省选 2024 模拟赛 Day 1　题解

2024 年 2 月

目录

A 弹性碰撞（physics）

首先注意到一点，由于碰撞后电性会发生反转，所以带正电的只会向左走，带负电的只会向右走。

先考察一种特殊情况: 左边全是负电荷，类型分别是 $f_{1\cdots k}$。最后一个是正电荷，类型为 g。最后一个正电荷会发生碰撞，然后向右走，类型变成 $\underline{g}$（下划线表示翻转）。前面的第 $2\cdots k$ 个负电荷会碰撞两次（向右一次，反向后向左再撞一次），方向和类型最终不改变。第一个负电荷在类型翻转后会被收集。

然后考察一般情况。我们可以认为是第一个正电荷向左走，其它正电荷不动；再是第二个正电荷向左走，其它正电荷不动。以此类推，显然按照这样的逻辑得到的结果是不变的。

假设所有问号都被确定了，那么设正电荷共有 k 个，显然只有前 k 个电荷会被收集。又根据之前的结论，如果初始状态是正电荷，收集时类型一定为 A ，否则一定为 B。

我们依次考虑每个电荷是否会产生贡献。有两个要求，第一个是这个电荷为负或问号，第二个是整个序列中的负电荷数量要大于等于它的下标。预处理组合数后缀和就可以做到 $\mathcal{O}(n)$。

B 连通（connect）

把 a_i, Y 都除掉 X，然后质因数分解一下，把质因数看成元素，每个点权值看成集合。

现在问题是对连通块计数，连通块内的点权的交集为空，并集为全集。

考虑这个等价于：对于每个元素，出现了 0，也出现了 1。

这个等价于：存在一条边，其两端点一个存在这个元素，另一个不存在这个元素。

那就边权为点权的异或，然后作树形 dp，用 FMT 优化转移。

$$f_{u,S} = f_{u,S} + \sum_{T_1 \cup T_2 \cup w = S} f_{u,T_1} f_{v,T_2}$$

但是注意到不需要每次都 FMT，维护 FMT 的点值就可以了。

然后我们还要求出每个 FMT 点值 IFMT 回来的某一项，考虑到 FMT 的本质实际上做高维前缀和，那么在这个点处做高维差分（其实就是子集反演）就可以了。

C 树上二维偏序问题（partial）

首先考虑暴力怎么做。

可以发现一个结论：如果一个 ? 填的是 0，那么它祖先的 ? 也就是 0。证明是显然的。

记 $c_{i,0/1/2}$ 表示结点 i 的祖先中 0/1/? 的数量，$d_{i,0/1/2}$ 表示子树中的数量（均不包含自身）。先钦定所有问号都填 1，此时每个 0 的贡献为 $f_i = d_{i,1} + d_{i,2}$。从根往下依次考虑每个问号是否被修改。对于一个结点 i，考虑它从 1 变成 0 后的变化量，可以得到 $\Delta i = (d_{i,1} + d_{i,2}) - (c_{i,0} + c_{i,2})$。

容易发现一个结点，它的变化量一定不大于祖先的变化量，所以我们直接把所有 $\Delta i > 0$ 的问号结点全部取反就行了。答案为 $\sum\limits_{a_i=0} f_i + \sum\limits_{a_i=?} \max(\Delta i, 0)$。每次修改重新计算 f, Δ 的值可以做到 $\mathcal{O}(nq)$。

接下来考虑链怎么做。

每次修改一个点后，可能会使得它的祖先 f 变化 1，祖先或子树的 Δ 变化 1。这不太好用数据结构维护，考虑询问分块。设立阈值 B，将询问涉及到的至多 B 个点作为关键点。可以发现关键点将链分成了至多 $B+1$ 段，每一段中的变化情况是一样的。

问题在于怎样快速处理块内 f, Δ 加减 1 的操作对答案的影响。修改 f 是简单的，只需要记录块中 $a_i = 0$ 的数量即可。修改 Δ 可以考虑对每一块预处理块内 Δ 整体 $+x$ 后对答案的贡献。具体地，维护一个桶 b_i 表示块内 $\Delta = i$ 的点的数量。那么整体 $+x$ 的贡献就是 $\sum\limits_{i+x\geq 0} b_i(i+x)$。维护 b_i 和 $i \times b_i$ 的后缀和即可得到答案。

注意到 x 是 $\mathcal{O}(B)$ 级别的，所以只需要计算 $\mathcal{O}(B)$ 个后缀和即可。

总时间复杂度 $\mathcal{O}(\frac{nq}{B} + qB)$，取 $B = \sqrt{n}$ 可以做到 $\mathcal{O}(q\sqrt{n})$。

最后考虑一般的情况。

把链上的思想拓展到树上，我们把这些关键点的虚树建出来，那么虚树上至多有 $2B$ 个结点。我们把每对相邻关键点路径上的非关键点形成的连通块压成一块；对每个关键点中，所有不含关键点的子树压成一块，至多有 $4B$ 个块。然后使用类似于链的做法就可以做到 $\mathcal{O}(q\sqrt{n})$。