

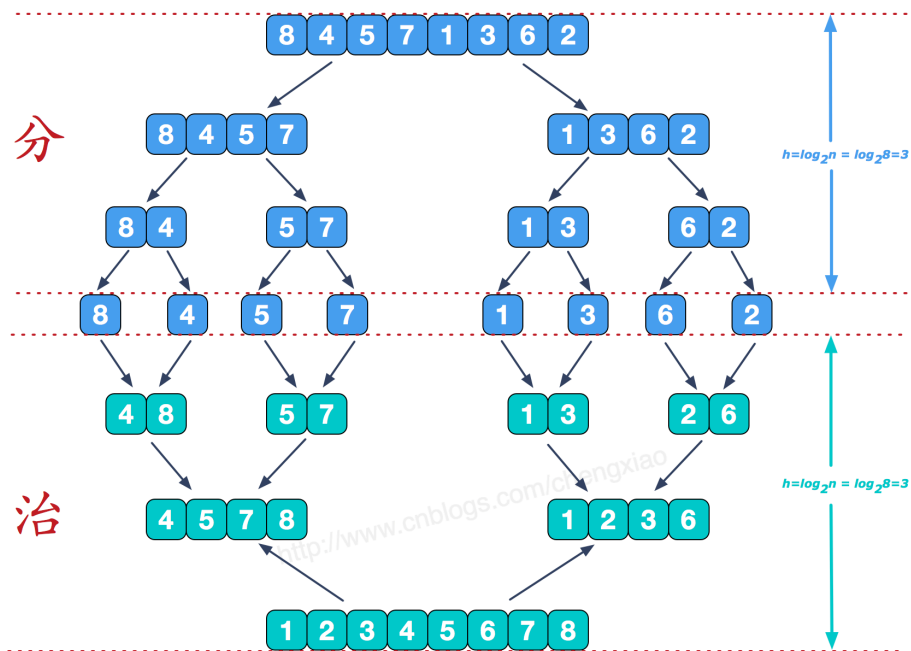
定义

归并排序是建立在归并操作上的一种有效的排序算法，该算法是采用分治法（Divide and Conquer）的一个非常典型的应用。

将已有序的子序列合并，得到完全有序的序列；即先使每个子序列有序，再使子序列段间有序。若将两个有序表合并成一个有序表，称

思想

归并排序（MERGE-SORT）是利用归并的思想实现的排序方法，该算法采用经典的分治（divide-and-conquer）策略（分治法将问题分解，而治(conquer)的阶段则将分的阶段得到的各答案"修补"在一起，即分而治之）



代码实现

```
1 package day11.排序;
2
3 import java.util.Arrays;
4
5 /**
6  * 归并排序
7  */
8 public class MergeSort {
9
10     public static void main(String[] args) {
11         int[] array = {8, 9, 1, 7, 2, 3, 5, 4, 6, 0};
12
13         //未排序数组顺序为
14         System.out.println("未排序数组顺序为: ");
```

```

15     display(array);
16     System.out.println("-----");
17     array = mergeSort(array);
18     System.out.println("经过排序后的数组顺序为: ");
19     display(array);
20 }
21
22 public static void display(int[] array) {
23     for (int i = 0; i < array.length; i++) {
24         System.out.print(array[i] + " ");
25     }
26     System.out.println();
27 }
28
29 /**
30  * 归并排序
31  *
32  * @param array
33  * @return
34  */
35 public static int[] mergeSort(int[] array) {
36     if (array.length < 2) return array;
37     int mid = array.length / 2;
38     int[] left = Arrays.copyOfRange(array, 0, mid);
39     int[] right = Arrays.copyOfRange(array, mid, array.length);
40     return merge(mergeSort(left), mergeSort(right));
41 }
42
43 /**
44  * 归并排序——将两段排序好的数组结合成一个排序数组
45  *
46  * @param left
47  * @param right
48  * @return
49  */
50 public static int[] merge(int[] left, int[] right) {
51     int[] result = new int[left.length + right.length];
52     for (int index = 0, i = 0, j = 0; index < result.length; index++) {
53         if (i >= left.length)
54             result[index] = right[j++];
55         else if (j >= right.length)
56             result[index] = left[i++];
57         else if (left[i] > right[j])
58             result[index] = right[j++];
59         else
60             result[index] = left[i++];
61     }
62     return result;
63 }
64 }

```

总结

归并排序是稳定排序，它也是一种十分高效的排序，能利用完全二叉树特性的排序一般性能都不会太差。
每次合并操作的平均时间复杂度为 $O(n)$ ，而完全二叉树的深度为 $\lceil \log_2 n \rceil$ 。总的平均时间复杂度为 $O(n \log n)$