```java
/* 给定数组a,如果 a[i] > a[j] (i < j)则称为一个反序，例如 {1,5,3,2,6}共有{5,3},{5,2},{3,2}三个反序对
找出任意给定数组的反序对个数 */

package array;

class CountInversePairsInArray {

    private static int inversePairsNumber = 0;

    public static Integer countInversePairsInArray(int[] nums) {

        if (nums == null || nums.length <= 1) {
            return 0;
        }

        mergeSortHelper(nums, 0, nums.length - 1);

        return inversePairsNumber;
    }

    private static void mergeSortHelper(int[] nums, int startIndex, int endIndex) {

        if (startIndex < endIndex) {

            int middleIndex = startIndex + (endIndex - startIndex) / 2;
            mergeSortHelper(nums, startIndex, middleIndex);
            mergeSortHelper(nums, middleIndex + 1, endIndex);
            mergeProcess(nums, startIndex, middleIndex, endIndex);
        }
    }

    private static void mergeProcess(int[] nums, int startIndex,
                                     int middleIndex, int endIndex) {

        int leftIndex = startIndex, rightIndex = middleIndex + 1;
        int newNumsIndex = 0;
        int[] newNums = new int[endIndex - startIndex + 1];

        while (leftIndex <= middleIndex && rightIndex <= endIndex) {

            if (nums[leftIndex] <= nums[rightIndex]) {
                newNums[newNumsIndex] = nums[leftIndex];
                leftIndex++;
            } else {
                newNums[newNumsIndex] = nums[rightIndex];
                rightIndex++;
                inversePairsNumber += (middleIndex - leftIndex + 1);
            }
            newNumsIndex++;
        }


        if (leftIndex <= middleIndex) {
            System.arraycopy(nums, leftIndex,
                    nums, startIndex + newNumsIndex,
                    middleIndex - leftIndex + 1);
        }
        System.arraycopy(newNums, 0, nums, startIndex, newNumsIndex);
    }

    public static void main(String[] args) {

        int[] nums = {5,5,3,2,1};

        System.out.println(countInversePairsInArray(nums));

        System.out.println("haha");
    }
}
```