# Problem 11~15

2021年10月12日　　21:18

[「2021-09-23 提高模拟赛」围墙 (wall)]

排列对应的图，一定是由若干棵内向基环树组成的。对于树的部分，叶子节点一定是必选为左括号的（因为点一定要有度数）；我们从叶子开始，抽丝剥茧，就能删到只剩环或者只剩一些链。对于二元环或者二元链，当然是让靠左的那个点为左括号更优；剩下的环都是四元以上的，我们直接爆搜，复杂度为 $O\left(2^{\frac{n}{4}}\right)$

[「2021-09-23 提高模拟赛」湖中漫步 (lake)]

这题可以很好地体现平面图的性质。我们处理出可以到达左部的所有在右边界上的点集，再处理出每个左部点，它能到达的上边界和下边界，那么这之间的所有点都是可达的。这个预处理甚至不用缩点。

[「2021-09-23 提高模拟赛」第 k 小可重集合 (multiset)]

第 $k$ 大问题中，有一类经典做法——$\mathrm{kth-element}$；能够在不能够直接找到中间值的题目中起到二分的作用。

在这题中，固定 $l$，$r$ 越大，序列越小。我们随机出一个区间，然后对于每个左端点，找到最小的右端点，使得右端点大于它的所有区间都比这个随机出来的区间来得小。我们不断进行这个过程，显然对于每个左端点，当前合法的右端点一定是一个区间。

考虑我们如何找到这个最小的右端点——二分，然后用可持久化线段树维护前缀中每种数字的数量的一个哈希值。这样就能做到 $O(n\log^3 n)$。

[「2021-09-24 提高模拟赛」开始的陈羽力改变了一切 (social)]

我们从右到左构造这个序列。考虑怎么将某个特定的数移到序列的最末端——我们不断将其置于偶数位进行操作，显然 $n-pos$ 每次至少除以二。总的复杂度也就是 $n\log n$ 的，可以通过最大的部分分。

正解做法就是把这个过程倒过来做，把排列 $1\sim n$ 转化为目标序列，操作也相应地反过来，变成一个倍增的形式，使得 $\frac{n}{pos_i}$ 每次至少除以二。这样做的期望操作次数为：

$$\sum_{i=1}^{n}\frac{1}{i}\sum_{k=1}^{n}\frac{i}{2^k}=n\sum_{k=1}^{n}\frac{n}{2^k}\le 3n$$

剩下的 $2n$ 次操作进行一次随机化即可。

[「2021-09-24 提高模拟赛」随后的林圣涵加大了力度 (np)]

暴力的话，建出子序列自动机，容易发现对答案有影响的很可能只有最后几位，前面的走动非常花时间，考虑怎么优化掉。

一种最朴素的想法是直接倍增，往最小的边走，一路倍增，到不能倍增为止；然后再去子序列自动机跑。

这么做复杂度明显不对劲。考虑将其进一步完善。将这个子序列自动机看作是一棵 $n^2$ 级别的大树。我们对每个点不再是往最小的去倍增，而是往子节点中 $dp$ 值最大的去倍增（如果有很多个大于 $10^{18}$，那么找第一个）。

每次拓展，均向着重链方向去二分；这个复杂度显然是 $O(\log(10^{18}))$ 的；轻边直接在子序列自动机上跳即可。但是这个子序列自动机不能是朴素的子序列自动机了，必须用可持久化线段树来维护，用线段树二分来跳轻边。