Problem 2350: Shortest Impossible Sequence of Rolls

Problem Information

Difficulty: Hard
Acceptance Rate: 69.21%
Paid Only: No
Tags: Array, Hash Table, Greedy

Problem Description

You are given an integer array `rolls` of length `n` and an integer `k`. You roll a `k` sided dice numbered from `1` to `k`, `n` times, where the result of the `ith` roll is `rolls[i]`.

Return _the length of the**shortest** sequence of rolls so that there's no such subsequence in _`rolls`.

A **sequence of rolls** of length `len` is the result of rolling a `k` sided dice `len` times.

Example 1:

Input: rolls = [4,2,1,2,3,3,2,4,1], k = 4 **Output:** 3 **Explanation:** Every sequence of rolls of length 1, [1], [2], [3], [4], can be taken from rolls. Every sequence of rolls of length 2, [1, 1], [1, 2], ..., [4, 4], can be taken from rolls. The sequence [1, 4, 2] cannot be taken from rolls, so we return 3. Note that there are other sequences that cannot be taken from rolls.

Example 2:

Input: rolls = [1,1,2,2], k = 2 **Output:** 2 **Explanation:** Every sequence of rolls of length 1, [1], [2], can be taken from rolls. The sequence [2, 1] cannot be taken from rolls, so we return 2. Note that there are other sequences that cannot be taken from rolls but [2, 1] is the shortest.

Example 3:

Input: rolls = [1,1,3,2,2,2,3,3], k = 4 **Output:** 1 **Explanation:** The sequence [4] cannot be taken from rolls, so we return 1. Note that there are other sequences that cannot be taken from rolls but [4] is the shortest.

Constraints:

* `n == rolls.length` * `1 <= n <= 105` * `1 <= rolls[i] <= k <= 105`

Code Snippets

C++:

```
class Solution {
public:
int shortestSequence(vector<int>& rolls, int k) {

}
};
```

Java:

```
class Solution {
public int shortestSequence(int[] rolls, int k) {

}
}
```

Python3:

```
class Solution:
def shortestSequence(self, rolls: List[int], k: int) -> int:
```