

选择排序（冒泡排序）

1.排序思想：

首先，找到数组中最小的那个元素，其次，将它和数组的第一个元素交换位置（如果第一个元素就是最小元素那么它就和自己交换）。再次，在剩下的元素中找到最小的元素，将它与数组的第二个元素交换位置。如此往复，直到将整个数组排序。

2.代码实现

```
public class Selection {
    ...
    public static void sort(Comparable[] a){
        int N = a.length;
        for(int i=0;i<N;i++){
            int min = i;
            for(int j=i+1;j<N;j++){
                if(less(a[j],a[min])){
                    min = j;
                }
            }
            exch(a,i,min);
        }
    }
    ...
    private static boolean less(Comparable v,Comparable w){
        return v.compareTo(w)<0;
    }

    private static void exch(Comparable[] a,int i,int j){
        Comparable t = a[i];
        a[i] = a[j];
        a[j] = t;
    }

    private static void show(Comparable[] a){
        for(int i=0;i<a.length;i++){
            System.out.print(a[i]+" ");
        }
        System.out.println();
    }

    public static boolean isSorted(Comparable[] a){

        for(int i=1;i<a.length;i++){
            if(less(a[i],a[i-1])){
                return false;
            }
        }

        return true;
    }

    public static void main(String[] args){
        Integer[] numArray = {4,878,9,655,225,76,3,26,59,46,12,36,33};
        sort(numArray);
        assert(isSorted(numArray));
        show(numArray);
    }
}
```

```
}
```

3.收获

在代码实现中，高亮的代码则是选择排序的核心算法，在以前，我已经把冒泡排序写成这样：

```
public static void mySort(Comparable[] a){
    for(int i=0;i<a.length;i++){
        for(int j=i+1;j<a.length;j++){
            if(a[i].compareTo(a[j])>0){
                Comparable t = a[i];
                a[i] = a[j];
                a[j] = t;
            }
        }
    }
}
```

我这种写法，跟算法书里不同的是，我的交换位置的代码放在了第二个循环里面，假如数组里面的对象，在排序之前是跟理想的顺序相反的，那么在我的程序里面，两个数组对象交换位置的代码则需要执行 $n^2 - n(n+1)/2$ 次，而高亮的代码，将交换位置的代码块放在内循环外面，内循环只负责标记范围内的最小数值，这样子整个程序的数组交换次数则为 n ，所以在大概率上，我的方法会耗费更多的时间，性能较低。

4.命题

对于长度为 N 的数组，选择排序需要大约 $N^2/2$ 次比较和 N 次交换。

一般认为选择排序的时间复杂度为 N^2 。