Problem 1437: Check If All 1's Are at Least Length K Places Away

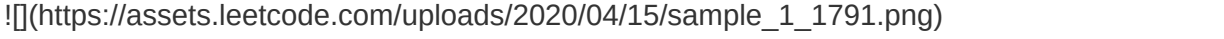
Problem Information

Difficulty: Easy
Acceptance Rate: 64.26%
Paid Only: No
Tags: Array

Problem Description

Given an binary array `nums` and an integer `k`, return `true` _if all_`1` _' s are at least _`k` _places away from each other, otherwise return_`false`.

Example 1:



Input: nums = [1,0,0,0,1,0,0,1], k = 2 **Output:** true **Explanation:** Each of the 1s are at least 2 places away from each other.

Example 2:

Input: nums = [1,0,0,1,0,1], k = 2 **Output:** false **Explanation:** The second 1 and third 1 are only one apart from each other.

Constraints:

* `1 <= nums.length <= 105` * `0 <= k <= nums.length` * `nums[i]` is `0` or `1`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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