

我们有一个非负整数数组 A 。

对于每个（连续的）子数组 $B = [A[i], A[i+1], \dots, A[j]]$ ($i \leq j$)，我们对 B 中的每个元素进行按位或操作，获得结果 $A[i] \mid A[i+1] \mid \dots \mid A[j]$ 。

返回可能结果的数量。（多次出现的结果在最终答案中仅计算一次。）

示例 1:

输入: [0]

输出: 1

解释:

只有一个可能的结果 0。

示例 2:

输入: [1,1,2]

输出: 3

解释:

可能的子数组为 [1], [1], [2], [1, 1], [1, 2], [1, 1, 2]。

产生的结果为 1, 1, 2, 1, 3, 3。

有三个唯一值，所以答案是 3。

示例 3:

输入: [1,2,4]

输出: 6

解释:

可能的结果是 1, 2, 3, 4, 6, 以及 7。

提示:

1. $1 \leq A.length \leq 50000$

2. $0 \leq A[i] \leq 10^9$

思路，其实自己想的和他差不多，但是自己把他看成了分支，走到最底层，导致消耗非常大，而作者代码的思路就是剪纸了，每条路径只走可能的结果，然后动态统计，然后可能结果往下走，像我走 0 0 0 1 1 0 0 0 作者就走 0 0 0 1 1 就不往下进行计算了。

代码:

```
class Solution {  
public:
```

```
int subarrayBitwiseORs(vector<int>& A) {  
    unordered_set<int> res, cur, cur2;  
    for (int i: A) {  
        cur2 = {i};  
        for (int j: cur) cur2.insert(i|j);  
        cur = cur2;  
        for (int j: cur) res.insert(j);  
    }  
    return res.size();  
}  
};
```