2024 省选模拟赛

2024 年 2 月 28 日

目录

A 数组（array）

1. 算法一

枚举每一种可能的初始情况并判断即可。

期望得分 10 分。

2. 算法二

当一个数被认定为 0 后，此前与它进行过 1 操作且在与这个数进行操作后没再进行过 1 操作，则必然也是 0。

每次 1 操作，若存在一个数不能被认定为 0 则此次操作的所有数都不能被认为是 0。

那么我们对于每个数记录最后一次 1 操作是什么时候，每次暴力枚举所有点是否可以为 1 即可。

期望得分 30 分。

3. 算法三

离线下所有询问。

我们对每个 1 操作建一个虚点，则每次 1 操作将这个虚点向所有这次操作操作的数上一次操作的点。最后一次操作能走到的所有点都确定为 0 或有能走到当前点的点确定为 0 则当前点为 0。若走到过一个 1，则是 1。其余的为不确定。所以我们进行一次 dfs 可以得到每个点是 1, 0 还是不确定。

然后所有询问都可以简单处理。

期望得分 40 分，拼上算法 2 可以获得 70 分。

4. 算法四

考虑将算法三在线维护。记录一个集合表示当前可以不是 0 且不在 S 里的所有数。每次确定一个为 0 的时候将当前位置所有能走到且没被钦定过的点钦定为 0 即可。

然而还有一种可能，就是一个点进行合并时的所有数都被认定为了 0，此时它也是 0。建反边并删除对应的边即可维护。

因为你只需要保证不在集合 S 内的点都能被认定为 0，所以当 2 操作 $y = 1$ 时实际上是没什么影响的，直接从不确定是否为 0 的集合中删除即可。

期望得分 100 分。

B 虚树（tree）

首先是 $n = m$ 的部分分。考虑贡献分两部分：每种颜色本身的点数和它们之间 LCA 的点数（在不重复的情况下）, 其中前面一部分的和是确定的, 为 n, 我们要最大化后面一部分。

我们考虑在一个点合并它的儿子时, 假设它的儿子子树中分别有 $s_1, s_2, \ldots, s_k$ 种颜色, 我们可以钦定两个子树中某一种颜色为同一种颜色, 一种颜色只能被钦定一次, 那么这种颜色会以当前点为 LCA 贡献 1, 且整棵树的颜色数会少 1, 那么最终贡献数就是 n 减去颜色数, 我们只需要最小化最终合并得到的颜色即可, 容易得到按上述条件合并儿子得到的颜色数为 $\min\left(\max\{s_i\}, \left\lceil\frac{\sum s_i}{2}\right\rceil\right)$, 然后我们再让当前点颜色为一种还没有出现过的颜色, 于是颜色数再加一。我们就这样自底而上在每个节点出做这样的合并, 最后的答案为 $2n$ 减去剩下的颜色数。

当 $n \neq m$ 时, 因为有了颜色数的限制, 我们要对上面的贪心算法进行拓展。

其实主要就是在一棵子树的颜色数超过 m 时要对子树中的颜色进行额外的合并, 为了保证策略的最优性, 需要讨论子树内的颜色种类数是否已经顶到 m。由于有些繁琐, 这里不详细展开, 最好的办法是用拍子调。好处不管什么情况都有唯一的最优决策的。

C 下落的数字（fall）

对于没有修改的部分分, 注意到到达任意一个点的数字都是一段区间, 把每个点对应的区间简单维护出来即可。

这是一个点到点走一条链的问题, 我们考虑跳重链来优化复杂度。

每个点走到它的重儿子的条件都是一段区间, 一个数字能沿一段重链一直走下去的条件是这个数字在这条重链上的所有区间的交中。

对每一条重链建线段树维护区间交, 然后线段树上二分即可找到这个数字走到重链上的那个点, 然后再跳轻儿子即可。

修改也很容易实现。复杂度 $O\left(n \log^2 n\right)$。