Problem 1703: Minimum Adjacent Swaps for K Consecutive Ones

Problem Information

Difficulty: Hard
Acceptance Rate: 42.27%
Paid Only: No
Tags: Array, Greedy, Sliding Window, Prefix Sum

Problem Description

You are given an integer array, `nums`, and an integer `k`. `nums` comprises of only `0`'s and `1`'s. In one move, you can choose two **adjacent** indices and swap their values.

Return _the**minimum** number of moves required so that _`nums` _has_`k` _**consecutive** _`1` _' s_.

Example 1:

Input: nums = [1,0,0,1,0,1], k = 2 **Output:** 1 **Explanation:** In 1 move, nums could be [1,0,0,0,_1_ ,_1_] and have 2 consecutive 1's.

Example 2:

Input: nums = [1,0,0,0,0,0,1,1], k = 3 **Output:** 5 **Explanation:** In 5 moves, the leftmost 1 can be shifted right until nums = [0,0,0,0,0,_1_ ,_1_ ,_1_].

Example 3:

Input: nums = [1,1,0,1], k = 2 **Output:** 0 **Explanation:** nums already has 2 consecutive 1's.

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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