

注：数字是考场分数，括号是题解对应得分。

- D1T1 set 集合 100

考虑每个数 x 出现的位置集合 S_x ，我们注意到两个序列可以通过将元素乘置换得到当且仅当两个排列的所有 S_x 组成的可重集相同。判断两个集合相等可以使用哈希，给每个位置赋一个 $0, 1^{64}$ 的随机权值，一个集合的权值就是所有位置的值的异或和。

这样两个不同集合的权值相等的概率显然是 $\frac{1}{2^{64}}$ ，可以近似地认为所有区间的权值两两不同，开一个 `unordered_map` 即可。直接上莫队做是 $O(n\sqrt{m})$ 的，可以获得一些分数。

考虑一个事实，让一个大区间满足条件比让包含它的小区间满足条件严格难。

也就是说，对于每个 l ，存在最大的 r 满足区间 $[l, t]$ 合法当且仅当 $t \in [l, r]$ ，且 r 随 l 递增是单调增的。这样可以双指针计算，时间复杂度即可降低到 $O(n)$ 。

- D1T2 richest 百万富翁 82(100)

注意到每次询问的信息是团时可以去除最多的可能，因此我们一定将点集划分成若干个团，最后团的数量作为这一轮删除的结果。

根据小学奥数，我们会发现让每个团的大小尽可能接近是优的，所以可以做一个 DP，我们就会有 $f_{x,y}$ 代表 x 轮确定 y 个数中最大值的最小步数，这样是 $O(n^2)$ 的，你可以通过跑足够久，或者决策单调性，或者手动限制取值范围来求出最优方案。

- D1T3 tree 树的定向 64(100)

考察 A 性质，我们发现一个事实：如果不存在长度为 1 的路径，将边进行二染色之后显然存在一组解。证明是简单的，考虑相邻的边至少有一条不符合要求即可。

因此我们可以得到一个多项式复杂度的判定算法：不断确定长度为 1 的路径的边的方向并合并该边的端点，直到出现矛盾或存在一组解。直接逐位判断答案是 $O(n^2m)$ 的。

容易发现根据我们的判定规则，在不存在长度为 1 的路径时，由于将每条边都取反不影响合法性，所以显然将任意一条边取任意方向后仍然存在解。所以我们可以考虑一边确定边的方向，一边寻找长度为 1 的路径。

一个直接的想法是对树进行点分治然后做折半报警器 Trick，时间复杂度为 $O(n \log n + m \log^2 n)$ 。

- D2T1 fraction 分数 95(100)

因为可以取倒数，所以我们可以用一个二元组 (x, y) 代表分子和分母，规则可以重写如下：

- $(0, 1)$ 合法。
- 如果 (x, y) 合法，那么 $(2y + x, y)$ 合法。
- 如果 (x, y) 合法，那么 $(x, 2x + y)$ 合法。

经过简单的观察可以找到判定一个二元组是否合法的方法：将两个数做辗转相除法的过程写下来，如果到 $(0, 1)$ 前每一步的商都是偶数则合法，反之则不合法。

上述证明同样说明了一个数只有一种被构造的方式，所以我们可以直接搜索出所有的二元组，只需要将 (x, y) 转移到 $(2ty + x, y)$ 即可。时间复杂度 $O(\text{answer})$ 。

正解并没有什么本质的新观察，只是考虑将第一步展开：将 k 作为参数搜索所有的 $(ak+b, ck+d)$ 。此时很多状态都被合并，一个二元组也只需要考虑合法的 k 的个数即可。时间复杂度未知，但是可以通过。

- D2T2 mountain 登山 100

简单的观察是每次跳跃到的点显然都是比子树中所有点还浅的。

考虑从上到下求出 DP 值。在求出一个点 x 的路径数 f_x 之后更新所有下一次跳跃到 x 的方案数。先不考虑滑落，那么所有能到 x 并且到 x 不会违反 h_i 的点都会被更新一次。而如果可以滑落，那么通过可以不违反 h_i 到达 x 的所有 y 都可以被更新。

因为我们可以扫描线维护每个点被更新的次数，先从小到大枚举深度，那么贡献的部分就是历史和线段树，一次区间贡献在 $d_i - r_i$ 处加入， $d_i - l_i$ 处移除。加了 h_i 其实是类似的，只不过有些点会直接从我们考虑的树之中移除，转化成链加也是简单的。

如果你使用树链剖分维护树上区间历史和是 $O(n \log^2 n)$ ，全局平衡二叉树可以做到 $O(n \log n)$ 。

- D2T3 graphee 树形图 35

这题只写了暴力，对不起。

测试点 2,7：能到所有点的就是一类点，求出拓扑序最小的强连通分量即可。

测试点 3,4：直接求出每个点的一棵 DFS 树，点是一类点当且仅当树上仅有返祖边。

测试点 1：先爆搜出所有一类点。

考虑 $m \leq 20$ 的情况，对于每个边的子集重新搜一类点，将新的一类点归为二类点即可。

测试点 8,9：先求出一类点的一棵 DFS 树，一个剩余的点可以成为一类点当且仅当其子树内恰有一条比它浅的返祖边（否则它的父亲有两种走过去的方法），此时世界线收束。