

# Problem 3171: Find Subarray With Bitwise OR Closest to K

## Problem Information

**Difficulty:** Hard

**Acceptance Rate:** 30.45%

**Paid Only:** No

**Tags:** Array, Binary Search, Bit Manipulation, Segment Tree

## Problem Description

You are given an array `nums` and an integer `k`. You need to find a subarray of `nums` such that the \*\*absolute difference\*\* between `k` and the bitwise `OR` of the subarray elements is as\*\*small\*\* as possible. In other words, select a subarray `nums[l..r]` such that `|k - (nums[l] OR nums[l + 1] ... OR nums[r])|` is minimum.

Return the \*\*minimum\*\* possible value of the absolute difference.

A \*\*subarray\*\* is a contiguous \*\*non-empty\*\* sequence of elements within an array.

**Example 1:**

**Input:** nums = [1,2,4,5], k = 3

**Output:** 0

**Explanation:**

The subarray `nums[0..1]` has `OR` value 3, which gives the minimum absolute difference `|3 - 3| = 0`.

**Example 2:**

**Input:** nums = [1,3,1,3], k = 2

**\*\*Output:\*\*** 1

**\*\*Explanation:\*\***

The subarray `nums[1..1]` has `OR` value 3, which gives the minimum absolute difference `|3 - 2| = 1`.

**\*\*Example 3:\*\***

**\*\*Input:\*\*** nums = [1], k = 10

**\*\*Output:\*\*** 9

**\*\*Explanation:\*\***

There is a single subarray with `OR` value 1, which gives the minimum absolute difference `|10 - 1| = 9`.

**\*\*Constraints:\*\***

\* `1 <= nums.length <= 105` \* `1 <= nums[i] <= 109` \* `1 <= k <= 109`

## Code Snippets

**C++:**

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

**Java:**

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

```
}
```

### Python3:

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