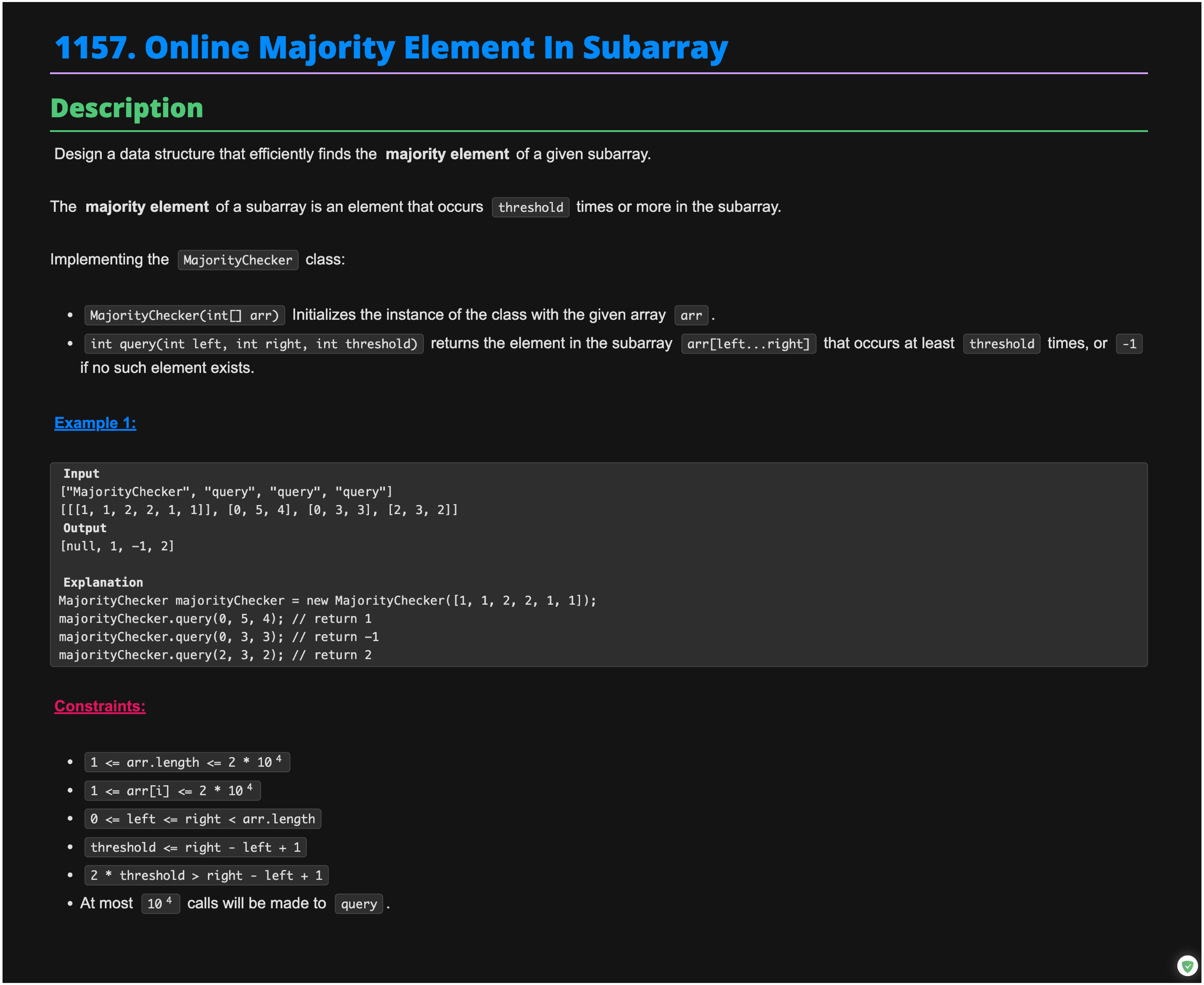

1157. Online Majority Element In Subarray

Description

Design a data structure that efficiently finds the **majority element** of a given subarray.

The **majority element** of a subarray is an element that occurs `threshold` times or more in the subarray.

Implementing the `MajorityChecker` class:

- `MajorityChecker(int[] arr)` Initializes the instance of the class with the given array `arr`.
- `int query(int left, int right, int threshold)` returns the element in the subarray `arr[left...right]` that occurs at least `threshold` times, or `-1` if no such element exists.

Example 1:

```
Input
["MajorityChecker", "query", "query", "query"]
[[[1, 1, 2, 2, 1, 1]], [0, 5, 4], [0, 3, 3], [2, 3, 2]]
Output
[null, 1, -1, 2]

Explanation
MajorityChecker majorityChecker = new MajorityChecker([1, 1, 2, 2, 1, 1]);
majorityChecker.query(0, 5, 4); // return 1
majorityChecker.query(0, 3, 3); // return -1
majorityChecker.query(2, 3, 2); // return 2
```

Constraints:

- $1 <= arr.length <= 2 * 10^4$
- $1 <= arr[i] <= 2 * 10^4$
- `0 <= left <= right < arr.length`
- `threshold <= right - left + 1`
- `2 * threshold > right - left + 1`
- At most 10^4 calls will be made to `query`.