```java
package array;

/* 数组 nums[0, mid - 1] 和 nums[mid, n-1]是各自有序(升序)的，现在要使得 nums[0, n-1]整体有序 */

import java.util.Arrays;

class MergeTwoSortedPartsInArray {

    public static void mergeTwoSortedPartsInArray(int[] nums, int mid) {

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

        int leftIndex = 0, rightIndex = mid;
        int leftEndIndex = mid;

        while (leftIndex < leftEndIndex && rightIndex < nums.length) {

            if (nums[rightIndex] < nums[leftIndex]) {

                reverseArray(nums, leftIndex, rightIndex - 1);
                reverseArray(nums, leftIndex, rightIndex);
                rightIndex++;
                leftIndex++;
                leftEndIndex++;

            } else {
                leftIndex++;
            }
        }
    }

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

        while (startIndex < endIndex) {

            int temp = nums[startIndex];
            nums[startIndex] = nums[endIndex];
            nums[endIndex] = temp;

            startIndex++;
            endIndex--;
        }

    }


    public static void main(String[] args) {

        int[] nums = {1, 5, 6, 7, 9, 2, 4, 8, 10, 13, 14};
        int mid = 5;

        mergeTwoSortedPartsInArray(nums, mid);

        System.out.println(Arrays.toString(nums));
    }
}
```