

算法思路
文字说明
图解说明
代码实现
总结

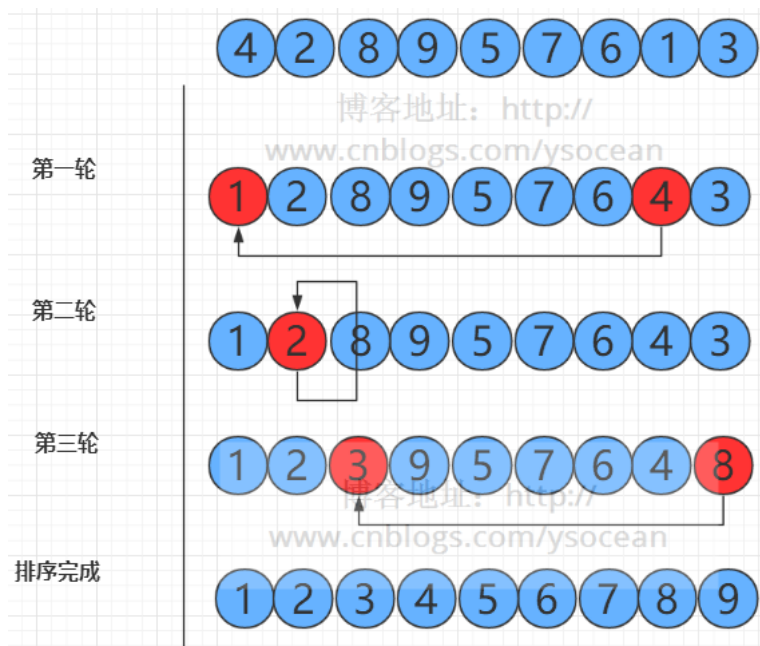
算法思路

与冒泡排序不同，冒泡是相邻两个不断对比
选择排序的思路是，减少交换次数：第一轮从第一个开始，遍历到末尾，找到最小值，与第一个元素交换 第二轮从第二个开始，遍历到末尾，找到最小值，与第二个元素交换 第三轮从第三个开始，遍历到末尾，找到最小值，与第三个元素交换 依次类推到 N - 1 轮就可以处理完毕

文字说明

第一轮
从第一个开始找到关键字最小的元素，将它与第一个元素互换
第二轮
继续从第二个开始遍历，找到最小元素，将它与第二个元素互换
最终
经过N-1轮之后，即可得到结果

图解说明



代码实现

```
1 package day11.排序;
2
3 /**
4  * 选择排序
5  */
6 public class ChooseSort {
7
8     public static void main(String[] args) {
9         int[] array = {4, 2, 8, 9, 5, 7, 6, 1, 3};
10
11         //未排序数组顺序为
12         System.out.println("未排序数组顺序为: ");
13         display(array);
14         System.out.println("-----");
15         chooseSort(array);
16         System.out.println("经过选择排序后的数组顺序为: ");
17         display(array);
18     }
19
20     private static void chooseSort(int[] array) {
21
22         for (int i = 0; i < array.length - 1; i++) {
23             int min_index = i;
24             for (int j = i + 1; j < array.length; j++) {
25                 if (array[min_index] > array[j]) {
26                     min_index = j;
27                 }
28             }
29             if (i != min_index) {
```

```

30         int tmp = array[min_index];
31         array[min_index] = array[i];
32         array[i] = tmp;
33     }
34
35     //第 i轮排序的结果为
36     System.out.print("第"+(i+1)+"轮排序后的结果为:");
37     display(array);
38 }
39
40 }
41
42 public static void display(int[] array) {
43     for (int i = 0; i < array.length; i++) {
44         System.out.print(array[i] + " ");
45     }
46     System.out.println();
47 }
48 }
49
50 /*输出
51 未排序数组顺序为:
52 4 2 8 9 5 7 6 1 3
53 -----
54 第1轮排序后的结果为:1 2 8 9 5 7 6 4 3
55 第2轮排序后的结果为:1 2 8 9 5 7 6 4 3
56 第3轮排序后的结果为:1 2 3 9 5 7 6 4 8
57 第4轮排序后的结果为:1 2 3 4 5 7 6 9 8
58 第5轮排序后的结果为:1 2 3 4 5 7 6 9 8
59 第6轮排序后的结果为:1 2 3 4 5 6 7 9 8
60 第7轮排序后的结果为:1 2 3 4 5 6 7 9 8
61 第8轮排序后的结果为:1 2 3 4 5 6 7 8 9
62 经过选择排序后的数组顺序为:
63 1 2 3 4 5 6 7 8 9
64 */

```

总结

选择排序和冒泡排序执行了相同次数的比较： $N * (N-1) / 2$ ，但是至多只进行了 N 次交换，但是由于选择排序交换的次数少，所以选择排

但是如果序列中元素的稳定性比较好，那么冒泡排序改进后的排序会比 选择排序快