# 简单排序

## 冒泡排序

从队列的最左边开始，比较0号位置和1号位置的队员，如果左边的队员（0号）高，就让两个队员交换，如果右边的队员高，就什么也不做，然后右移一个位置，比较1号位置和2号位置的队员。和刚才一样，如果左边队员高，则两个队员交换位置。





算法：

/\*\*

\* 交换数组中两个索引位中的数据

\*/

**private** **void** swap(**int**[]sort ,**int** index0,**int** index1){

**int** temp = sort[index0];

sort[index0] = sort[index1];

sort[index1] = temp;

}

/\*\*

\* 冒泡排序法

\*/

**public** **int**[] bubbleSort(**int**[] sort){

**int** out,in;

**for**(out = sort.length-1;out>1;out--){

**for**(in=0;in<out;in++){

**if**(sort[in]>sort[in+1]){

**this**.swap(sort, in, in+1);

}

}

}

**return** sort;

}

## 选择排序







算法：

/\*\*

\* 交换数组中两个索引位中的数据

\*/

**private** **void** swap(**int**[]sort ,**int** index0,**int** index1){

**int** temp = sort[index0];

sort[index0] = sort[index1];

sort[index1] = temp;

}

/\*\*

\* 选择法排序

\* **@param** sort

\* **@return**

\*/

**public** **int**[] selectionSort(**int**[] sort){

**int** out,in,min;

**for**(out = 0; out<sort.length-1; out++){

min = out;

**for**(in=out+1;in<sort.length;in++){

**if**(sort[in]<sort[min]){

min = in;

}

}

**this**.swap(sort, out, min);

}

**return** sort;

}

## 插入排序



算法：

/\*\*

\* 插入排序

\*/

**public** **int**[] insertionSort(**int**[] sort){

**int** out,in;

**for**(out = 1; out<sort.length; out++){

**int** temp = sort[out];

in = out;

**while**(in>0&&sort[in-1]>=temp){

sort[in] = sort[in-1];

in--;

}

sort[in] = temp;

}

**return** sort;

}