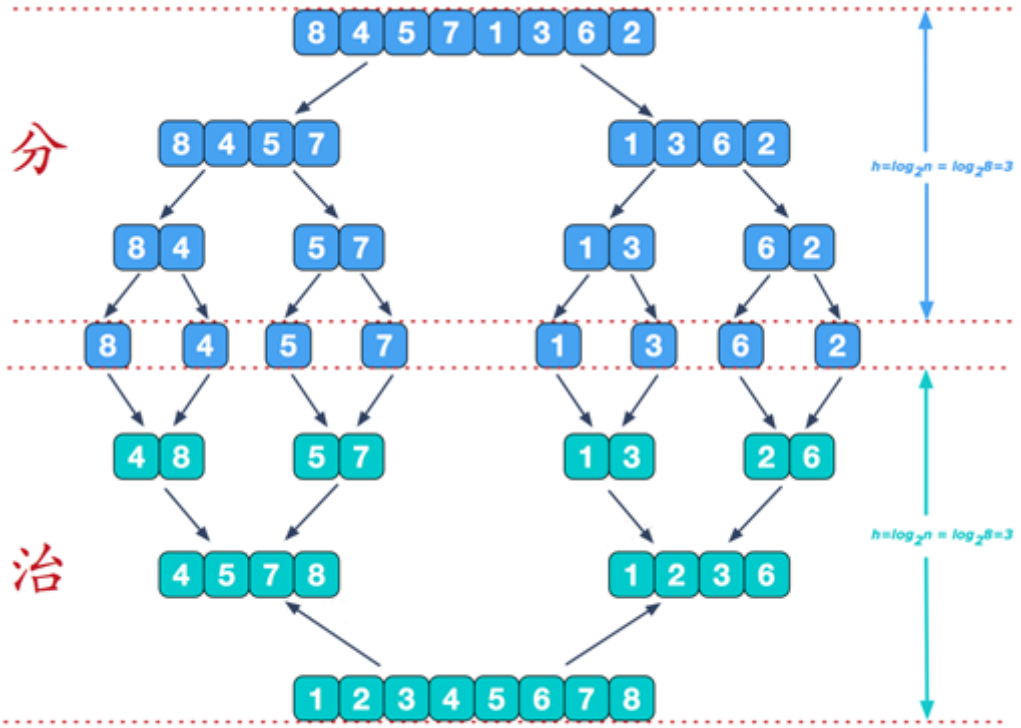
归并排序

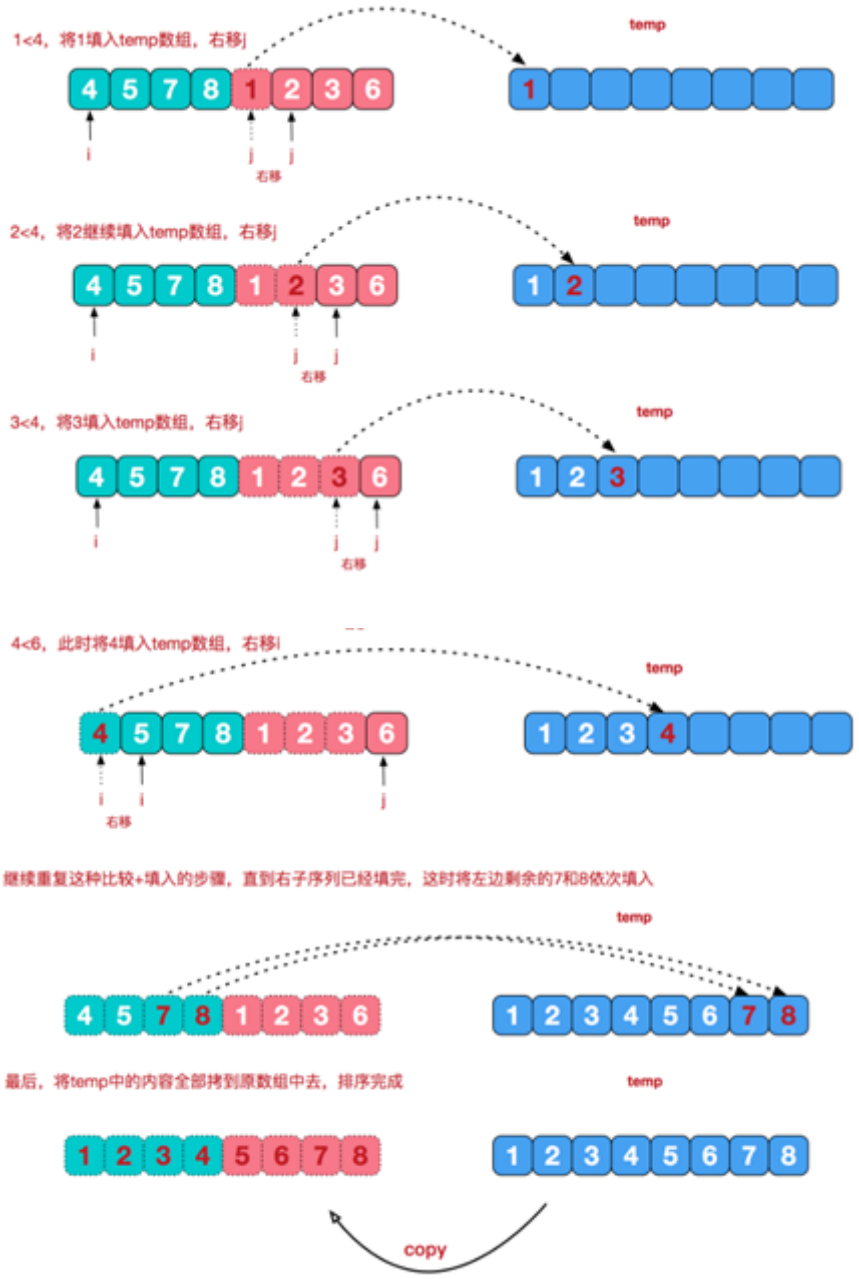
1.排序原理

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(divide)成一些小的问题然后递归求解，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)。

2.图解



治的思路：



治的过程，就是将2个有序数组合并为一个有序数组

3.代码实现

```
package Sort排序;

import com.sun.org.apache.xpath.internal.WhitespaceStrippingElementMatcher;

import java.util.Arrays;
import java.util.Date;
import java.util.Random;

/**
 * @author jndeng
 * @create 2019-12-18 13:48
 */
```

```
public class MergeSort {
    public static void main(String[] args) {

        int[] array = new int[80000];
        for (int i = 0; i < array.length; i++) {
            array[i] = new Random().nextInt()*80000;
        }
        Date date = new Date();
        long time = date.getTime();
        int[] temp = new int[array.length];

        mergeSort(array, 0, array.length - 1, temp);


        Date date1 = new Date();
        long time1 = date1.getTime();
        System.out.println("归并排序:");
        System.out.println(time1-time);
    }

    public static void mergeSort(int[] arr, int left, int right, int[] temp) {
        if (left < right) {
            int mid = (right + left) / 2;
            mergeSort(arr, left, mid, temp);
            mergeSort(arr, mid + 1, right, temp);
            merge(arr, left, right, mid, temp);
        }
    }

    public static void merge(int[] arr, int left, int right, int mid, int[] temp) {
        int i = left;
        int j = mid + 1;
        int t = 0;
        while (i <= mid && j <= right) {
            if (arr[i] < arr[j]) {
                temp[t] = arr[i];
                t++;
                i++;
            } else {
                temp[t] = arr[j];
                t++;
                j++;
            }
        }
        while (j <= right) {
            temp[t] = arr[j];
            j++;
```

```
            t++;
        }
        while (i <= mid) {
            temp[t] = arr[i];
            i++;
            t++;
        }
        t = 0;
        for (int k = left; k <= right; k++) {
            arr[k] = temp[t];
            t++;
        }
    }
    /**
     * 算法书里的原地归并方法,原理一样
     * @param arr
     * @param left
     * @param right
     * @param mid
     * @param temp
     */
    public static void merge1(int[] arr, int left, int right, int mid, int[] temp) {
        int i = left;
        int j = mid + 1;
        for (int k = left; k <= right; k++) {
            temp[k] = arr[k];
        }
        for (int k = left; k <= right; k++) {
            if (i > mid) {
                arr[k] = temp[j++];
            } else if (j > right) {
                arr[k] = temp[i++];
            } else if (temp[i] < temp[j]) {
                arr[k] = temp[i++];
            }else {
                arr[k] = temp[j++];
            }
        }
    }
}
```