```java
package sortAlgorithm;

class MergeSort {

    public static void mergeSort(int[] nums) {

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

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

    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);

            mergeTwoParts(nums, startIndex, endIndex, middleIndex);
        }
    }


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

        int[] sortedNums = new int[endIndex - startIndex + 1];
        int firstPointer = startIndex, secondPointer = middleIndex + 1;
        int sortedNumsPointer = 0;

        while (firstPointer <= middleIndex && secondPointer <= endIndex) {
            if (nums[firstPointer] < nums[secondPointer]) {
                sortedNums[sortedNumsPointer] = nums[firstPointer];
                firstPointer++;
            } else {
                sortedNums[sortedNumsPointer] = nums[secondPointer];
                secondPointer++;
            }
            sortedNumsPointer++;
        }
        if (firstPointer <= middleIndex) {
            System.arraycopy(nums, firstPointer, sortedNums, sortedNumsPointer,
                    sortedNums.length - sortedNumsPointer);
        } else if (secondPointer <= endIndex) {
            System.arraycopy(nums, secondPointer, sortedNums, sortedNumsPointer,
                    sortedNums.length - sortedNumsPointer);
        }

        System.arraycopy(sortedNums, 0, nums, startIndex, sortedNums.length);
    }

    public static void main(String[] args) {

        int[] nums = {2, 7, 4, 2, 3, 9, -1, 9, 18};

        mergeSort(nums);

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