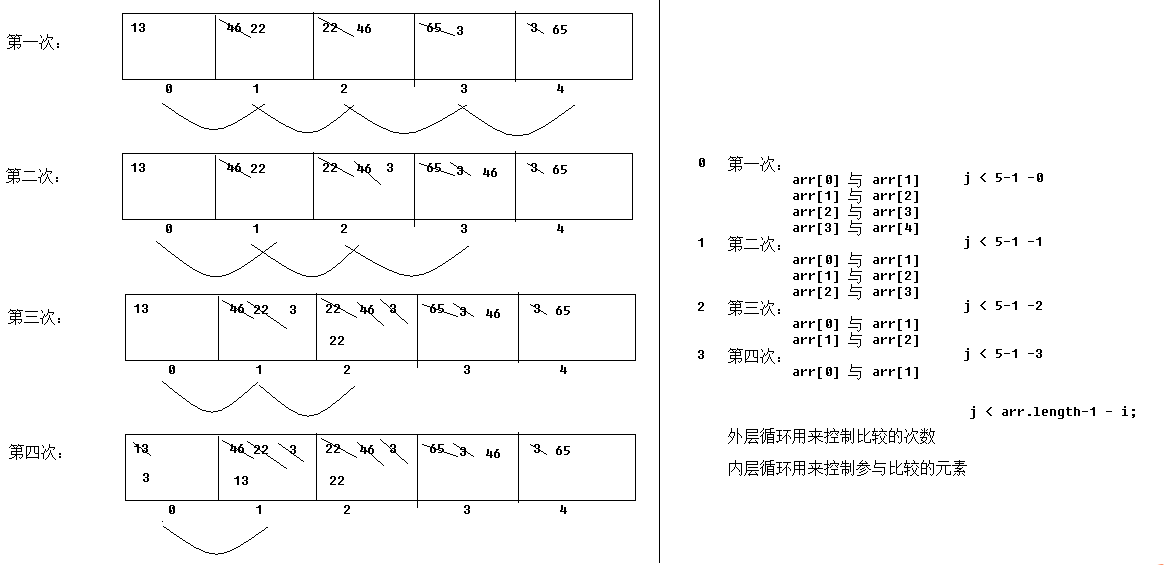
}

}

}

##### 数组元素冒泡排序

图解：



代码如下：

//冒泡排序

**public** **static** **void** bubbleSort(**int**[] arr) {

//功能

//外层循环用来控制数组循环的圈数

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

//j < arr.length-1 为了避免角标越界

//j < arr.length-1-i 为了比较效率,避免重复比较

//内层循环用来完成元素值比较，把大的元素值互换到后面

**for** (**int** j = 0; j < arr.length-1-i; j++) {

**if** (arr[j] > arr[j+1]) {

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

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

arr[j+1] = temp;

}

}

}

}