

Nice girl

由于位运算各个位是独立的，因此将 p 拆开 31 位来算。

考虑第 i 位：

1、 p 的第 i 位为 0，那么 $a[l], a[l+1]..a[r]$ 的第 i 位必须为 0

2、 p 的第 i 位为 1，那么 $a[l], a[l+1]..a[r]$ 的第 i 位中有一个为 1

那么就是有些值是强制为 0 的，有些值是可选的，但是可选的情况下要保证在约束区间内至少有一个 1。

Nice girl

第一步：扫描所有约束，对于 p 的第 i 位为 0 的，用在 l, r 处用左括号 / 右括号标记。复杂度 $o(m)$

第二步：扫描 $1 \sim n$ 的节点，遇到左括号计数器加 1，遇到右括号减 1。当计数器非 0 的时候说明这个节点必须为 0。记录下必须为 0 的位置，并处理成前缀和。复杂度 $o(n)$

第三步：扫描所有约束，对于 p 的第 i 位为 1 的，用上面处理的前缀和求 $[l, r]$ 区间必须为 0 的个数，如果等于区间长度，那么说明一个 1 都不可能，无解。复杂度 $o(m)$

Nice girl

输出字典序最大串
不要求是 0 的地方都填 1

输出字典序最小串：
在保证约束的情况下尽可能往右边填 1。
扫描 $1 \sim n$ 节点，处理出每个节点的左边最近的不是强制为 0 的节点位置。 $O(n)$
对于 p_i 为 1 的区间，在右端点的节点处记下来这个区间，最后 $1 \sim n$ 扫一遍，扫的过程中记录最右边的已填 1 的位置 p ，碰到有标记的节点检验是否标记的区间内含有 p 。如果有那么继续下一个，否则将左边最近的不是强制为 0 的节点变成 1。