Sorts – III

1. sort an array with many duplicated Values : Given an array with many duplicated elements, sort it in linear time where the order of equal elements doesn't matter.

For example,

Input: { 4, 2, 40, 10, 10, 1, 4, 2, 1, 10, 40 }

Output: { 1, 1, 2, 2, 4, 4, 10, 10, 10, 40, 40 }

2. Find a sorted triplet in the given array

Given an array A[] of integers, find a sorted triplet such that A[i] < A[j] < A[k] and 0 <= i < j < k < n where n is the size of the array.

For example,

Input: A[] = { 5, 4, 3, 7, 6, 1, 9 }

Output: Any one of below triplets:

(5, 7, 9)

(4, 7, 9)

(3, 7, 9)

(5, 6, 9)

3. Find smallest window in array sorting which will make entire array sorted

Given an array of integers, find the smallest window in array sorting which will make the entire array sorted in increasing order.

For example,

Input: { 1, 2, 3, 7, 5, 6, 4, 8 }

Output: Sort the array from index 3 to 6

Input: { 1, 3, 2, 7, 5, 6, 4, 8 }

Output: Sort the array from index 1 to 6

4. In-place merge two sorted arrays: Given two sorted arrays X[] and Y[] of size m and n each, merge elements of X[] with elements of array Y[] by maintaining the sorted order. i.e. fill X[] with first m smallest elements and fill Y[] with remaining elements.

The conversion should be done in-place and without using any other data structure.

For example,

Input:

X[] = { 1, 4, 7, 8, 10 }

Y[] = { 2, 3, 9 }

Output:

X[] = { 1, 2, 3, 4, 7 }

Y[] = { 8, 9, 10 }

5. Merge two arrays by satisfying given constraints

Given two sorted arrays X[] and Y[] of size m and n each where m >= n and X[] has exactly n vacant cells, merge elements of Y[] in their correct position in array X[] i.e. merge (X, Y) by keeping the sorted order.

For example,

Input:

X[] = { 0, 2, 0, 3, 0, 5, 6, 0, 0}

Y[] = { 1, 8, 9, 10, 15 }

The vacant cells in X[] is represented by 0

Output:

X[] = { 1, 2, 3, 5, 6, 8, 9, 10, 15 }

6. Sort an array based on order defined by another array

Given two arrays, reorder elements of first array by order of elements defined by the second array. Elements which are not present in the second array but present in first array should be appended in the end sorted. Second array can contain some extra elements which are not part of first array.

For example,

Input:

arr1 = [5, 8, 9, 3, 5, 7, 1, 3, 4, 9, 3, 5, 1, 8, 4]

arr2 = [3, 5, 7, 2]

output = [3 3 3 5 5 5 7 1 1 4 4 8 8 9 9]

7. Sort an array in one swap whose two elements are swapped by mistake

Given an array where all its elements are sorted except two elements which were swapped, sort the array in linear time. Assume there are no duplicates in the array.

For example,

Input: A[] = [3, 8, 6, 7, 5, 9] OR [3, 5, 6, 9, 8, 7] OR [3, 5, 7, 6, 8, 9]

Output: A[] = [3, 5, 6, 7, 8, 9]