Problem 2190: Most Frequent Number Following Key In an Array

Problem Information

Difficulty: Easy
Acceptance Rate: 59.32%
Paid Only: No
Tags: Array, Hash Table, Counting

Problem Description

You are given a **0-indexed** integer array `nums`.**** You are also given an integer `key`, which is present in `nums`.

For every unique integer `target` in `nums`, **count** the number of times `target` immediately follows an occurrence of `key` in `nums`. In other words, count the number of indices `i` such that:

* `0 <= i <= nums.length - 2`, * `nums[i] == key` and, * `nums[i + 1] == target`.

Return _the_`target` _with the**maximum** count_. The test cases will be generated such that the `target` with maximum count is unique.

Example 1:

Input: nums = [1,100,200,1,100], key = 1 **Output:** 100 **Explanation:** For target = 100, there are 2 occurrences at indices 1 and 4 which follow an occurrence of key. No other integers follow an occurrence of key, so we return 100.

Example 2:

Input: nums = [2,2,2,2,3], key = 2 **Output:** 2 **Explanation:** For target = 2, there are 3 occurrences at indices 1, 2, and 3 which follow an occurrence of key. For target = 3, there is only one occurrence at index 4 which follows an occurrence of key. target = 2 has the maximum number of occurrences following an occurrence of key, so we return 2.

Constraints:

* `2 <= nums.length <= 1000` * `1 <= nums[i] <= 1000` * The test cases will be generated such that the answer is unique.

Code Snippets

C++:

```
class Solution {
public:
int mostFrequent(vector<int>& nums, int key) {

}
};
```

Java:

```
class Solution {
public int mostFrequent(int[] nums, int key) {

}
}
```

Python3:

```
class Solution:
def mostFrequent(self, nums: List[int], key: int) -> int:
```