SET-A

You are given two integers **left** and **right**, representing an inclusive range of integers . Determine the result obtained by applying the **bitwise AND** operation to all integers within this range.

Your solution should be designed to achieve a **time complexity better than linear time**, without iterating through every number in the range.

Example 1:

Input: left = 5, right = 7

Output: 4

Example 2:

Input: left = 0, right = 0

Output: 0

Example 3:

Input: left = 1, right = 2147483647

Output: 0

Constraints:

- $0 <= left <= right <= 2^{31} - 1$

SET-B

Given an integer array nums and an integer k, return *the* k^{th} *largest element in the array*.

Note that it is the k^{th} largest element in the sorted order, not the k^{th} distinct element.

Can you solve it without sorting?

Example 1:

Input: nums = [3,2,1,5,6,4], k = 2

Output: 5

Example 2:

Input: nums = [3,2,3,1,2,4,5,5,6], k = 4

Output: 4

Constraints:

- 1 <= k <= nums.length <= 10^5
- -10^4 <= nums[i] <= 10^4