Algorithms & Asymptotic Analysis:

Write an algorithm and identify best and worst case inputs for the below questions

1. **Union of two arrays**

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.

Note : Elements are not necessarily distinct.

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.

2. **Minimize number of Students to be removed**

N Students of different heights are attending an assembly. The heights of the students are represented by an array **H[].** The problem is that if a student has less or equal height than the student standing in front of him, then he/she cannot see the assembly. Find the minimum number of students to be removed such that maximum possible number of students can see

the assembly.

Example 1:

Input:
N = 6
H[] = {9, 1, 2, 3, 1, 5}
Output:
2
Explanation:
We can remove the students at 0 and 4th index.
which will leave the students with heights
1,2,3, and 5.

3. **Missing number in array**

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

Greedy Algorithms:

1. Minimum product subset of an array

Given array a, we have to find the minimum product possible with the subset of elements present in the array. The minimum product can be a single element also.

Examples:
Input : *a[] = { -1, -1, -2, 4, 3 }*
Output : *-24*
Explanation : *Minimum product will be (-2 * -1 * -1 * 4 * 3) = -24*

2. **Minimum sum by choosing minimum of pairs from array**

Given an array A[] o n-elements. We need to select two adjacent elements and delete the larger of them and store smaller of them to another array say B[]. We need to perform this operation till array A[] contains only single element. Finally, we have to construct the array B[] in such a way that total sum of its element is minimum. Print the total sum of array B[]

Examples:
Input : A[] = {3, 4}

Output : 3

Input : A[] = {2, 4, 1, 3}

Output : 3

Divide and Conquer:

1. Find the element that appears once in sorted array

Given a sorted array arr[] of size N. Find the element that appears only once in the array. All other elements appear exactly twice.

Example 1:

Input:
N = 11
arr[] = {1, 1, 2, 2, 3, 3, 4, 50, 50, 65, 65}
Output: 4
Explanation: 4 is the only element that appears exactly once.

2. Sum of Middle Elements of two sorted arrays

Given 2 sorted arrays **Ar1** and **Ar2** of size **N** each. Merge the given arrays and find the sum of the two middle elements of the merged array.

Example 1:

Input:
N = 5
Ar1[] = {1, 2, 4, 6, 10}
Ar2[] = {4, 5, 6, 9, 12}
Output: 11
Explanation: The merged array looks like {1,2,4,4,**5,6,**6,9,10,12}. Sum of middle elements is 11 (5 + 6).