

你好，世界

我的第一篇文章sslfalfjslfjwieifslfsjl烦死了烦就是雷锋精神疗法时间方式累计发生了减法所肩负历史交锋历史

手机发了手机发了时间烦死了方式解决方式累计发生了房间睡懒觉烦死了烦诶飞机上了飞机死了烦死我就放了是否就是雷锋精神了方式jlwieifjjfls三级分类收集方式

[下载 PDF](#)

```

package com.test;

public class Sort {
    // 对R[low..high]快速排序
    public static void QuickSort(Comparable[] data, int low, int high) {
        int pivotpos; // 划分后的基准记录的位置

        if (low < high) { // 仅当区间长度大于1时才须排序
            pivotpos = Partition(data, low, high); // 对Comparable[] data做划分
            QuickSort(data, low, pivotpos - 1); // 对左区间递归排序
            QuickSort(data, pivotpos + 1, high); // 对右区间递归排序
        }
    } // QuickSort

    private static int Partition(Comparable[] data, int i, int j) {
        // 从数组两端交替地向中间扫描
        Comparable pivotKey = data[i];
        // 进行扫描的指针i,j;i从左边开始,j从右边开始
        while (i < j) {
            while (i < j && data[j].compareTo(pivotKey) > 0) {
                j--;
            } // end while
            if (i < j) {
                // 比枢纽元素小的移动到左边
                data[i] = data[j];
                i++;
            } // end if
            while (i < j && data[i].compareTo(pivotKey) < 0) {
                i++;
            } // end while
            if (i < j) {
                // 比枢纽元素大的移动到右边
                data[j] = data[i];
                j--;
            } // end if
        } // end while
        // 枢纽元素移动到正确位置
        data[i] = pivotKey;
        return i;
    }

    public static void main(String[] args) {
        Comparable[] c = {4, 9, 23, 1, 45, 27, 5, 2 };
        QuickSort(c, 0, c.length - 1);
        for (Comparable data : c) {
            System.out.println(data);
        }
    }
}

```