我是A题

你发现序列的权值就是异或和，导出集合屁用没有。

所以就是求所有子区间的异或和的和。

这个当然是好做的。首先只有 01 你发现很简单，就相当于问有多少个区间有奇数个 1。枚举区间的右端点看有多少合法左端点就行了。

然后不是 01 的话每一位独立单独算就行了。时间复杂度 $\mathcal{O}(n\log W)$。

我是B题

首先你要会判断合法括号序列：维护一个 x，初始为 0，左括号加一，右括号减一，过程中非负且最后和为零则合法。

然后你要会数多少个子序列是合法括号序列，整一个 $f_{i,x}$ 表示在前 i 个位置中选出的子序列，和为 x 的方案数。转移也是简单的。

然后子序列再套上子串也是简单的，相当于你先枚举一个合法子序列，然后前面和后面任选，这部分权值乘上即可。

时间复杂度 $\mathcal{O}(n^2)$。

我是C题

你发现 $(\beta\cdot\gamma)/(\epsilon_n\cdot\gamma)$ 能取到的值就是 (a_i,b_i) 构成凸包然后和一条平行于 y 轴的直线的相交部分。所以你写一个凸包就能秒了 50 分。

当然你说你不想写凸包你可以枚举 (a_i,b_i) 和 (a_j,b_j) 使得他们和直线的交点最高。

然后第二个问题相当于要 (a_i,b_j) 和 (a_k,b_w) 和直线的交点最高。你发现 b_j,b_w 肯定是最大值和次大值，a_i,a_k 肯定有一个是最小值或者最大值。秒了。

我是D题

我们把 $i\to e_i$ 叫做 **false边**，$i\to c_i$ 叫做 **true边**。

到一个点如果条件满足就走 **true边** 否则走 **false边**。

你发现一条 **true边** 如果走了两次那么肯定不会停机。所以你要是能够比较快找到下一条 **true边** 那么就能判断能否停机。

然后我们只连 **false边**，然后你就会发现肯定是若干棵内向基环树，然后想要找到下一条 **true边** 也就是找到第一次满足 $x = a_i$ 是在哪个位置。

那你跳肯定可以分为树的部分和环的部分。

树的部分就是加入你当前在 i，你想要找到最近的祖先 j 满足 $a_j = x + s_i - s_j$ 也就是 $a_j + s_j = x + s_i$，s_i 表示从这里跳到树的根 x 会增加多少。你用可持久化线段树存一下是谁就行了。

然后环的部分先跳到环的第一个位置，这中间如果有满足的就用类似树的方法判一下。否则我们就到了环的第一个位置。

接着我们要转圈圈。不妨假设一圈会增加 cir。如果 $cir = 0$ 转一圈还转不到就肯定会死循环。不妨假设 $cir > 0$。

考虑需要转几圈。记一个 S_i 表示从环的第 1 个点跳到第 i 个点 x 会增加多少。然后如果转若干圈后会在 i 停下，那么满足：$x + S_i \le a_i, a_i - S_i - x \equiv 0 \pmod{cir}$。

首先你可以求出最少几圈会停下，停下的圈数就是所有 $a_i - s_i \equiv x \pmod{cir}, \lfloor \frac{a_i - S_i}{cir} \rfloor \ge \lfloor \frac{x}{cir} \rfloor$ 中最小的 $need = \lfloor \frac{a_i - S_i}{cir} \rfloor - \lfloor \frac{x}{cir} \rfloor$。

你知道要跑多少圈之后找到最近的 i 满足 $a_i - S_i = x + need \times cir$。这个也是好找的。

于是我们可以在 $\mathcal{O}(\log n)$ 的时间内找到下一条 **true边** 或判断无法停机。于是我们得到了一个 $\mathcal{O}(n \log n)$ 的解法。