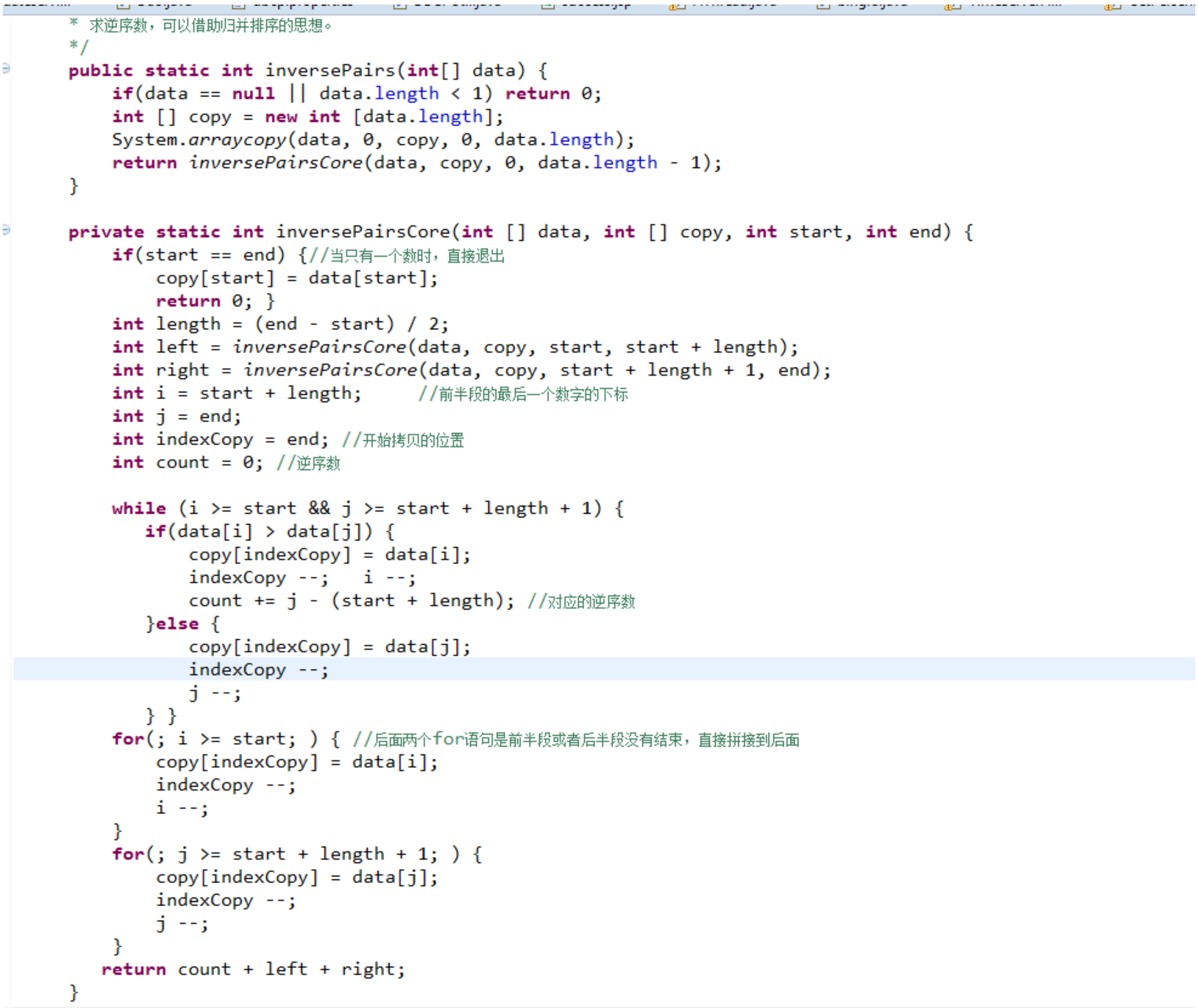

```
    * 求逆序数，可以借助归并排序的思想。
    */
    public static int inversePairs(int[] data) {
        if(data == null || data.length < 1) return 0;
        int [] copy = new int [data.length];
        System.arraycopy(data, 0, copy, 0, data.length);
        return inversePairsCore(data, copy, 0, data.length - 1);
    }

    private static int inversePairsCore(int [] data, int [] copy, int start, int end) {
        if(start == end) {//当只有一个数时，直接退出
            copy[start] = data[start];
            return 0; }
        int length = (end - start) / 2;
        int left = inversePairsCore(data, copy, start, start + length);
        int right = inversePairsCore(data, copy, start + length + 1, end);
        int i = start + length;      //前半段的最后一个数字的下标
        int j = end;
        int indexCopy = end; //开始拷贝的位置
        int count = 0; //逆序数

        while (i >= start && j >= start + length + 1) {
           if(data[i] > data[j]) {
               copy[indexCopy] = data[i];
               indexCopy --;   i --;
               count += j - (start + length); //对应的逆序数
           }else {
               copy[indexCopy] = data[j];
               indexCopy --;
               j --;
           } }
        for(; i >= start; ) { //后面两个for语句是前半段或者后半段没有结束，直接拼接到后面
            copy[indexCopy] = data[i];
            indexCopy --;
            i --;
        }
        for(; j >= start + length + 1; ) {
            copy[indexCopy] = data[j];
            indexCopy --;
            j --;
        }
       return count + left + right;
    }
```