Introduction

This pattern describes an interesting approach to deal with problems involving arrays containing numbers in a given range. For example, take the following problem:

> You are given an unsorted array containing numbers taken from the range 1 to 'n'. The array can have duplicates, which means that some numbers will be missing. Find all the missing numbers.

To efficiently solve this problem, we can use the fact that the input array contains numbers in the range of 1 to 'n'. For example, to efficiently sort the array, we can try placing each number in its correct place, i.e., placing '1' at index '0', placing '2' at index '1', and so on. Once we are done with the sorting, we can iterate the array to find all indices that are missing the correct numbers. These will be our required numbers.

Let's jump on to our first problem to understand the **Cyclic Sort** pattern in detail.

Back
Solution Review: Problem Challenge 3

Next
Cyclic Sort (easy)

Completed

Report an Issue Ask a Question