Problem 3576: Transform Array to All Equal Elements

Problem Information

Difficulty: Medium
Acceptance Rate: 32.77%
Paid Only: No
Tags: Array, Greedy

Problem Description

You are given an integer array `nums` of size `n` containing only `1` and `-1`, and an integer `k`.

You can perform the following operation at most `k` times:

* Choose an index `i` (`0 <= i < n - 1`), and **multiply** both `nums[i]` and `nums[i + 1]` by `-1`.

Note that you can choose the same index `i` more than once in **different** operations.

Return `true` if it is possible to make all elements of the array **equal** after at most `k` operations, and `false` otherwise.

Example 1:

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

Output: true

Explanation:

We can make all elements in the array equal in 2 operations as follows:

* Choose index `i = 1`, and multiply both `nums[1]` and `nums[2]` by -1. Now `nums = [1,1,-1,-1,1]`. * Choose index `i = 2`, and multiply both `nums[2]` and `nums[3]` by -1. Now

`nums = [1,1,1,1,1]`.

Example 2:

Input: nums = [-1,-1,-1,1,1,1], k = 5

Output: false

Explanation:

It is not possible to make all array elements equal in at most 5 operations.

Constraints:

* `1 <= n == nums.length <= 105` * `nums[i]` is either -1 or 1. * `1 <= k <= n`

Code Snippets

C++:

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

}
};
```

Java:

```java
class Solution {
public boolean canMakeEqual(int[] nums, int k) {

}
}
```

Python3:

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