898. Bitwise ORs of Subarrays

Medium 410 85 Add to List Share

We have an array `A` of non-negative integers.

For every (contiguous) subarray `B = [A[i], A[i+1], ..., A[j]]` (with `i <= j`), we take the bitwise OR of all the elements in `B`, obtaining a result `A[i] | A[i+1] | ... | A[j]`.

Return the number of possible results. (Results that occur more than once are only counted once in the final answer.)

Example 1:

```
Input: [0]
Output: 1
Explanation:
There is only one possible result: 0.
```

Example 2:

```
Input: [1,1,2]
Output: 3
Explanation:
The possible subarrays are [1], [1], [2], [1, 1], [1, 2], [1, 1, 2].
These yield the results 1, 1, 2, 1, 3, 3.
There are 3 unique values, so the answer is 3.
```

Example 3:

```
1 # @param {Integer[]} a
2 # @return {Integer}
3 def subarray_bitwise_o_
4
5 end
```