Merge sort exercise

code

class Tester {

public static void mergeSort(int[] elements, int size) {

if (size < 2) {

return;

}

int mid = size / 2;

int[] left = new int[mid];

int[] right = new int[size - mid];

// Fill left and right subarrays

for (int i = 0; i < mid; i++) {

left[i] = elements[i];

}

for (int i = mid; i < size; i++) {

right[i - mid] = elements[i];

}

// Recursively sort left and right subarrays

mergeSort(left, mid);

mergeSort(right, size - mid);

// Merge sorted left and right subarrays into elements

merge(elements, left, right, mid, size - mid);

}

public static void merge(int[] elements, int[] left, int[] right, int leftMerge, int rightMerge) {

int i = 0, j = 0, k = 0;

// Merge left and right arrays into elements

while (i < leftMerge && j < rightMerge) {

if (left[i] <= right[j]) {

elements[k++] = left[i++];

} else {

elements[k++] = right[j++];

}

}

// Copy remaining elements of left array, if any

while (i < leftMerge) {

elements[k++] = left[i++];

}

// Copy remaining elements of right array, if any

while (j < rightMerge) {

elements[k++] = right[j++];

}

}

public static void displayArray(int[] elements) {

for (int element : elements) {

System.out.print(element + " ");

}

System.out.println();

}

public static void main(String[] args) {

int[] elements = { 95, 56, 20, 98, 34, 77, 80 };

System.out.println("Given Array:");

displayArray(elements);

mergeSort(elements, elements.length);

System.out.println("Sorted Array:");

displayArray(elements);

}

}

class Tester {

public static void mergeSort(int[] elements, int size) {

if (size < 2) {

return;

}

int mid = size / 2;

int[] left = new int[mid];

int[] right = new int[size - mid];

// Fill left and right subarrays

for (int i = 0; i < mid; i++) {

left[i] = elements[i];

}

for (int i = mid; i < size; i++) {

right[i - mid] = elements[i];

}

// Recursively sort left and right subarrays

mergeSort(left, mid);

mergeSort(right, size - mid);

// Merge sorted left and right subarrays into elements

merge(elements, left, right, mid, size - mid);

}

public static void merge(int[] elements, int[] left, int[] right, int leftMerge, int rightMerge) {

int i = 0, j = 0, k = 0;

// Merge left and right arrays into elements

while (i < leftMerge && j < rightMerge) {

if (left[i] <= right[j]) {

elements[k++] = left[i++];

} else {

elements[k++] = right[j++];

}

}

// Copy remaining elements of left array, if any

while (i < leftMerge) {

elements[k++] = left[i++];

}

// Copy remaining elements of right array, if any

while (j < rightMerge) {

elements[k++] = right[j++];

}

}

public static void displayArray(int[] elements) {

for (int element : elements) {

System.out.print(element + " ");

}

System.out.println();

}

public static void main(String[] args) {

int[] elements = { 95, 56, 20, 98, 34, 77, 80 };

System.out.println("Given Array:");

displayArray(elements);

mergeSort(elements, elements.length);

System.out.println("Sorted Array:");

displayArray(elements);

}

}class Tester {

public static void mergeSort(int[] elements, int size) {

if (size < 2) {

return;

}

int mid = size / 2;

int[] left = new int[mid];

int[] right = new int[size - mid];

// Fill left and right subarrays

for (int i = 0; i < mid; i++) {

left[i] = elements[i];

}

for (int i = mid; i < size; i++) {

right[i - mid] = elements[i];

}

// Recursively sort left and right subarrays

mergeSort(left, mid);

mergeSort(right, size - mid);

// Merge sorted left and right subarrays into elements

merge(elements, left, right, mid, size - mid);

}

public static void merge(int[] elements, int[] left, int[] right, int leftMerge, int rightMerge) {

int i = 0, j = 0, k = 0;

// Merge left and right arrays into elements

while (i < leftMerge && j < rightMerge) {

if (left[i] <= right[j]) {

elements[k++] = left[i++];

} else {

elements[k++] = right[j++];

}

}

// Copy remaining elements of left array, if any

while (i < leftMerge) {

elements[k++] = left[i++];

}

// Copy remaining elements of right array, if any

while (j < rightMerge) {

elements[k++] = right[j++];

}

}

public static void displayArray(int[] elements) {

for (int element : elements) {

System.out.print(element + " ");

}

System.out.println();

}

public static void main(String[] args) {

int[] elements = { 95, 56, 20, 98, 34, 77, 80 };

System.out.println("Given Array:");

displayArray(elements);

mergeSort(elements, elements.length);

System.out.println("Sorted Array:");

displayArray(elements);

}

}

