Two Swaps

solved by	Senan
Platform	GeeksForGeeks
difficulty	Easy
tags	Array Sorting
language	C++
solved on	@16/10/2024
link	https://www.geeksforgeeks.org/problems/two-swaps--155623/1
Completion	✓

Intuition

The problem checks whether the given array can be sorted by performing at most one swap of any two elements. This can happen if:

1. The array is already sorted (i.e., no swaps are required).
2. Exactly two elements are out of order, and swapping them can result in a sorted array.

The approach is based on detecting how many misplaced elements there are and whether swapping them can sort the array.

Approach

1. We iterate through the array to identify misplaced elements — elements not in their correct position.
2. If an element is out of place, we swap it with the element at the correct position.
3. We count the number of swaps made.
4. If the array is already sorted or can be sorted by exactly one swap (i.e., the swap count is either 0 or 2), we return `true`; otherwise, return `false`.

Complexity

Time Complexity:

- **O(n)**: We traverse the array once, performing constant time operations per element. Each swap places an element in its correct position, ensuring that each element is swapped at most once.

Space Complexity:

- **O(1)**: The algorithm uses a constant amount of extra space regardless of the input size.

Code

```
class Solution {
  public:
    bool checkSorted(vector<int> &arr) {
        int i = 0;
        int count = 0;
        while(i < arr.size()) {
            if(arr[i] != i + 1) {
                swap(arr[i], arr[arr[i] - 1]);
                count++;
            } else {
                i++;
            }
        }
        if(count == 0 || count == 2) return true;
        return false;
    }
};
```