

DAILY PROGRAMMING CHALLENGE



Sort an Array of 0s, 1s, and 2s

You are given an array `arr` consisting only of 0s, 1s, and 2s. The task is to sort the array in increasing order in linear time (i.e., $O(n)$) without using any extra space. This means you need to rearrange the array in-place.

Input:

An integer array `arr` of size `n` where each element is either 0, 1, or 2.

Example : `arr = [0, 1, 2, 1, 0, 2, 1, 0]`

Output:

The sorted array, arranged in increasing order of 0s, 1s, and 2s.

Example: `[0, 0, 0, 1, 1, 1, 2, 2]`

Constraints:

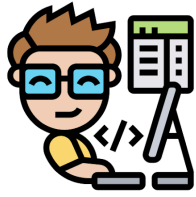
- The input array may have up to 10^5 elements.
- The values in the array are limited to 0, 1, and 2.
- The sorting algorithm must operate in linear time (i.e., $O(n)$) and in-place (i.e., without using extra space beyond a few variables).

Approach:

You can solve this problem using the Dutch National Flag Algorithm developed by Edsger Dijkstra.

Test Cases:

1. Test Case 1
Input: `[0, 1, 2, 1, 0, 2, 1, 0]`
Output: `[0, 0, 0, 1, 1, 1, 2, 2]`
2. Test Case 2:
Input: `[2, 2, 2, 2]`
Output: `[2, 2, 2, 2]`
3. Test Case 3:
Input: `[0, 0, 0, 0]`
Output: `[0, 0, 0, 0]`



DAILY PROGRAMMING

CHALLENGE



-
4. Test Case 4:
Input: [1, 1, 1, 1]
Output: [1, 1, 1, 1]
 5. Test Case 5:
Input: [2, 0, 1]
Output: [0, 1, 2]
 6. Test Case 6:
Input: []
Output: []

Edge Cases:

1. Empty array.
2. Array with all elements the same (all 0s, all 1s, or all 2s).
3. Array already sorted.