Problem 1157: Online Majority Element In Subarray

Problem Information

Difficulty: Hard
Acceptance Rate: 39.80%
Paid Only: No
Tags: Array, Binary Search, Design, Binary Indexed Tree, Segment Tree

Problem 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 * 104` * `1 <= arr[i] <= 2 * 104` * `0 <= left <= right < arr.length` * `threshold <= right - left + 1` * `2 * threshold > right - left + 1` * At most `104` calls will be made to `query`.

Code Snippets

C++:

```cpp
class MajorityChecker {
public:
MajorityChecker(vector<int>& arr) {

}

int query(int left, int right, int threshold) {

}
};

/**
 * Your MajorityChecker object will be instantiated and called as such:
 * MajorityChecker* obj = new MajorityChecker(arr);
 * int param_1 = obj->query(left,right,threshold);
 */
```

Java:

```java
class MajorityChecker {

public MajorityChecker(int[] arr) {

}

public int query(int left, int right, int threshold) {

}
}

/**
 * Your MajorityChecker object will be instantiated and called as such:
 * MajorityChecker obj = new MajorityChecker(arr);
 * int param_1 = obj.query(left,right,threshold);
 */
```

Python3:

```python
class MajorityChecker:

    def __init__(self, arr: List[int]):


    def query(self, left: int, right: int, threshold: int) -> int:



# Your MajorityChecker object will be instantiated and called as such:
# obj = MajorityChecker(arr)
# param_1 = obj.query(left,right,threshold)
```