

NOIP 2019 模拟赛 Contest 5

diamond_duke

题目名称	染色	乘方	位运算
可执行文件名	color	power	bitop
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
时间限制	1s	3s	1s
内存限制	512MB	512MB	512MB
子任务个数	4	5	6
题目类型	传统型	传统型	传统型

请注意： 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

1 染色

我们可以把染黑看成删掉这条边，那么每次相当于是，如果存在一个点有且仅有一条出边，那么把这条边也删掉，最后要删完所有的边。

那么我们考虑一个环，可以发现这个环如果一条边都不删，那么最后一定会被留下来，因为每个点都至少有两个出边。

所以，在初始的边删完之后，图一定得无环。我们考虑一个无环的图，可以发现他就是一个森林。考虑森林的叶子节点，可以发现每次叶子节点的父边都会被删掉，而删掉之后森林还是森林。因此，在有限步操作之后，所有的边都会被删掉，所以无环就一定合法了。

因此我们最后的答案就是，给每个联通块都保留一个生成树，其他边全都删掉。

考虑这个东西怎么算。因为一棵生成树的边数是点数减一，所有联通块的总点数是 n ，所以如果设联通块数为 C ，则最后剩下的边个数就是 $n - C$ ，因此答案就是 $m - n + C$ 。

时间复杂度： $\Theta(n\alpha(n) + m)$ 。

2 乘方

考虑二分答案，则问题变为了求 $[1, M]$ 内有多少个数字在至少一个 $S(n_i)$ 里面。

考虑容斥：我们考虑先计算出有多少个 $[1, M]$ 中的数在 $S(n_1), S(n_2), \dots, S(n_k)$ 中，然后把它们都加起来。

显然这样算多了：如果某个数字同时在多个 $S(n_i)$ 里面，那么他就重复了。于是我们考虑再去掉在至少两个 $S(n_i)$ 中的，然后加上至少三个 $S(n_i)$ 中的，去掉至少四个 $S(n_i)$ 中的……

那么我们现在相当于是需要考虑一个 $\{n_1, n_2, \dots, n_k\}$ 的子集 N ，计算有多少个数字 $x \in [1, M]$ ，满足 $\forall t \in N$ ，有 $x \in S(t)$ 。可以发现， $\bigcap_{t \in N} S(t) = S(\text{lcm}_{t \in N}(t))$ ，因此可以直接二分得到这样的 x 个数。

直接枚举 S 进行计算的时间复杂度为 $\Theta(q \cdot 2^k \cdot \log_2^2 M)$ ，可以通过子任务 3。

注意到我们最终的答案只与 $\text{lcm}_{t \in N}(t)$ 有关，因此我们可以考虑 DP：设 $f_{i,t}$ 表示考虑 n_1, n_2, \dots, n_i ，从中选出一个子集，满足他们的 lcm 是 t 的容斥系数之和。

注意到当 $\text{lcm}_{t \in N}(t) \geq 60$ 时，一定不存在这样的 x （因为答案不超过 10^{17} ，所以 $M \leq 10^{17}$ ）。于是在 DP 的第二个维度只保留不超过 60 的那些值即可。

时间复杂度： $\Theta(q(k \log_2 M + \log_2^3 M))$ 。

3 位运算

考虑我们如何比较两个数的大小，可以发现是按照从高位到低位的顺序依次比较，直到第一个不同的位出现，那么两个数的大小关系就是这个不同的位的大小关系。值得一提的是，这里我们认为较短的数字已经在前面补前导 0 使得两个数字长度相同。

那么对于本题，在二进制下比较方式同样不变，因此我们可以按照从高位到低位的顺序逐位确定，这样就可以得到最大值了。

3.1 按位与

我们按照从高位到低位的顺序进行贪心，如果当前位有至少两个数字是 1，那么我们可以让这一位是 1，因此最终答案的这一位一定是 1。

那么如果我们确定了某一位是 1，我们最后选择的两个数的这一位都得是 1，所以我们可以去掉所有这一位为 0 的数字，然后继续向下贪心即可。最后的方案数就是剩余的数字中选出两个的方案数。

时间复杂度： $\Theta(n \log_2 a_i)$ 。

3.2 按位异或

同样地从高到低贪心，逐位确定。但是直接确定下来某一位是 1 并不能和按位与那样减少候选范围。

考虑先确定最后选择的一个数字 a_i ，考虑如何求出另一个 a_j 使得 $a_i \text{ xor } a_j$ 最大。

考虑将所有数字建出一个 trie 树，然后同样从高到低进行贪心：如果当前位可以取得和 a_i 这一位不一样，那么就取不一样的，否则取一样的。方案数可以在 trie 树的每个节点上存储对应数字个数得到。

时间复杂度： $\Theta(n \log_2 a_i)$ 。

3.3 按位或

同样地在确定了一个数之后从高到低贪心。但是这里又多了一个问题：如果这个数字的当前位是 1，那么我们仍然没法减少候选范围。

那么在贪心到某一步之后，相当于给定一个位的集合 S ，判断是否一个数的这些位都是 1，其余位不管。接下来我们用 A_i 表示数字 a_i 中 1 的位置集合。

考虑 DP（事实上，这个 DP 方法叫做**高维前缀和**）：设 $f_{i,S}$ 表示使得 $S \subseteq A_t$ 且 $A_t \setminus S \subseteq \{1, 2, \dots, i\}$ 的 t 的个数。那么转移时，考虑 a_t 的当前位是 0 还是 1，然后可以变成 $f_{i-1,S}$ 以及 $f_{i-1,S \setminus \{i\}}$ 两种，直接相加即可。

那么我们就可以判断是否存在一个合法的数字了。最后的方案数同样可以使用这个 DP 数组求出。时间复杂度： $\Theta(a_i \log_2 a_i)$ 。