UNO 题解

这个题的定位是签到题。

前 30% 数据是给搜索的。后面的数据应该只有 DP of DP 能做，实现不同得分不同。

考虑对于一副手牌，怎么判断它能不能一次打完。考虑把牌按照点数分层，每一层中把存在的点之间连边。最后把这些图合并，得到的图中如果存在的点都是连通的，那么就可行。（这个性质是需要保证可以复制才有的）

所以只需要跑个 DP，即设 $f_{i,sta}$ 表示前 i 种点数牌，当前连通性和颜色的出现与否是 sta 的方案数。转移的时候枚举当前层点的 2^4 种选的状态进行转移。通过搜索可以发现 sta 的大小为 52（其实就是贝尔数的第五个，把和第五个元素分在一起的颜色看作没有出现过）。但其实随便怎么压，只要状态够并且足够小即可，因为这题只有四个颜色。

显然这个 DP 可以矩阵快速幂优化。预处理转移矩阵即可，复杂度为 $O(52^3\times\log n)$，足以通过本题。

神仙验题人提供了另一种做法。一个点数的颜色存在状态只有 2^4 种，所以所有的点数的颜色存在状态并只有 2^{16} 种。枚举存在状态并，问题转化成把 n 个有标号小球放到 m 个有标号盒子里，每个盒子非空的方案数，就是 $\sum_{i=0}^m(-1)^{m-i}i^n\binom{m}{i}$，因为 $m\le 16$，所以可以直接求。得到方案数之后就只需要暴力判断颜色存在状态并是不是连通的即可。复杂度 $O(2^{2^4}+16^2\log n)$，吊打 std。

实际上这个题的颜色数如果开大到 5，那么状压连通性的状态只有 200 多个，而后一种做法显然无法承受 2^{2^5} 的复杂度。且这个题的 n 显然可以开到高精度。但是考虑到这个题是签到题（同时也因为 std 懒），故没有做以上强化。

Treap 题解

这个题的定位是中等数据结构题。

直接暴力模拟可以获得 20 分。

特殊性质不知道有没有什么和正解毫不相干的做法，至少本意是给有分析但是分析不全面或者不会用数据结构维护的同学的。这里就直接分析正解了。

考虑把答案拆开，即一个点的贡献是原树中的 dep 加上合并时新增的 dep'。

原树中的 $\sum dep$ 在 Rotate 时是很好维护的，画个图就可以发现一棵树 $\sum dep$ 的改变就是一些对于 $size$ 的加加减减。

考虑怎么计算 $\sum dep'$。

一个通用技巧是 $\sum dep=\sum size$。在合并的时候，如果当前点是左边的点，那么 $\sum dep'$ 会加上当前右边剩下的树的大小，反之亦然。

而在合并的时候，只有左边的树的右链和右边的树的左链有效，所以可以把他们提取出来。这样问题就变成了有两个不降递减序列 A,B，以及两个指针 i,j，一开始 $i=j=1$。如果 $A_i\ge B_j$，那么 $\sum dep'+=B_j$ 且 $i=i+1$，反之则 $\sum dep'+=A_i$，且 $j=j+1$。

对于 Rotate 操作，最多会修改 A 序列或者 B 序列中的两个值，考虑修改操作对答案的影响。

具体的，我们实际上只需要知道 A,B 序列中的每个值的贡献次数。而 A_i 的贡献次数是 B 序列有多少个数在 $(A_i,A_{i-1}]$ 中，B_j 的贡献次数是 A 序列有多少个数在 $[B_j,B_{j-1})$ 中（因为是 $A_i\ge B_j$，所以这里的区间的开闭关系是这样的）。不难发现一个数的修改造成的影响也是常数级别的，所以用值域线段树维护即可。复杂度 $O(n\log n)$。（其实开到 5×10^5 std 也只要 0.3s，但是不想让选手被卡常……）

Cartesian 题解

这个题的定位是较难的计数题。

可以发现随机打扫的过程就是在随机 DFS。

对于一个点，我们可以看成它把【进入孩子 ch】和【回溯】这几个操作等概率随机排列，并按顺序执行。

先考虑对于给定树，怎么算舒适度的期望。根据期望线性性，考虑计算每个点对舒适度的贡献。一个点有贡献当且仅当它被随机 DFS 遍历到了。而它会被遍历到当且仅当它的祖先的操作排列中，进入这个子树的操作在回溯操作的前面，显然对于某个祖先的概率是 $\frac{1}{2}$。所以一个点 x 对舒适度的贡献是 $\frac{1}{2^{dep_x-1}}\times dep_x$。

再考虑对于所有排列，一个点的贡献是什么。假设我们可以知道 $F_{i,j}$ 表示满足节点 i 深度为 j 的笛卡尔树的个数，那么最终的答案就是 $\sum_i\sum_j F_{i,j}\times\frac{1}{2^{j-1}}\times j$。

考虑 DP 求 F。设 $f_{i,j}$ 表示在前 i 个数中，根到 i 的路径是一条左链，节点 i 的深度为 j 的方案数。那么 $F_{i,j}=f_{i,j}\times 2^{j-1}\times\binom{n}{i}\times(n-i)!$，即先用前 i 个点构造这么一棵树，而因为笛卡尔树是左右区分的，所以乘上 2^{j-1}，然后把剩下的 $(n-i)$ 个点随便插进去。

考虑 $f_{i,j}$ 的转移，枚举它的双亲，我们有 $f_{i,j}=\sum_{k=1}^{i-1}f_{k,j-1}\times\binom{i-2}{i-k-1}\times(i-k-1)!$。即如果 k 是 i 的双亲，那么在区间 $[k+1,i-1]$ 的数不能放在 i 和 k 之间，但是可以和前面 $[1,k-1]$ 内的数随便排。边界条件是 $f_{1,1}=1$。这样就可以做到 $O(n^3)$。

明眼人都可以看出来这个东西随便前缀和优化，就可以做到 $O(n^2)$ 了。

考虑整理一下求和式，我们有 $\sum_i\sum_j f_{i,j}\times\binom{n}{i}\times(n-i)!\times j$。

如果头铁的话说不定可以用生成函数求，因为有个姐妹题就是用生成函数做的，在此就不展开了。欢迎用生成函数碾 std。

可以发现瓶颈是求 f。

如果你和神仙验题人一样强是可以继续做的。考虑设 $g_{i,j}=f_{i,j}\times j$。那么在转移的时候 f,g 是可以通过前缀和互相转移的。且在转移中没有出现 $\times j$。所以可以把 j 这一维压掉。

但是出题人还是想从另一个方面来考虑这个题目。

对于一个节点 x，考虑把其深度的贡献拆开，即 $1+\sum_y[y\text{ 是 }x\text{ 的祖先}]$。

先考虑计算 $\sum_i\sum_j f_{i,j}\times\binom{n}{i}\times(n-i)!$，对于某个 i，其实际意义是满足根到 i 的路径为左链的笛卡尔树方案数。显然只需要满足所有小于 i 的数在 i 的右边即可，所以方案数是 $\binom{n}{i}(i-1)!(n-i)!$。对所有 i 求和就是 $\sum_i\binom{n}{i}(i-1)!(n-i)!$。整理一下是 $n!\sum_i\frac{1}{i}$。

考虑计算后半部分。设 $g_{i,y}$ 为满足根到 i 的路径为左链且 y 是 i 的祖先的笛卡尔树的个数。那么后半部分的贡献是 $\sum_i\sum_y^{i-1}g_{i,y}$。考虑 $g_{i,y}$ 怎么求。首先 i 在 y 的左边，其次 $[1,y-1]$ 的数不能在 $[i,y]$ 之间，不然 y 就不是 i 的祖先，又因为根到 i 的路径是左链，所以 $[1,y-1]$ 的数也不能放在 i 的左边，所以 $[1,y-1]$ 的数只能放在 y 的右边。而 $[y+1,i-1]$ 的需要放在 i 的右边，$[i+1,n]$ 的数任意。

所以先从 n 个位置中选择 i 个填 $[1,i]$，点 i 需要放在这 i 个位置的最左边，所以在剩下的 $i-1$ 个位置中选择 y 个放 $[1,y]$，点 y 放在需要放在这 y 个位置的最左边。所以 $g_{i,y}=\binom{n}{i}\binom{i-1}{y}(n-i)!(i-y-1)!(y-1)!$，整理就是 $\frac{n!}{i\times y}$。

所以原式就是 $n!\sum_i\frac{1}{i}\sum_y^{i-1}\frac{1}{y}$，后面的 $\sum$ 直接前缀和即可。

故答案为 $n!\sum_i\frac{1}{i}(1+\sum_y^{i-1}\frac{1}{y})$。

显然通篇只涉及了逆元，线性求逆元做就完了。复杂度 $O(n)$。但是这样需要保证模数为质数。

可以发现 $n!$ 和里面的东西乘起来之后就没有了求逆元操作。直接处理即可。