Intuition

We are tasked with finding the shortest subarray whose bitwise OR is greater than or equal to a given value k. We can use a sliding window approach to explore each subarray while maintaining the bitwise OR of the elements in the window. If the OR value satisfies the condition, we try to minimize the window's length.

Approach

1. We use two pointers (left and right) to represent a sliding window over the array.
2. We maintain a bitCount array to keep track of the count of set bits at each position.
3. For each new element in the window, we update the bitwise OR.
4. Whenever the bitwise OR of the window meets or exceeds k, we try to shrink the window from the left to find the shortest possible valid subarray.
5. Return the length of the shortest valid subarray, or -1 if no such subarray exists.

Time Complexity: O(32*N)

Space Complexity: O(32)