T1 magic

因为 $x = \max(a,b), y = \max(a,c)$，所以 $\max(x,y) = \max(a,b,c)$

同理，$\max(y,z) = \max(x,z) = \max(a,b,c)$

当且仅当 x,y,z 中两两之间最大值都相同时才有答案，否则输出 NO

不妨设 $a \ge b \ge c$

那么有$x = a, y = b, z = c$

还有 $x = z \ge y$

也就是说有解必须满足其中有两个数相等，且这两个数都不小于第三个

60 pts 是 int 范围

100 pts 是 long long 范围

T2 count

枚举 x 的具体权值判定是否是回文数，复杂度 n 的三次根号。

T3 gcd

30 pts

枚举两个数并判断，$O(n^2 \log n)$

60 pts

根据 xor 的性质，$a \operatorname{xor} b = c$ 等价于 $a \operatorname{xor} c = b$

那么有 $\gcd(a,b) = \gcd(a, a \operatorname{xor} c) = c$

而其中 c 是 a 的约数，因此我们可以枚举 c，再枚举 $a = i \times c$，由于 $\frac{n}{1} + \frac{n}{2} + \frac{n}{3} + \ldots + \frac{n}{n} = O(n \log n)$，再算上 gcd 的一只 $\log$，总复杂度 $O(n \log^2 n)$

100 pts

因为 $a = b$ 是肯定无解，所以不妨设 $a > b$

那么有 $\gcd(a,b) \le a - b, a \operatorname{xor} b \ge a - b$，很明显有 $c = a - b$。那么我们在 60 pts 的算法基础上，有 $\gcd(a, a-c) = c$，所以我们只需判断 $a \operatorname{xor} c = a - c$即可，复杂度 $O(n \log n)$

T4 inverse

20 pts

指数暴力枚举每个机遇选或不选

50 pts

考虑一个 DP

$f(i,j,k)$ 表示经过 i 次机遇后，$a_j > a_k$ 的方案数

如何统计答案?

答案就是所有满足 $i < j$ 的 $f(m,i,j)$ 之和

如何转移?

枚举状态 i,j,k，设第 i 次机会是交换 a_x 和 a_y

若 x,y 和 j,k 完全不同，则 $f(i,j,k) = 2 \times f(i-1,j,k)$

若 $x = j, y \neq k$，则

$f(i,j,k) = f(i-1,x,k) + f(i-1,y,k)$，$f(i,k,j) = f(i-1,k,x) + f(i-1,k,y)$

若 $x = j, y = k$，则 $f(i,j,k) = f(i-1,j,k) + f(i-1,k,j)$

复杂度 $O(n^3)$

100 pts

怎么优化?

第一维可以滚动数组。

注意到，每次转移只有 $O(n)$ 个位置不是 $\times 2$，其他 $O(n^2)$ 个位置都是 $\times 2$

那么通过这个观察，我们能否优化掉这 $O(n^2)$ 个位置的转移呢?

答案是可以的，我们设

$g(i,j,k) = \frac{f(i,j,k)}{2^i}$，这样每次就不需要 $\times 2$ 了，最终答案 $\times 2^m$ 即可

时间复杂度 $O(nm)$