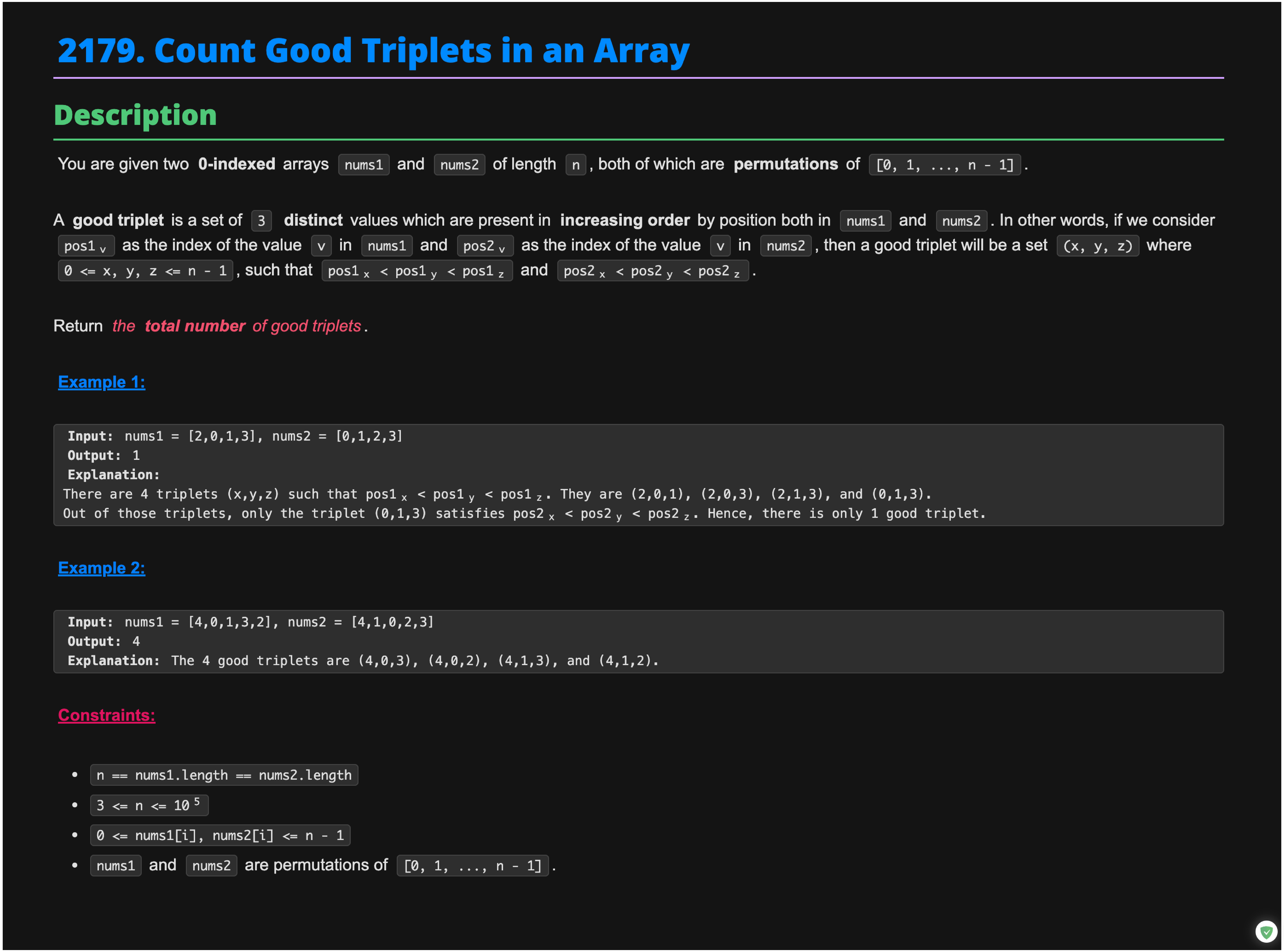

2179. Count Good Triplets in an Array

Description

You are given two **0-indexed** arrays `nums1` and `nums2` of length `n`, both of which are **permutations** of `[0, 1, ..., n - 1]`.

A **good triplet** is a set of `3` **distinct** values which are present in **increasing order** by position both in `nums1` and `nums2`. In other words, if we consider $pos1_v$ as the index of the value `v` in `nums1` and $pos2_v$ as the index of the value `v` in `nums2`, then a good triplet will be a set `(x, y, z)` where `0 <= x, y, z <= n - 1`, such that $pos1_x < pos1_y < pos1_z$ and $pos2_x < pos2_y < pos2_z$.

Return *the* ***total number*** *of good triplets*.

Example 1:

```
Input: nums1 = [2,0,1,3], nums2 = [0,1,2,3]
Output: 1
Explanation:
There are 4 triplets (x,y,z) such that pos1_x < pos1_y < pos1_z. They are (2,0,1), (2,0,3), (2,1,3), and (0,1,3).
Out of those triplets, only the triplet (0,1,3) satisfies pos2_x < pos2_y < pos2_z. Hence, there is only 1 good triplet.
```

Example 2:

```
Input: nums1 = [4,0,1,3,2], nums2 = [4,1,0,2,3]
Output: 4
Explanation: The 4 good triplets are (4,0,3), (4,0,2), (4,1,3), and (4,1,2).
```

Constraints:

- `n == nums1.length == nums2.length`
- $3 <= n <= 10^5$
- `0 <= nums1[i], nums2[i] <= n - 1`
- `nums1` and `nums2` are permutations of `[0, 1, ..., n - 1]`.