以下内容节选自 Java 私塾自编经典教材：

下面是 Java 实现的一些常见排序算法。

1：冒泡排序

对几个无序的数字进行排序，比较常用的方法是冒泡排序法。冒泡法排序是一个比较简单的排序方法，在待排序的数列基本有序的情况下排序速度较快。

基本思路：对未排序的各元素从头到尾依次比较相邻的两个元素是否逆序（与欲排顺序相反），若逆序就交换这两元素，经过第一轮比较排序后便可把最大（或最小）的元素排好，然后再用同样的方法把剩下的元素逐个进行比较，就得到了你所要的顺序。

可以看出如果有 N 个元素，那么一共要进行 N-1 轮比较，第 I 轮要进行 N-I 次比较。（如：有 5 个元素，则要进行 5-1 轮比较。第 3 轮则要进行 5-3 次比较）

示例如下：

```
public class Test {
    public static void main(String[] args) {
        //需要排序的数组，目前是按照升序排列的
        int a[] = new int[5];
        a[0] = 3;
        a[1] = 4;
        a[2] = 1;
        a[3] = 5;
        a[4] = 2;

        //冒泡排序
        for(int i=0;i<a.length;i++){
            for(int j=i+1;j<a.length;j++){//注意j的开始值是i+1，因为按照排序规则，比a[i]大的值都应该在它后面
                if(a[i] > a[j]){
                    int temp = a[j];
                    a[j] = a[i];
                    a[i] = temp;
                }
            }
        }
        //检测一下排序的结果
        for(int i : a){
            System.out.println("i="+i);
        }
    }
}
```

运行结果：

```
i=1
i=2
```

```
i=3
i=4
i=5
```

如果你想要按照降序排列，很简单，只需把：**if**(a[i] > a[j])改成：**if**(a[i] < a[j])就可以了。

2：选择排序

基本思路：从所有元素中选择一个最小元素 a[i]放在 a[0]（即让最小元素 a[i]与 a[0]交换），作为第一轮；第二轮是从 a[1]开始到最后的各个元素中选择一个最小元素，放在 a[1]中；……依次类推。n 个数要进行（n－1）轮。比较的次数与冒泡法一样多，但是在每一轮中只进行一次交换，比冒泡法的交换次数少，相对于冒泡法效率高。

示例如下：

```
public class Test {
    public static void main(String[] args) {
        //需要排序的数组，目前是按照升序排列的
        int a[] = new int[5];
        a[0] = 3;
        a[1] = 4;
        a[2] = 1;
        a[3] = 5;
        a[4] = 2;

        //选择法排序
        int temp;
        for (int i = 0; i<a.length; i++) {
            int lowIndex = i;
            //找出最小的一个的索引
            for (int j=i+1;j<a.length;j++) {
                if (a[j] < a[lowIndex]) {
                    lowIndex = j;
                }
            }
            //交换
            temp=a[i];
            a[i]=a[lowIndex];
            a[lowIndex]=temp;
        }

        //检测一下排序的结果
        for(int i : a){
            System.out.println("i="+i);
        }
    }
```

```
}
```

运行结果：

```
i=1
i=2
i=3
i=4
i=5
```

如果你想要按照降序排列，很简单，只需要把： `if (a[j] < a[lowIndex])`这句话修改成：`if (a[j] > a[lowIndex])`就可以了。

3：插入法排序

基本思路：每拿到一个元素，都要将这个元素与所有它之前的元素遍历比较一遍，让符合排序顺序的元素挨个移动到当前范围内它最应该出现的位置。

举个例子来说，就用前面的数组，我们要对一个有 5 个元素的数组进行升序排列，假设第一个元素的值被假定为已排好序了，那么我们就将第 2 个元素与数组中的部分进行比较，如果第 2 个元素的值较小，则将它插入到第 1 个元素的前面，现在就有两个元素排好序了，我们再将没有排序的元素与排好序的元素列表进行比较，同样，如果小于第一个元素，就将它插入到第一个元素前面，但是，如果大于第一个元素的话，我们就将它再与第 2 个元素的值进行比较，小于的话就排在第 2 个元素前面，大于的话，就排在第 2 个元素的后面。以此类推，直到最后一个元素排好序。

示例如下：

```
public class Test {
    public static void main(String[] args) {
        // 需要排序的数组，目前是按照升序排列的
        int a[] = new int[5];
        a[0] = 3;
        a[1] = 4;
        a[2] = 1;
        a[3] = 5;
        a[4] = 2;

        // 插入法排序
        int temp;
        for (int i = 1; i < a.length; i++) {// i=1开始，因为第一个元素认为是已经排好序了的
            for (int j = i; (j > 0) && (a[j] < a[j - 1]); j--) {
                //交换
                temp = a[j];
                a[j] = a[j - 1];
                a[j - 1] = temp;
            }
        }
        // 检测一下排序的结果
```

```
        for (int i : a) {
            System.out.println("i=" + i);
        }
    }
}
```

运行结果：

```
i=1
i=2
i=3
i=4
i=5
```

如果你想要按照降序排列，很简单，只需要把： a[j] < a[j - 1]这句话修改成：a[j] > a[j - 1]就可以了。

4：希尔(Shell)法排序

从前面介绍的冒泡排序法，选择排序法，插入排序法可以发现，如果数据已经大致排好序的时候，其交换数据位置的动作将会减少。例如在插入排序法过程中，如果某一整数 d[i]不是较小时，则其往前比较和交换的次数会更少。如何用简单的方式让某些数据有一定的大小次序呢？Donald Shell（Shell 排序的创始人）提出了希尔法排序。

基本思路：先将数据按照固定的间隔分组，例如每隔 4 个分成一组，然后排序各分组的数据，形成以分组来看数据已经排序，从全部数据来看，较小值已经在前面，较大值已经在后面。将初步处理了的分组再用插入排序来排序，那么数据交换和移动的次数会减少。可以得到比插入排序法更高的效率。

示例如下：

```
public class Test {
    public static void main(String[] args) {
        // 需要排序的数组，目前是按照升序排列的
        int a[] = new int[5];
        a[0] = 3;
        a[1] = 4;
        a[2] = 1;
        a[3] = 5;
        a[4] = 2;

        // shell法排序
        int j = 0;
        int temp = 0;
        //分组
        for (int increment = a.length / 2; increment > 0; increment /= 2)
{
            //每个组内排序
            for (int i = increment; i < a.length; i++) {
                temp = a[i];
```

```
                for (j = i; j >= increment; j -= increment) {
                    if (temp < a[j - increment]){
                        a[j] = a[j - increment];
                    }else{
                        break;
                    }
                }
                a[j] = temp;
            }
        }

        // 检测一下排序的结果
        for (int i2 : a) {
            System.out.println("i=" + i2);
        }
    }
}
```

运行结果：

```
i=1
i=2
i=3
i=4
i=5
```

如果你想要按照降序排列，很简单，只需要把：`if (temp < a[j - increment])` 这句话修改成：`if (temp > a[j - increment])`就可以了。

5：数组排序

事实上，数组的排序不用那么麻烦，上面只是想让大家对一些基本的排序算法有所了解而已。在 java.util.Arrays 类中有一个静态方法 sort，可以用这个类的 sort 方法来对数组进行排序。

示例如下：

```
public class Test {
    public static void main(String[] args) {
        // 需要排序的数组，目前是按照升序排列的
        int a[] = new int[5];
        a[0] = 3;
        a[1] = 4;
        a[2] = 1;
        a[3] = 5;
        a[4] = 2;

        //数组排序
        java.util.Arrays.sort(a);
```

```
        // 检测一下排序的结果
        for (int i2 : a) {
            System.out.println("i=" + i2);
        }
    }
}
```

注意：现在的 sort 方法都是升序的，要想实现降序的，还需要 Comparator 的知识，这个在后面会学到。