T1:

最大化 min{a_i}*min{b_i}，那肯定贪心的选择若干个 a_i 最小的矩形和若干个 b_i 最小的矩形删除。

枚举删除几个 a_i 最小的矩形,剩下的删除 b_i 最小的矩形,用排序预处理 min{a_i}，用堆维护 min{b_i}。

T2:

令 f 为给定的 k 个点的 lca，那么 $S=\bigcup_{i=1}^{k} Path(p_i,f)$，所以 $\min\limits_{p\in S}\{|a_p-r|\}=\min\limits_{i=1\sim k}\{\min\limits_{u\in Path(p_i,f)}\{|a_u-r|\}\}$。

所以现在需要维护一个数据结构，支持查询一条链的 $\min\limits_{u\in Chain}\{|a_u-r|\}$。主席树，每次查询不大于 r 的最大值，和不小于 r 的最小值。

T3:

对于任意两个数，找出它们最高的不同二进制位，那么这对数是否对 f(x)产生贡献，由 x 这一位上的数决定。所以每一个二进制位对 f(x)的贡献是独立的，我们直接预处理出来：每次处理一个集合，集合内的数有一个公共前缀，按照最高不同位的 0/1 情况分成两个子集，得到该集合在这一位上的贡献情况，并递归处理两个子集。

二分答案：每次查询有多少个(f(x),x)小于等于当前二元组。

meet in middle，把所有二进制位分为两组，分别排序二元组。每次 two pointers 查询，$(a_1,b_1)+(a_2,b_2)=(a_1+a_2,b_1*2^{\frac{k}{2}}+b_2)=(a,b)$。