紫丁香

先考虑怎么解决一次询问。

二分答案，现在问题转化为给定一个位的集合 B，求是否存在一种方式操作完后 B 中的位全都是 `1`。将 B 的限制写成一个由 `1,*` 组成的串 R，为 `1` 的位表示操作完之后要是 1，为 `*` 的位表示对这一位没有限制。

现在考虑最后一次操作，设这次操作对应的串是 T_i，那么：

- R 中一个为 `*` 的位，T_i 中可以是 `0,1,-` 中的任何一个，因为没有限制。
- R 中一个为 `1` 的位，T_i 中只能是 `1,-`，因为否则最后就会是 0。

假设某个 R 中为 `1` 的位，在 T_i 中也是 `1`，那么这一位在更之前的操作中就没有限制了（反正最后一步会变成 `1`），所以我们可以将 R 的这一位改成 `*`。

于是判定的过程可以看成这样：

- 每一轮，选出一个满足条件的 T_i，然后把 R,T_i 中都为 `1` 的位在 R 中改成 `*`。
- 重复上述过程直到 R 无法再被更新。

最终，R 可能还剩下一些 `1`，这些 `1` 无法通过操作产生，我们只需检验初始串 S 中这些位置是不是都是 `1`，如果是则说明判定成功，否则判定失败。

直接这么做复杂度可能是 $O(nqm^2)$ 的。

注意到判定过程除了最后一步之外只和 R 有关，而 R 只有 2^m 种，所以我们不妨对 R 进行 DP。设 $f(R)$ 表示 R 通过上面的更新过程能更新到的 `1` 的数量最小的串，再设 $g(R)$ 表示仅从 R 开始进行一次操作能够变成 `*` 的位的并。

那么，$f(R)=g^{\infty}(R)$，而 $g(R)$ 的计算是简单的：比如把操作记在某个对应的 R_0 上，然后做个高维前缀或即可。

预处理 $f(R)$ 后，我们就可以 $O(1)$ 或 $O(m)$ 地进行二分答案中的一次判定。

总复杂度 $O((n+q+2^m)\times m)$。