Problem 898: Bitwise ORs of Subarrays

Problem Information

Difficulty: Medium
Acceptance Rate: 56.63%
Paid Only: No
Tags: Array, Dynamic Programming, Bit Manipulation

Problem Description

Given an integer array `arr`, return _the number of distinct bitwise ORs of all the non-empty subarrays of_ `arr`.

The bitwise OR of a subarray is the bitwise OR of each integer in the subarray. The bitwise OR of a subarray of one integer is that integer.

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

Example 1:

Input: arr = [0] **Output:** 1 **Explanation:** There is only one possible result: 0.

Example 2:

Input: arr = [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:

Input: arr = [1,2,4] **Output:** 6 **Explanation:** The possible results are 1, 2, 3, 4, 6, and 7.

Constraints:

* `1 <= arr.length <= 5 * 104` * `0 <= arr[i] <= 109`

Code Snippets

C++:

```cpp
class Solution {
public:
int subarrayBitwiseORs(vector<int>& arr) {

}
};
```

Java:

```java
class Solution {
public int subarrayBitwiseORs(int[] arr) {

}
}
```

Python3:

```python
class Solution:
def subarrayBitwiseORs(self, arr: List[int]) -> int:
```