

算法思路
文字说明
图解说明
代码实现
总结

## 算法思路

### 文字说明

想象一下有9张扑克牌按照这样的顺序 {4, 2, 8, 9, 5, 7, 6, 1, 3} 在牌堆里面

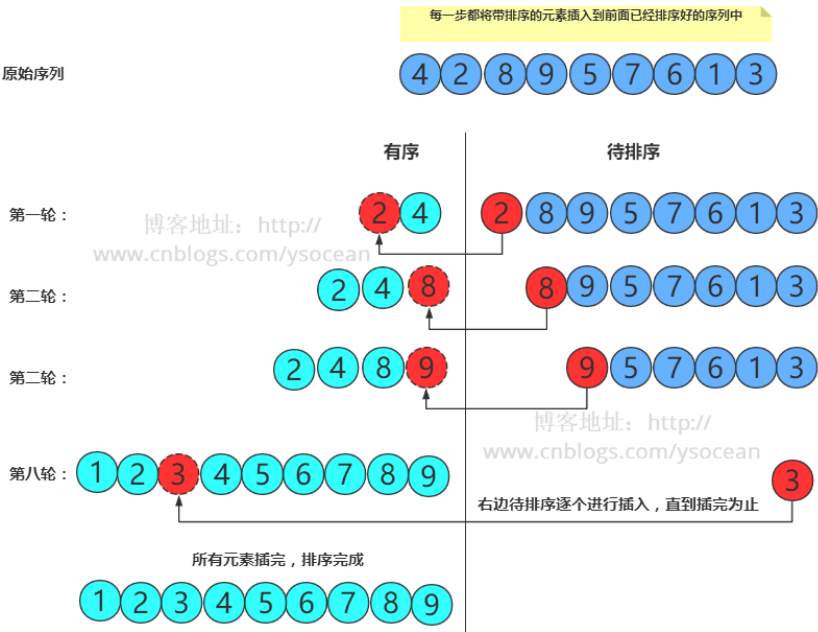
第一次拿一张牌，牌堆少一张

第二次拿一张，牌堆又少一张，将拿到的新牌与手中的牌比较，如果比较比第一张大，就将新牌放到第一个位置

第三次拿一张，牌堆又少一张，将拿到的新牌与手中的牌比较，看是放到末尾，还是放到第一个，或者是插入到两个牌中间

反复如此，直到牌堆没牌。拿到手里的牌就是排序好的结果

### 图解说明



## 代码实现

```
1 package day11.排序;
2
3 /**
4  * 插入排序
5  */
6 public class InsertSort {
7
```

```

8     public static void main(String[] args) {
9
10         int[] array = {4, 2, 8, 9, 5, 7, 6, 1, 3};
11
12         //未排序数组顺序为
13         System.out.println("未排序数组顺序为: ");
14         display(array);
15         System.out.println("-----");
16         insertSort(array);
17         System.out.println("经过插入排序后的数组顺序为: ");
18         display(array);
19
20     }
21
22     public static void insertSort(int[] array) {
23         int j;
24         //从下标为1的元素开始选择合适的位置插入，因为下标为0的只有一个元素，默认是有序的
25         for (int i = 1; i < array.length; i++) {
26             int tmp = array[i]; //记录要插入的数据
27             j = i;
28             while (j > 0 && tmp < array[j - 1]) { //从已经排序的序列最右边的开始比较，找到比其小的数
29                 array[j] = array[j - 1]; //向后挪动
30                 j--;
31             }
32             array[j] = tmp; //存在比其小的数，插入
33         }
34     }
35
36     public static void display(int[] array) {
37         for (int i = 0; i < array.length; i++) {
38             System.out.print(array[i] + " ");
39         }
40         System.out.println();
41     }
42
43 }
44

```

## 总结

在第一轮排序中，它最多比较一次，第二轮最多比较两次，一次类推，第N轮，最多比较N-1次。因此有  $1+2+3+\dots+N-1 = N*(N-1)/2$ ，用别