Microsoft Programming - June 2024

1. Pothole Road Repair Optimization

You are tasked to determine the minimum number of drives a road roller needs to fix potholes on a straight road.

Description:

There are N potholes positioned along a straight road parallel to the y-axis. The positions of these potholes are described by two arrays X and Y:

- X[K] and Y[K] represent the coordinates of the Kth pothole ($0 \leq K < N$).

A road roller of width W will be used to patch these potholes. During each drive:

- The roller starts from a given x-coordinate and patches all potholes that fall within the range [x, x + W].
- Each drive can start from any x-coordinate at the beginning of the road.

Your task is to determine the **minimum number of drives** required to patch all potholes.

Function Signature:

```
public func solution(_ X: inout [Int], _ Y: inout [Int], _ W: Int) -> Int
```

Input:

- X: Array of integers representing the x-coordinates of the potholes.
- Y: Array of integers representing the y-coordinates of the potholes (unused for this calculation).
- W: An integer representing the width of the road roller.

Output:

- Return an integer representing the minimum number of drives required to patch all potholes.

Constraints:

- 1 ≤ N ≤ 1,000,000
- -1,000,000,000 ≤ X[K], Y[K] ≤ 1,000,000,000
- 1 ≤ W ≤ 1,000,000,000

Examples:

1. **Input:**
    - X = [2, 4, 2, 6, 7, 1]
    - Y = [0, 5, 3, 2, 1, 5]
    - W = 2
2. **Output:** 3 **Explanation:** The road roller requires three drives, starting from points 1, 4, and 7.
3. **Input:**
    - X = [4, 8, 2, 2, 1, 4]
    - Y = [1, 2, 3, 1, 2, 3]
    - W = 3
4. **Output:** 2 **Explanation:** The road roller can cover all potholes with two drives starting at points 1 and 6.
5. **Input:**
    - X = [0, 3, 6, 5]
    - Y = [0, 3, 2, 4]
    - W = 1
6. **Output:** 3 **Explanation:** The roller requires three drives starting from 0, 3, and 5.

2 .Problem Statement:

A one-day-long training session will be conducted twice during the next 10 days. There are **N employees** (numbered from `0` to `N-1`) willing to attend it. Each employee has provided a list of which of the next 10 days they are available to participate in the training.

The employees' preferences are represented as an array of strings, **E**. Each string consists of digits (`'0'-'9'`), representing the days the corresponding employee is available to attend the training.

Your task is to schedule the training sessions on two days such that the maximum number of employees can attend at least one of the two scheduled training days.

Write a function:

```
public func solution(_ E : inout [String]) -> Int
```

Input:

- **E:** An array of N strings denoting the available days for each employee. Each string consists only of digits (`'0'-'9'`), where each digit represents a day.

Output:

- The function should return the maximum number of employees who can attend at least one of the two scheduled training days.

Examples:

Input:
E = ["039", "4", "14", "32", "", "34", "7"]

Output:
5

Explanation:

Training on days 3 and 4 allows employees 0, 1, 2, 3, and 5 to attend the training.

Input:
E = ["081234567", "180234567", "0", "189234567", "891234567", "98", "9"]

Output:
7

Explanation:

Training on days 0 and 9 allows all employees to attend the training.

Input:
E = ["5421", "245", "1452", "9345", "53", "354"]

Output:
6

Explanation:

Training on day 5 allows all employees to attend the training.

Assumptions:

1. **N** is an integer within the range [1..100].
2. Each element of **E** consists only of digits ('0'-'9').
3. Each string in **E** has a length within the range [0..10].
4. Characters in each string in **E** are distinct.

Constraints:

- Focus on correctness. The performance of your solution will not be the focus of the assessment.

3. Write a function:

public func solution(_ A : inout [Int]) -> Int

that, given an array A of N integers, returns the smallest positive integer (greater than 0) that does not occur in A.

Requirements:

- The function should efficiently find the smallest missing positive integer.
- The algorithm should satisfy the following constraints:
    - N (the size of array A) is an integer within the range [1,100,000].
    - Each element of array A is an integer within the range [−1,000,000,1,000,000].

Examples:

1. Given A = [1, 3, 6, 4, 1, 2], the function should return 5.
2. Given A = [1, 2, 3], the function should return 4.
3. Given A = [-1, -3], the function should return 1.

Additional Notes:

- The function should handle both positive and negative integers, as well as zeros.
- The solution must be optimized to run efficiently for large input sizes.

Write an efficient algorithm to solve this problem.