A - 纵使日薄西山

考虑对于两个不同的二元组 (i,j) 和 (k,l) ，如果 $a_i \oplus a_j = a_k \oplus a_l$ ，由于 a 中的数互不相等，所以不会存在 $(i,j),(i,k)$ 的情况，于是 (i,j,k,l) 一定是合法的。每个合法的 (i,j,k,l) 都会被这样算到三遍。我们令 t_x 是 $a_i \oplus a_j = x$ 的二元组个数，答案就是 $\frac{\sum \binom{t_x}{2}}{3}$ 。接下来有两种方法。

方法一：利用 FWT 求出 t_x ，直接得到答案。

方法二：利用答案对 m 取 min 的条件。考虑 n 较大的时候，$\sum \binom{t_x}{2} \geq \sum (t_x - 1) = \binom{n}{2} - 2^{18}$ ，于是当 $\binom{n}{2} \geq 3m + 2^{18}$ 时直接输出 m 即可， 否则暴力计算 t_x 并统计答案。

复杂度 $O(m+V)$ 或 $O(V \log V)$ 。

B - 即便看不到未来

为了方便，我们假设在 (x,y,z) 能射击到的点 (x',y',z') 满足 $x' \in [x, x+2k], y' \in [y, y+2k], z' \in [z, z+2k]$ ，显然这样不影响答案。我们令 $K = 2k$ ，只需要计算 K 的最小值并除以 2 上取整。

找出所有点中 x,y,z 的最小值 x_a, y_b, z_c 以及它们对应的点 a,b,c 。我们可以贪心地走到 (x_a, y_b, z_c) ，这是不影响答案的。

此时必须要击杀 a,b,c 中的一个，对于点 a 我们计算 $v_a = \max(y_a - y_b, z_a - z_c)$ 表示击杀它所需的 K 最小值，同理得到 v_b, v_c 。那我们就能得到 $K \geq min(v_a, v_b, v_c)$ 。不妨令 v_a 最小，那此时就能忽略掉 a 这个点的影响，找出新的三个 a', b', c' 做下去。

复杂度 $O(n \log n)$ ，因为需要排序。

C - 此时此刻的光辉

做法 1：考虑折半，把点分成大小为 B 和 $n-B$ 的两部分。在两部分的内部处理出有哪些染色方案是满足条件的，再将其拼合起来：对于大小为 $n-B$ 的部分，枚举每个合法染色方案，我们将其对另一边的限制用一个长度为 $3B$ 的二进制数表示，第 $3i+j-2$ 位代表第 i 个点能不能染成颜色 j 。对于大小为 B 的部分的合法染色方案，我们也用一个二进制数表示，第 $3i+j-2$ 位代表第 i 个点是否染成颜色 j 。

做一次 FMT 即可。复杂度 $O(B2^{3B} + 3^{n-B} + 3^B)$ ，在 $B=6$ 或 7 时最优。

做法 2：考虑枚举颜色 2 的点的集合，它们给剩下的点带来了一些限制，此时有一些点 $0,1$ 都能取，有一些点只能取其中一种。此时我们首先需要判断新确定了颜色的点之间的限制是否满足；然后再考虑这些点对还未确定颜色的点的限制。这样循环做下去，直到不存在新增的确定颜色的

点。设还未确定颜色的点的集合是 S ，我们求出 f_S 表示，只考虑 S 内的点，只能染成 $0,1$ 的方案数即可。

怎么求出 f_S 呢？思路是类似的：取出 S 中任意一个点 x ，枚举它的颜色，然后根据限制连续地确定剩下的一些点的颜色，设最后还剩集合 S' 颜色未确定，用 $f_{S'}$ 更新 f_S 即可。

复杂度 $O(2^n poly(n))$ 。

D - 盼君勿忘

先思考如何计算 $f(s,t)$ 。我们先做一个变换，即令 s' 为一个序列满足 $s'_i = s_i \oplus [i\text{的深度是奇数}]$ ，观察每次操作后 s' 的变化，发现是直接 swap 了 s'_u 和 s'_v 。我们对 s,t 同时做变换之后再来看怎么计算操作次数。不妨以 1 为根，令 $z_i = |\sum\limits_{v\in subtree(i)} s_v - t_v|$ ，则存在方案当且仅当 $z_1 = 0$ ， 且最小操作次数为 $\sum z_i$ 。这是怎么得到的呢？转化一下角度，我们把 $s_i = 1$ 的点看成是点 i 上有白棋，$t_i = 1$ 看成是点 i 上有黑棋，白棋黑棋遇上了可以抵消。虽然题意相当于只能移动白棋，但我们不妨看成：白棋，黑棋都只能向上走，这样不影响答案。于是从下往上地贪心，在当前点能抵消就尽量抵消，再把剩下的点往上移。就可以得到上面的式子了。

于是我们成功的把 $f(s,t)$ 拆成了 z 之和的形式。

现在要计算所有可能的 (a,b) 的 $f(a,b)$ 之和。首先还是对初始的 a,b 先做变换，即对于深度为奇数的点，把 a_i,b_i 异或上 1 ，如果是问号就不管。

现在令 a 中问号个数是 A， b 中问号个数是 B ，已知部分的权值和是 C；

对于一个点 u ，令其子树内 a 问号个数为 x ，b 问号个数为 y ，已知部分权值和是 z ，那可以得到 z_u 对答案的贡献为：

$\sum\limits_{i\le x}\sum\limits_{j\le y}\sum\limits_{k\le A-x}\sum\limits_{l\le B-y}[C+i+k-j-l=0]\binom{x}{i}\binom{y}{j}\binom{A-x}{k}\binom{B-y}{l}|z+i-j|$ 。

虽然看着很吓人，但我们可以发现两件事情：我们只关注 $i-j$ ；我们只关注 $k-l$ 。

枚举 $p=i-j,q=k-l$ 。

注意到 $\sum\limits_i \binom{x}{i}\binom{y}{i-p} = \binom{x+y}{x-p}$ ，$\sum\limits_k \binom{A-x}{k}\binom{B-y}{k-q} = \binom{A+B-x-y}{A-x-q}$ 。

于是上面的式子可以改写成：$\sum\limits_p\sum\limits_q[C+p+q=0]\binom{x+y}{x-p}\binom{A+B-x-y}{A-x-q}|z+p| = \sum\limits_p \binom{x+y}{x-p}\binom{A+B-x-y}{A-x+C+p}|z+p|$ 。

直接枚举即可单次 $O(n)$ 计算，复杂度 $O(n^2)$ ，考虑如何优化。

我们再改写一下式子，令 $p := x - p$ ，即 $\sum\limits_p \binom{x+y}{p}\binom{A+B-x-y}{A+C-p}|z+x-p|$ 。令 $f(x,y) = \sum\limits_i \binom{x}{i}\binom{X-x}{Y-i}|y-i|$ ，其中 $X = A+B, Y = A+C$ 。

接下来我们尝试 $O(1)$ 地从 $f(x,y)$ 推到 $f(x,y+1)$ 和 $f(x+1,y)$ 。

如果能做到这一点，那考虑对树重链剖分后，我们依次对每条重链计算 f 。对于一条重链，可以发现从从链顶算到链尾的过程中 x,y 的变化量是和链顶的子树大小一个级别的，这是 $O(n\log n)$ 的。（事实上直接莫队也不好卡）现在就只需要思考如何做到上述的事情了。

继续简化 $f(x,y)$ ，把绝对值拆了，即 $\sum\limits_i \binom{x}{i}\binom{X-x}{Y-i}(i-y) + 2\sum\limits_{i\le y}\binom{x}{i}\binom{X-x}{Y-i}(y-i)$ 。前面一个式子可以直接 $O(1)$ 计算，我们来看后面的式子。先考虑 $\sum\limits_{i\le y}\binom{x}{i}\binom{X-x}{Y-i}i$ 这个式子，它等于 $x\sum\limits_{i\le y-1}\binom{x-1}{i}\binom{X-x}{Y-1-i}$ 。

现在就只需要研究 $g(x,y) = \sum\limits_{i\le y}\binom{x}{i}\binom{X-x}{Y-i}$ 是如何 $O(1)$ 推到 $g(x,y+1)$ 和 $g(x+1,y)$ 的。

从 $g(x,y)$ 推到 $g(x,y+1)$ 非常容易：加上 $\binom{x}{y+1}\binom{X-x}{Y-(y+1)}$ 即可。再看从 $g(x,y)$ 推到 $g(x+1,y)$ ，这可以从组合意义来推：$g(x,y)$ 相当于从 X 个球中选 Y 个染色，要求前 x 个球中至多 y 个被染了。

于是 $g(x,y) - g(x+1,y)$ 等同于前 x 个球恰好被染了 y 个，且第 $x+1$ 个球被染的方案数。这就是 $\binom{x}{y}\binom{X-1-x}{Y-1-y}$ ！需要注意在一些边界情况下这个组合意义是不合法的，需要特殊计算。

总复杂度 $O(n\log n)$ 。