选择排序

选择排序的算法是这样的：

首先，找到数组中最小的那个数，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，那么他就和自己交换）。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。

选择排序实质上是在不断的选择剩余元素之中的最小者。

选择排序的内循环只是在比较当前元素与目前已知的最小元素（以及将当前索引加一和检查是否代码越界）。交换元素的代码写在内循环之外，每次交换都能排定一个元素，因此交换的总次数是N，所以算法的时间效率取决于比较的次数。

对于长度为N的数组，选择排序需要大约N2/2次比较和N次交换

总的来说，排序是一种很容易理解和实现的简单排序算法，它有两个很鲜明的特点：

运行时间和输入无关。为了找出最小的元素而扫描一遍数组并不能成为下一遍扫描提供什么信息。这种性质在某些情况下是缺点。

数据移动是最少的。每次交换都会改变两个数组的元素的值，因此选择排序用了N次交换—交换次数和数组的大小是线性关系。其他的任何算法都不具备这个特征（大部分的增长数量级都是线性对数或是平方级别的）

排序主要代码，按照之前的模板来写的话，如下：

1. **package** com.lagoon.sort;
3. /\*\*
4. \* @Author WinkiLee
5. \* @Date 2019/4/10 19:49
6. \* @Description
7. \*/
8. **public** **class** SelectSort {
10. **public** **static** **void** sort(Comparable[] a){
11. /\*\*
12. \* 算法编写区域
13. \*/
14. //将a[]按升序排列
15. **int** N=a.length;
16. **for** (**int** i=0;i<N;i++){
17. //将a[i]与a[i+1..N]中最小元素交换
18. //最小元素索引
19. **int** min=i;
20. **for** (**int** j=i+1;j<N;j++){
21. **if** (less(a[j],a[min])){
22. min=j;
23. }
24. }
25. exch(a,i,min);
26. }
27. }
29. /\*\*
30. \* 比较大小
31. \* @param v
32. \* @param w
33. \* @return false or true
34. \*/
35. **private** **static** **boolean** less(Comparable v,Comparable w){
36. **return** v.compareTo(w)<0;
37. }
39. /\*\*
40. \* 交换数
41. \* @param a
42. \* @param i
43. \* @param j
44. \*/
45. **private** **static**  **void** exch(Comparable[] a,**int** i,**int** j){
46. Comparable t=a[i];
47. a[i]=a[j];
48. a[j]=t;
49. }
51. /\*\*
52. \* 打印结果
53. \* @param a
54. \*/
55. **private**  **static** **void** show(Comparable[] a){
56. //在单行中打印数组
57. **for** (**int** i=0;i<a.length;i++){
58. System.out.print(a[i]+"");
59. }
60. System.out.println("");
61. }
63. /\*\*
64. \* 测试元素数组是否有序
65. \* @param a
66. \* @return true or false
67. \*/
68. **public** **static** **boolean** isSorted(Comparable[] a){
69. //测试元素数组是否有序
70. **for** (**int** i=1;i<a.length;i++){
71. **if** (less(a[i],a[i-1])){
72. **return** **false**;
73. }
74. }
75. **return** **true**;
76. }
78. /\*\*
79. \* main方法及测试
80. \* @param args
81. \*/
82. **public** **static** **void** main(String[] args) {
84. Comparable[] a ={6,4,1,3,5,9,7,8,2};
85. Comparable[] b ={'a','e','w','p','s','b','q'};
86. sort(a);
87. sort(b);
88. show(a);
89. show(b);
90. }
91. }

测试结果：

