Problem 3679: Minimum Discards to Balance Inventory

Problem Information

Difficulty: Medium
Acceptance Rate: 34.25%
Paid Only: No
Tags: Array, Hash Table, Sliding Window, Simulation, Counting

Problem Description

You are given two integers `w` and `m`, and an integer array `arrivals`, where `arrivals[i]` is the type of item arriving on day `i` (days are **1-indexed**).

Items are managed according to the following rules:

* Each arrival may be **kept** or **discarded** ; an item may only be discarded on its arrival day. * For each day `i`, consider the window of days `[max(1, i - w + 1), i]` (the `w` most recent days up to day `i`): * For **any** such window, each item type may appear **at most** `m` times among kept arrivals whose arrival day lies in that window. * If keeping the arrival on day `i` would cause its type to appear **more than** `m` times in the window, that arrival **must** be discarded.

Return the **minimum** number of arrivals to be discarded so that every `w`-day window contains at most `m` occurrences of each type.

Example 1:

Input: arrivals = [1,2,1,3,1], w = 4, m = 2

Output: 0

Explanation:

* On day 1, Item 1 arrives; the window contains no more than `m` occurrences of this type, so we keep it. * On day 2, Item 2 arrives; the window of days 1 - 2 is fine. * On day 3, Item 1

arrives, window `[1, 2, 1]` has item 1 twice, within limit. * On day 4, Item 3 arrives, window `[1, 2, 1, 3]` has item 1 twice, allowed. * On day 5, Item 1 arrives, window `[2, 1, 3, 1]` has item 1 twice, still valid.

There are no discarded items, so return 0.

Example 2:

Input: arrivals = [1,2,3,3,3,4], w = 3, m = 2

Output: 1

Explanation:

* On day 1, Item 1 arrives. We keep it. * On day 2, Item 2 arrives, window `[1, 2]` is fine. * On day 3, Item 3 arrives, window `[1, 2, 3]` has item 3 once. * On day 4, Item 3 arrives, window `[2, 3, 3]` has item 3 twice, allowed. * On day 5, Item 3 arrives, window `[3, 3, 3]` has item 3 three times, exceeds limit, so the arrival must be discarded. * On day 6, Item 4 arrives, window `[3, 4]` is fine.

Item 3 on day 5 is discarded, and this is the minimum number of arrivals to discard, so return 1.

Constraints:

* `1 <= arrivals.length <= 105` * `1 <= arrivals[i] <= 105` * `1 <= w <= arrivals.length` * `1 <= m <= w`

Code Snippets

C++:

```
class Solution {
public:
int minArrivalsToDiscard(vector<int>& arrivals, int w, int m) {

}
};
```

Java:

```
class Solution {
public int minArrivalsToDiscard(int[] arrivals, int w, int m) {

}
}
```

Python3:

```
class Solution:
def minArrivalsToDiscard(self, arrivals: List[int], w: int, m: int) -> int:
```