Problem 3176: Find the Maximum Length of a Good Subsequence I

Problem Information

Difficulty: Medium
Acceptance Rate: 32.21%
Paid Only: No
Tags: Array, Hash Table, Dynamic Programming

Problem Description

You are given an integer array `nums` and a **non-negative** integer `k`. A sequence of integers `seq` is called **good** if there are **at most** `k` indices `i` in the range `[0, seq.length - 2]` such that `seq[i] != seq[i + 1]`.

Return the **maximum** possible length of a **good** subsequence of `nums`.

Example 1:

Input: nums = [1,2,1,1,3], k = 2

Output: 4

Explanation:

The maximum length subsequence is `[_1_ ,_2_ ,_1_ ,_1_ ,3]`.

Example 2:

Input: nums = [1,2,3,4,5,1], k = 0

Output: 2

Explanation:

The maximum length subsequence is `[_1_ ,2,3,4,5,_1_]`.

Constraints:

* `1 <= nums.length <= 500` * `1 <= nums[i] <= 109` * `0 <= k <= min(nums.length, 25)`

Code Snippets

C++:

```
class Solution {
public:
int maximumLength(vector<int>& nums, int k) {

}
};
```

Java:

```
class Solution {
public int maximumLength(int[] nums, int k) {

}
}
```

Python3:

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