string

首先，只会交换不同的字符。考虑第一次交换了 $i, i+1$ 和 $i+1, i+2$，则原本 $c_i \neq c_{i+1} = c_{i+2}$，从 011 变成了 110。则可以通过两次取反达成这个效果。因此，不存在一个位置参与了至少两次交换，只会出现 01 变成 10。贪心一下即可。

ball

假设选出的球的 a 从小往大是 $a_1 \sim a_k$，则方案数为 $\prod\limits_{i=1}^{k}(a_i - i + 1)$。证明我们在放第 i 个球时前缀中恰有 $i-1$ 个位置已被占用，方案数为每一步方案数之积。

考虑我们不要求 a 从小往大，则答案并不会变大，计算方案数时只会少考虑情况。因此我们可以给选出的每个 a 任意分配不同的 i（即插入顺序）。

设 $f_{i,S}$ 表示考虑了前 i 个球，已分配的插入位置集合为 S，从 $f_{i-1,S}$ 和 $f_{i-k,S-2^l}$ 转移过来。需要实现一个 m 位的高精度。时间复杂度 $O(nm^2 2^m)$。

tree

一个权值集合能被删空，当且仅当集合大小是偶数，且出现最多的元素出现次数不超过集合大小一半。

先计算集合大小是偶数的方案数，然后再枚举超过一半的元素 c，将这部分贡献减去。可以将 c 当成 1，其他当成 -1，那么要求这个连通块的权值和为正偶数。显然这个值不超过 c 的出现次数 cnt_c。

考虑树形 dp，$f_{i,j}$ 表示 i 子树内权值为 j 的方案数，则 j 只用保留 $-cnt_c \sim cnt_c$ 的部分，时间复杂度 $O(n\sum cnt_c) = O(n^2)$，可以通过。

graph

结论：所有生成树异或和相同等价于**同一点双内的边权值相同**。

证明：首先前者等价于一个简单环上的所有边权值相等。如果两条边不在一个点双，显然不存在简单环同时经过这两条边。如果在同一个点双，则考虑将每条边 (u,v) 拆点为 (u,w) 和 (v,w)，不改变点双性质。问题转化为存在一个简单环经过这两个拆出来的点，这就是点双的性质。因为只考虑点双内不存在割点，因此任意两个点之间的最大流至少为 2。即存在一个简单环包含这两个点。

然后就是经典问题。如果一个点双内有奇数个点，那么对异或和的贡献就是 $[0, \min w]$，否则令方案数乘上 $(\min w) + 1$。考虑枚举最高的有一个数脱离上界的位置，那么上方异或和要和 z 一致，其他数下面可以随便选，这个脱离上界的数恰有一种方案合法。可以写出 dp：$f_{i,0/1,0/1}$ 表示考虑到 i，当前异或和以及是否已经有一个数脱离上界。发现不需要记是否有脱离上界，最后减掉全都不脱离的情况即可。那么 $f_{i,0/1}$ 和加入顺序无关，删除一个数是好做的，需要特判这个数下面的位全是 1 的情况，可能没有逆元。考虑要求 $O(n\log V)$ 次逆元，可以预处理 $1 \sim 2^{25}$ 的逆元。时间复杂度 $O((n+q)\log V(\log V - 25))$