博弈问题（game）

线段树合并

首先我们来看一看如果只要算一棵子树的答案怎么做。考虑到异或是按位操作，且有两数字某一位不同大小必定优于任意多的更低位不同的贪心性质，可以用一颗 Trie 树表示出这棵子树内的所有数字并在 Trie 上进行 DP。

对于 Trie 上一个点 i，如果它的子树里面总共只表示了一个数字（这里的"一个"指可重集意义下的一个，下同），答案为 -1；如果它的左子树或者右子树中只有一个数字，那么 Alice 只要走进这个只有一个数字的子树 Bob 就必须走另一个，可以对答案有最大的贡献，就计算一下此时的答案（此时 Alice 选的数字确定了，Bob 选择的也随之确定）；否则的话这点的答案就是左右子树的答案中较大的（Alice 选择走进答案较大的子树，Bob 跟进去）。

如果要计算多个子树的答案，只需要利用线段树合并的思想把一个点表示的数字的 Trie 和它的所有子树的 Trie 合并起来，合并的同时根据上面的方法维护答案即可。

兔子猜拳（rabbit）

首先考虑什么样的序列会等价。把兔子的输赢建成树，从赢的向输的连一条有向边。

把树摆在平面上，从左到右依次是一只兔子赢的所有兔子。一只兔子贡献那一维的绝对值是 2^d，如果它多一个儿子，就要乘上一个 -1；如果在往上爬的过程中右边多一个点，就要乘上一个 -1。

两个序列相同的等价条件是两棵树上所有点深度对应相同，乘的 -1 的个数也相同。不妨设 $f(i)$ 表示 i 个点的答案，每次枚举下一层选了 $\boldsymbol{x}$ 个正号和 $\boldsymbol{y}$ 个负号，每次转移一层，但是发现这样根本没办法转移，因为这样的状态是有后效性的，每个点的符号会被自己的儿子影响。

考虑设 $f(i,j)$ 为 i 个点，最后一层有 j 个点目前的符号和未来的符号不同的方案数，但是这些状态不一定互斥，我们需要对于每种情况取出一个特征量。我们发现如果存在两个点一正一负且都有奇数个儿子，交换这两个点方案不变，且强制奇数的数量少了 2。如果所有点都同号，无论如何都不可能达到强制大等更小的值。所以得到我们状态中的 j 只会有一种颜色，不妨设它是负号。那么就有 j 个强制叶子是奇数的点，不妨令它们都只有 1 个，因为偶数可以随便拼。那么就理应存在 $\frac{x+y-j}{2}$ 个负号和 $\frac{x+y+j}{2}$ 个正号。我们至少需要修改 $|\boldsymbol{x}-\frac{x+y+j}{2}|=|\frac{x-y-j}{2}|$ 个符号，而且根据上面的最小表示特性，我们只要修改这么多个。也就是说 $f(i,j)$ 可以转移到 $f(i+\boldsymbol{x}+\boldsymbol{y},|\frac{x-y-j}{2}|)$ 的状态，直接写复杂度是 $\mathcal{O}(n^4)$。

注意到当 $(i+\boldsymbol{x},\boldsymbol{x}-j)$ 相同时，对于所有 $\boldsymbol{y}$ 的转移是一样的。我们可以把 $f(i,j)$ 转移到 $g(i+\boldsymbol{x},\boldsymbol{x}-j)$，再从 g 转移过来，需要注意的一点是，定义域是 $\boldsymbol{x}+\boldsymbol{y}\geq j$ 且 $\boldsymbol{x}+\boldsymbol{y}+j\equiv 0 \pmod 2$，以及 $\boldsymbol{x}+\boldsymbol{y}\neq 0$ 的条件要特殊处理，这样就可以得到一个 $\mathcal{O}(n^3)$ 的做法。

你的世界(world)

48 分解法：$f_{i,j}$ 表示前 i 个数还有 j 个 i_0 没有匹配 j_0，直接 $O(n^2)$ dp 即可。

正解部分：先考虑费用流，建立源 s，汇 t 以及 n 个点 $v_1, \ldots, v_n$，连边 $(s, v_i, 1, a_i)$, $(v_i, v_{i+1}, inf, 0)$, $(v_i, t, 1, b_i)$，则一个流量为 k 的 $s-t$ 流对应一种方案，代价即为费用。

考虑优化费用流，由于流量的限制处理比较困难，而最小费用 $f(x)$ 与流量 x 的关系图像上凸，故对于任意 $k \geq 0$，存在一条与该函数图像相切的直线 $l: y = ax + b$ 经过点 $(k, f(k))$，可以通过二分 a，然后判断 $x = k$ 是否能使 $f(x) - ax$ 取到最大，以及求出其最大值。$f(x) - ax$ 就是把所有 a_i 减去 a 再求最小费用可行流的结果。

二分之后问题转为优化求最小费用可行流，可以用动态加点的方法，假设一开始 $v_1, \ldots, v_n$ 不在图中，接着依次加入，并更新导出子图的最小费用流。加入 v_i 时，导出子图中加入了边 (s, v_i), (v_i, t), (v_{i-1}, v_i)，考虑此时可能产生的负权增广路或环。

- $s \to v_j \to v_{j+1} \to \ldots \to v_i \to t$，即选一个 a 中未被选且 a_j 最小的 $j \leq i$，选取 a_j, b_i。
- $t \to v_j \to v_{j+1} \to \ldots \to v_i \to t$，即选一个 b 中已被选且 b_j 最大的 $j < i$，去掉 b_j，选取 b_i。

所有情况可以分为 (v_i, t) 不流和 (v_i, t) 流一次这两种情况，后者又有上面两种情况，一共三种，取费用最小的一种更新图的最小费用流即可。

该费用流过程可以用优先队列模拟，每次流的复杂度为 $O(n \log n)$，套上二分，总复杂度为 $O(n \log n \log A)$，其中 A 为 a_i, b_i 的最大值。