✓ Done

Opened: Tuesday, 10 October 2023, 12:00 AM
Due: Monday, 16 October 2023, 12:00 AM

Array-based Activities

Problem1:

Given an array of size N-1 such that it only contains distinct integers in the range of 1 to N. Find the missing element.

Example 1:

Input:

N = 5

A[] = {1,2,3,5}

Output: 4

Example 2:

Input:

N = 10

A[] = {6,1,2,8,3,4,7,10,5}

Output: 9

Expected Time Complexity: O(N)

Constraints: 1 ≤ N ≤ 100, 1 ≤ A[i] ≤ 100

Problem 2:

Given an array of size N containing only 0s, 1s, and 2s; sort the array in ascending order.

Example 1:

Input:

N = 5

arr[]= {0 2 1 2 0}

Output:

0 0 1 2 2

Example 2:

Input:

N = 3

arr[] = {0 1 0}

Output:

0 0 1

Expected Time Complexity: O(N)

Constraints: 1 <= N <= 10^6, 0 <= A[i] <= 2

?

Problem 3: Given an array of integers. Find the Inversion Count in the array.

Inversion Count: For an array, inversion count indicates how far (or close) the array is from being sorted. If the array is already sorted then the inversion count is 0.

If an array is sorted in the reverse order then the inversion count is the maximum.

Formally, two elements a[i] and a[j] form an inversion if a[i] > a[j] and i < j.

Example 1:

Input: N = 5, arr[] = {2, 4, 1, 3, 5}

Output: 3

Explanation: The sequence 2, 4, 1, 3, 5 has three inversions (2, 1), (4, 1), (4, 3).

Example 2:

Input: N = 5

arr[] = {2, 3, 4, 5, 6}

Output: 0

Explanation: As the sequence is already sorted so there is no inversion count.

Example 3:

Input: N = 3, arr[] = {10, 10, 10}

Output: 0

Explanation: As all the elements of array are same, so there is no inversion count.

Expected Time Complexity: O(NLogN).

Constraints: 1 ≤ N ≤ 5*105, 1 ≤ arr[i] ≤ 1018

Problem 4: Given two arrays a[] and b[] of size n and m respectively. The task is to find the number of elements in the union between these two arrays.

Union of the two arrays can be defined as the set containing distinct elements from both the arrays. If there are repetitions, then only one occurrence of element should be printed in the union.

Example 1:

Input:

5 3

1 2 3 4 5

1 2 3

Output:

5

Explanation:

1, 2, 3, 4 and 5 are the elements which comes in the union set of both arrays. So count is 5.

Example 2:

Input:

6 2

85 25 1 32 54 6

85 2

Output:

7

Explanation:

85, 25, 1, 32, 54, 6, and 2 are the elements which comes in the union set of both arrays. So count is 7.

Constraints: 1 ≤ n, m ≤ 105, 0 ≤ a[i], b[i] < 105

Expected Time Complexity : O(n+m)

Array-based Activities

Problem1: Given an array of size N-1 such that it only contains distinct integers in the range of 1 to N. Find the missing element.

Example 1:

Input:

N = 5

A[] = {1,2,3,5}

Output: 4

Example 2:

Input:

N = 10

A[] = {6,1,2,8,3,4,7,10,5}

Output: 9

Expected Time Complexity: O(N)

Constraints: 1 ≤ N ≤ 100, 1 ≤ A[i] ≤ 100

Problem 2: Given an array of size N containing only 0s, 1s, and 2s; sort the array in ascending order.

Example 1:

Input:

N = 5

arr[]= {0 2 1 2 0}

Output:

0 0 1 2 2

Example 2:

Input:

N = 3

arr[] = {0 1 0}

Output:

0 0 1

Expected Time Complexity: O(N)

Constraints: 1 <= N <= 10^6, 0 <= A[i] <= 2

Problem 3: Given an array of integers. Find the Inversion Count in the array.

Inversion Count: For an array, inversion count indicates how far (or close) the array is from being sorted. If the array is already sorted then the inversion count is 0.

If an array is sorted in the reverse order then the inversion count is the maximum.

Formally, two elements a[i] and a[j] form an inversion if a[i] > a[j] and i < j.

Example 1:

Input: N = 5, arr[] = {2, 4, 1, 3, 5}

Output: 3

Explanation: The sequence 2, 4, 1, 3, 5 has three inversions (2, 1), (4, 1), (4, 3).

Example 2:

Input: N = 5

arr[] = {2, 3, 4, 5, 6}

Output: 0

Explanation: As the sequence is already sorted so there is no inversion count.

Example 3:

Input: N = 3, arr[] = {10, 10, 10}

Output: 0

Explanation: As all the elements of array are same, so there is no inversion count.

Expected Time Complexity: O(NLogN).

Constraints: 1 ≤ N ≤ 5*105, 1 ≤ arr[i] ≤ 1018

Problem 4:

Given two arrays a[] and b[] of size n and m respectively. The task is to find the number of elements in the union between these two arrays.

Union of the two arrays can be defined as the set containing distinct elements from both the arrays. If there are repetitions, then only one occurrence of element should be printed in the union.

Example 1:

Input:

5 3

1 2 3 4 5

1 2 3

Output:

5

Explanation:

1, 2, 3, 4 and 5 are the elements which comes in the union set of both arrays. So count is 5.

Example 2:

Input:

6 2

85 25 1 32 54 6

85 2

Output:

7

Explanation:

85, 25, 1, 32, 54, 6, and 2 are the elements which comes in the union set of both arrays. So count is 7.

Constraints: 1 ≤ n, m ≤ 105, 0 ≤ a[i], b[i] < 105

Expected Time Complexity : O(n+m)

Add submission

File submissions

Maximum file size: 2 MB, maximum number of files: 20

- Files
  - P1.c
  - P2.c
  - P3.c
  - P4.c

Accepted file types:

All file types

Save changes Cancel