educative(/learn)

Dutch National Flag Problem (medium)

We'll cover the following

- Problem Statement
- Try it yourself
- Solution
  - Code
  - Time complexity
  - Space complexity

Problem Statement#

Given an array containing 0s, 1s and 2s, sort the array in-place. You should treat numbers of the array as objects, hence, we can't count 0s, 1s, and 2s to recreate the array.

The flag of the Netherlands consists of three colors: red, white and blue; and since our input array also consists of three different numbers that is why it is called Dutch National Flag problem (https://en.wikipedia.org/wiki/Dutch_national_flag_problem).

Example 1:

```
Input: [1, 0, 2, 1, 0]
Output: [0 0 1 1 2]
```

Example 2:

```
Input: [2, 2, 0, 1, 2, 0]
Output: [0 0 1 2 2 2 ]
```

Try it yourself#

Try solving this question here:

Java | Python3 | JS | C++

```java
class DutchFlag {

  public static void sort(int[] arr) {
    // TODO: Write your code here
  }
}

```

Test Save Reset

Solution#

The brute force solution will be to use an in-place sorting algorithm like Heapsort (https://en.wikipedia.org/wiki/Heapsort) which will take $O(N * logN)$. Can we do better than this? Is it possible to sort the array in one iteration?

We can use a **Two Pointers** approach while iterating through the array. Let's say the two pointers are called `low` and `high` which are pointing to the first and the last element of the array respectively. So while iterating, we will move all 0s before `low` and all 2s after `high` so that in the end, all 1s will be between `low` and `high`.

Code#

Here is what our algorithm will look like:

Java Python3 C++ JS

```java
class DutchFlag {

  public static void sort(int[] arr) {
    // all elements < low are 0 and all elements > high are 2
    // all elements from >= low < i are 1
    int low = 0, high = arr.length - 1;
    for (int i = 0; i <= high;) {
      if (arr[i] == 0) {
        swap(arr, i, low);
        // increment 'i' and 'low'
        i++;
        low++;
      } else if (arr[i] == 1) {
        i++;
      } else { // the case for arr[i] == 2
        swap(arr, i, high);
        // decrement 'high' only, after the swap the number at index 'i' 
        high--;
      }
    }
  }
  private static void swap(int[] arr, int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
  }

```

Run Save Reset

Time complexity#

The time complexity of the above algorithm will be $O(N)$ as we are iterating the input array only once.

Space complexity#

The algorithm runs in constant space $O(1)$.

← Back

Subarrays with Product Less than a Ta...

Next →

Problem Challenge 1

✓ Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/dutch-national-flag-problem-medium__pattern-two-pointers__grokking-the-coding-interview-patterns-for-coding-questions?open=true&ctag=grokking-the-coding-interview-patterns-for-coding-questions__design-gurus&aid=5668639101419520&cid=5671464854355968&pid=5784530449334272)