

5624. Minimum Adjacent Swaps for K Consecutive Ones

[My Submissions \(/contest/biweekly-contest-42/problems/minimum-adjacent-swaps-for-k-consecutive-ones/submissions/\)](/contest/biweekly-contest-42/problems/minimum-adjacent-swaps-for-k-consecutive-ones/submissions/)[Back to Contest \(/contest/biweekly-contest-42/\)](/contest/biweekly-contest-42/)

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.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Constraints:

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

JavaScript



```
1 /**
2  * @param {number[]} nums
3  * @param {number} k
4  * @return {number}
5  */
6 var minMoves = function(nums, k) {
7
8 };
```

☐ Custom Testcase [Use Example Testcases](#)