铁甲战士

考虑确定要打哪些以后，再确定打的顺序。

可以发现，优先打成长值高的，一定是最优的。证明考虑每次交换相邻逆序对，一定得到更优解。

因此，先对所有咔咔按成长值从大到小排序，这样打的顺序一定是从前往后。然后考虑使用动态规划求解。

令 $f[i][j]$ 表示前 i 只咔咔里，选了 j 只打死的最优方案。

则转移为 $f[i][j]=\max\{f[i-1][j],f[i-1][j-1]+a_i+(j-1)\times b_i\}$

求解的时间复杂度为 $O(n^2)$。

静默猎手

发现，s_i 与 s_{i+1} 只在第 i 位出现不同，且之后前 i 位不会再改变。

换言之，若 $s_i<s_{i+1}$，则 $s_i<s_k,k=i+2,i+3,\cdots$。$s_i>s_{i+1}$ 同理。

对于 $s_i=s_{i+1}$ 的情况，这些一定是相邻的，因此可以记录下来，等到第一次出现不同的时候一起处理。

从小到大枚举 i，然后维护两个部分：A 部分是字典序排在之后所有串的前面的，B 部分是字典序排在之后所有串的后面的。

每次出现 $s_i\neq s_{i+1}$ 的时候，就可以通过大小关系确定 s_i（以及和它相同的串）要放在 A 部分末尾还是 B 部分开头。

最后将两个部分前后拼接得到最终排列。然后根据式子计算输出的结果即可。

也可以不记录字符串，直接记录每部分的长度、p_i 的和，以及 $p_i\times i$ 的和，可以直接维护出最终答案。

时间复杂度 $O(n)$。

故障机器人

首先，对 a 进行排序。

假设存在一个 $a=0$。

我们要使这些数的异或和为 0，那么从最高位开始考虑。我们每次考虑进行一些操作，去掉当前异或和的最高位的 1。假设最高位为 P。

一个显然的结论是，若 $a_i\le a_j$，则进行若干次操作以后，a_i 与 a_j 之间仍然满足 $a_i\le a_j$。因为若 $a_i>a_j$，则必然在某次操作之后 $a_i=a_j$，此时对 a_j 进行操作显然是等价的。

这意味着，我们可以找一个满足第 P 位为 0 的，且后 $P-1$ 位最大的数，让它恰好加到第 P 位为 1 为止，从而消掉这位的 1。这样操作不会改变相对的大小顺序，且花费了最少的步数，所以这个贪心是优的。

由于存在 0，因此我们每次操作都可以找到这样的数。这样的时间复杂度是 $O(n\log V)$ 的（V 为值域，下同）。

那么现在不存在 0 了，就可能导致一开始找不到这样的数（显然，第一次找到以后，就会一直存在后面的位为 0 的数）。那么，我们需要人为的造一个 0 出来。对第 X 位，我们找两个数，它们的 X 位为 0，且后面 $X-1$ 位尽可能大。然后我们将它们加到 X 位均为 1。这样 X 位的异或和没有被改变，同时我们花费最小的代价得到了两个 0。然后做上面存在 $a=0$ 时的步骤即可。

我们需要从大到小枚举这个 X，然后在这一位上造两个 0（如果有必要），然后再进行上述操作。因此时间复杂度为 $O(n\log^2 A)$。

观察到，我们取出来用来造 0 的两个数的过程，和贪心的时候取出来操作的数的过程是一样的。也就是说，很多数都是没用的，我们没必要每次都枚举 n 个数去找要修改的数。事实上，我们只要对每个 X，选择两个（尽可能找两个）满足第 X 位为 0，后面 $X-1$ 位尽可能大的，**且前面没被选**的数，然后对这些选上的数做贪心的步骤即可（即，只对这 $\log^2 A$ 个数做贪心，而不是所有 n 个数）。

这样的时间复杂度就是 $O(n\log A)$。

观者

对每个结点，我们求出它上方放**最少**的结点时，上方结点最小距离和 l 和最大距离和 r，以及下方结点的最小距离和 L 和最大距离和 R（直接将 Z 加到 l 和 r 中，之后不用考虑 Z）。

同样地，我们求出它上方放**最多**的结点时，上方结点最小距离和 l' 和最大距离和 r'，以及下方结点的最小距离和 L' 和最大距离和 R'。

由于边权为 1，所以我们可以通过调整，使得对于最小值为 x，最大值为 y，区间 $[x,y]$ 中的距离和都能取到。若 $[l,r]$ 与 $[L,R]$ 有交，或者 $[l',r']$ 与 $[L',R']$ 有交，则答案为 0。

否则，由于多选择一个结点，距离和会变大，对于 $r<L$ 且 $r'>L'$ 的情况，即两个区间的大小关系发生了变化，我们可以说明，在绝大部分情况下，两个区间在某一时刻有交。如果两个区间的大小关系自始至终没变化，则选择两个区间距离最近的时候的答案即可。

考虑某一时刻 $r_0<L_0$，且下一时刻 $r_1>L_1$，由于 r 代表最大值，所以此时一定能够选择一个距离为 1 的结点，也就是说 $r_0+1\sim r_1$ 都是可以得到的。同理，$L_1\sim L_0-1$ 也都是可以得到的。那么仅当 $r_0=L_0-1$ 且 $l_1=R_1+1$ 的长度都是 1 时，两个区间没有交（此时答案为 1），其他情况两个区间都有交，答案为 0。这种情况只会出现在上方选 0 个结点或者全选的情况，特判即可。

一个结点上方的结点到它的距离为 1 开始的连续数列，它的上下界可以直接计算。

我们需要维护，一个结点子树内取 k 个数的最大值以及最小值。使用线段树合并维护结点深度，然后用线段树上二分来求前 k 小/大数的和即可。

时间复杂度 $O(n\log n)$。