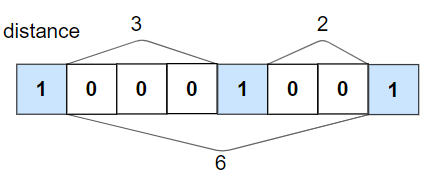
# 题目

给你一个由若干0和1组成的数组nums以及整数k。如果所有1都至少相隔k个元素，则返回True；否则，返回False。

示例 1：

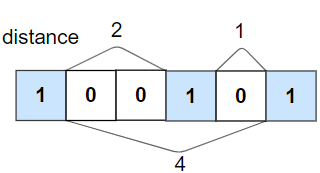


输入：nums = [1,0,0,0,1,0,0,1], k = 2

输出：true

解释：每个 1 都至少相隔 2 个元素。

示例 2：



输入：nums = [1,0,0,1,0,1], k = 2

输出：false

解释：第二个 1 和第三个 1 之间只隔了 1 个元素。

示例 3：

输入：nums = [1,1,1,1,1], k = 0

输出：true

示例 4：

输入：nums = [0,1,0,1], k = 1

输出：true

提示：

1 <= nums.length <= 10^5

0 <= k <= nums.length

nums[i] 的值为0或1

# 分析

## 方法一：遍历

**思路：**

「所有1都至少相隔k个元素」等价于「任意两个相邻的1都至少相隔k个元素」，因此我们只需要从左到右遍历数组，并记录上一个1出现的位置。

在遍历的过程中，如果我们找到了一个新的1，就需要判断其与上一个1之间是否至少相隔k个元素。如果不满足要求，那么直接返回False作为答案，否则继续进行遍历。

在遍历完成之后即可返回True作为答案。

**代码：**

class Solution {

public:

bool kLengthApart(vector<int>& nums, int k) {

int n = nums.size();

int prev = -1;

for (int i = 0; i < n; ++i) {

if (nums[i] == 1) {

if (prev != -1 && i - prev - 1 < k) {

return false;

}

prev = i;

}

}

return true;

}

};

**复杂度分析：**

时间复杂度：O(N)，其中N是数组nums的长度。

空间复杂度：O(1)。