2022 杭电多校第十场 题解

Problem A Winner Prediction

先让 1 号选手赢下所有和他有关的比赛，设此时选手 i 赢了 a_i 场比赛。如果存在某个 $a_i > a_1$ 则 1 号选手不可能成为冠军。否则选手 i 至多还能再赢 $b_i = a_1 - a_i$ 场比赛。考虑建立一张网络流图：每场未进行的比赛在图中用一个点表示，源点向它连容量为 1 的边，它向它的两个参赛选手的对应点各自连容量为 1 的边；选手 i 的对应点向汇点连容量为 b_i 的边。计算该图最大流，若源点出发的边满流则答案为 `YES`，否则为 `NO`。

Problem B Photos

简单起见，我们把左下的漂亮格子沿对角线翻转到右上去，显见不影响答案。观察到以下事实：

- 如果格点 (i,i) 被选作某个正方形的顶点，那么将不可能覆盖到格点 (i,i) 右上的所有格子；
- 如果格点 $(i-1,i-1)$ 和 (i,i) 不在同一个正方形里，那么将不可能覆盖到格子 (i,i) 以及其右上的所有格子。
- 不能覆盖的全体格子恰好是以上两类所得出的不能覆盖的格子的并集。

其中"格子"对应原文 `cell`，"格点"对应原文 `grid point`。

称某个（主对角线上的）格点是可用的，当且仅当其右上没有漂亮格子。设 S 是可用格点的集合。对 $(i,i) \in S$，记 s_i 为在只考虑前 i 行前 i 列的情况下，用若干正方形覆盖其中全部漂亮格的方案数。转移有以下两种形式：

$$s_j \to s_i (j < i, j \in S)$$

（用了左上为 (j,j)，右下为 (i,i) 的正方形。）

$$s_{i-1} \to s_i$$

（(i,i) 不作为某个正方形右下角。要求格子 (i,i) 右上没有其他格子。）

再记录 t_i 表示 s_i 的前缀和。直接转移是 $O(n)$ 的，不能承受。注意到事实上所有转移分为 $O(m)$ 段，每段内 (s_{i-1}, t_{i-1}) 与 (s_i, t_i) 的转移可以写成矩阵乘法的形式，这样即可做到 $O(m \log n)$。

Problem C Wavy Trees

先确定是奇数位为大数还是偶数位为大数。接着从左往右贪心，对每个 $i = 2, \ldots, n$，如果 a_i, a_{i-1} 大小关系不符合条件，则调整 a_i。容易证明这是对的。时间复杂度 $O(n)$。

Problem D Average Replacement

结论一 每个连通块内的数最后会趋于同一个值。

证明 参见 CMO 2021 第二试第一题。

结论二 记 $\deg(i)$ 表示 i 的朋友数量，则每个连通块内 $\sum a_i(\deg(i)+1)$ 为一定值。

证明 显然。

综合以上两点即可解出最后收敛到的那个值。时间复杂度 $O(n)$。

后记 本题略为卡常，在此我们表示歉意。以及有一定程度上的精度问题，这是 HDOJ 系统经常遇到的问题，我们已经尽力避免了，如果还是干扰了选手做题我们感到抱歉。

Problem E Apples

考虑枚举 $(1,n)$ 之间传输了多少苹果，设为 x。则接下来每两人之间传输的苹果数是固定的，可知答案为 $\sum_{i=1}^{n} l_i|x-a_i|$。容易知道这个式子的最小值在 x 取 $a_1,\ldots,a_n$ 的加权中位数时取到。将 a 先离散化、排序，用序列数据结构维护 l 的前缀和即可。时间复杂度 $O(n\log n)$。

Problem F Triangle Rotation

定义 一个排列是奇排列当且仅当其逆序对数为奇数；反之亦然。

容易发现，旋转操作不改变排列奇偶性。对于偶排列，我们可以采用从下往上逐行复原的方法，复原一个元素只需要 $O(n)$ 步。总步数在 $O(n^3)$ 级别，常数限制比较松。注意实现细节。

Problem G Even Tree Split

对每条边，如果删去该边后两棵树点数都为奇数，则称其为好边。显见一个边集是合法的当且仅当边集内都为好边。则答案为 2^c-1，其中 c 为好边数量。时间复杂度 $O(n)$。

Problem H Minimum Diameter

先考虑对于某个特定的森林怎么求答案。显见只有每个连通块的直径长度需要考虑。一种最优的连接方法显然是以直径最长的那个连通块的直径中点为中心，把其他连通块直径中点都连上去。每次如果都这样做的话是 $O(n)$ 的。

注意到如果最大直径为 d，那么我们只需要知道直径为 $d,d-1,d-2$ 的连通块数量就能求答案了。因而考虑先将所有边离线下来，这样合并两个连通块时就能快速查询直径了，再开一个数组存储每个直径长度的连通块数量。需要一些仔细的分类讨论。时间复杂度 $O(n\log n)$。

Problem I Painting Game

我们把涂黑操作想象成剪掉连续的三个格子。如果每个连续段长度都 ≤ 2，那么结果事实上已经决定了。在此之前，可以发现，Alice 的一种最优策略是：选某个连续段的左数第二个格子涂黑；Bob 的一种最优策略是：选某个连续段的左数第三个格子涂黑。设 $f(n),g(n)$ 分别为 Alice、Bob 面对长度为 n 的空纸带时的答案。则有 $f(n)=g(n-3)+1,g(n)=f(n-4)+2$ （$n\ge 7$）。注意到 $f(n)=f(n-7)+3,g(n)=g(n-7)+3$，因而可以快速计算答案。时间复杂度 $O(1)$。

Problem J Tree

矩阵树定理。每加一条边 (x,y)，那么 $A_{x,y}-1,A_{y,y}+1$。

如果矩阵 A 满秩，求 A 的余子式只需要求行列式和逆元。

但显然基尔霍夫矩阵 A 的秩 $r(A)$ 至多是 $n-1$。

如果秩不为 $n-1$，那么所有代数余子式均为 0。下面仅考虑秩为 $n-1$ 的情况：

记 A^* 为伴随矩阵，显然有 $A^*A=|A|I=0$，那么 $r(A^*)+r(A)\le n$，即 $r(A^*)\le 1$。而 A^* 不为零矩阵，因此 $r(A^*)=1$。

既然秩为 $n-1$，那么存在不全为 0 的实数 $x_1,x_2,\cdots,x_n$ ，使得 $\sum x_i\overrightarrow{a_i}=0$，其中 $\overrightarrow{a_i}$ 为矩阵第 i 行的形成的行向量。

假设现在有了某个余子式 $A_{i,j}$，那么对于同一列上的某个 $A_{k,j}$，可以证明：$A_{k,j}=\frac{x_k}{x_i}A_{i,j}$。

同理有不全为 0 的实数 $y_1,y_2,\cdots,y_n$ 使得 $\sum y_i\overrightarrow{a_i'}=0$，其中 $\overrightarrow{a_i'}$ 为列向量。$A_{i,k}=\frac{y_k}{y_j}A_{i,j}$。

那么有：$A_{a,b}=A_{i,j}\times\frac{x_ay_b}{x_iy_j}$。

解出一组 x, y 后，随便找到一个不为零的 x_i 和一个不为零的 y_j，就可以计算出所有的代数余子式了。

时间复杂度 $O(n^3)$。

Problem K Maximum Triangles

我们考虑对于一个包含原点的三角形，其任意一个顶点，另外两个顶点都被这个顶点与原点的连线分在两侧；而对于一个不包含原点的三角形，一定存在恰好两个顶点，使得另外两个顶点都被这个顶点与原点的连线分在同侧。

所以我们想到一种统计包含原点的三角形的方法，即假设第 i 个点与原点的连线两侧的点数分别为 l_i, r_i ,则包含原点的三角形数为 $\binom{n}{3} - \frac{\sum_{i=1}^{n} \binom{l_i}{2} + \binom{r_i}{2}}{2}$ ，三角形个数最大时需要对于每个点 l_i 与 r_i 的差尽量小，即为 $\forall i \in [1, n], l_i = \lfloor \frac{n-1}{2} \rfloor, r_i = \lceil \frac{n-1}{2} \rceil$ 。

std的构造方式为选择所有 $x = \pm 1$ 和 $y = \pm 1$ 的点，关于原点对称的两点仅选一个，具体构造方式可以参考std。

Problem L Expected Inversions

根据期望的线性性，我们可以将逆序对数的期望拆解为：对所有的 $x < y$，$d_x > d_y$ 的概率之和。

我们考虑对给定的根节点 u，一对 $x < y$ 的 DFS 序将带来怎样的贡献。可以发现：

- 如果 x 是 y 的祖先，那么一定有 $d_x < d_y$，贡献为 0；
- 如果 y 是 x 的祖先，那么一定有 $d_x > d_y$，贡献为 1；
- 否则记 $l = \mathrm{LCA}(x, y)$，那么 d_x 和 d_y 哪个大取决于在 l 处 DFS 时，先进入 x 的子树还是先进入 y 的子树，这两种情况显然概率是相等的，因此贡献为 $\frac{1}{2}$。

如果对每个 u 都枚举 x, y，时间复杂度 $O(n^3)$。一个更好的做法是：先默认每对点贡献都为 $\frac{1}{2}$，然后枚举 (x, y)，计算其带来的贡献影响即可。可以发现这样的贡献作用于两棵子树，用 DFS 序维护即可。时间复杂度 $O(n^2)$，

考虑进一步优化。对于每个 x, y，记 $T(x, y)$ 表示 x 和 x 所有不包含 y 的子树的点构成的集合。不妨设 $x < y$，相当于我们要给 $T(x, y)$ 的全部点减去 $\frac{1}{2}$，同时给 $T(y, x)$ 的全部点加上 $\frac{1}{2}$。我们记前者是**在 x 处发生的**，后者是**在 y 处发生的**。

对于每个点 x，我们来计算所有在 x 处发生的贡献。不难看出，在 x 的同一个子树内的点受到相同的贡献。进一步地，我们有如下计算贡献的方法：

- 对于 x 的每棵子树 T'，记 T' 中比 x 小的点有 a 个，比 x 大的点有 b 个，那么所有不在 T' 内的点受到 $\frac{a-b}{2}$ 的贡献。

由于树只有 $O(n)$ 条边，上述计算的总次数是 $O(n)$ 的。我们来优化单次查询的复杂度：我们要查询某棵子树内小于 x 的数有多少个（大于该值的数量可以通过总体减去小于的数量求得）。这是简单的：我们从小到大加入点，维护 DFS 序上的区间和即可。这一步可以用树状数组做到单次 $O(\log n)$。贡献的计算也是关于子树的，打差分标记即可。

总时间复杂度 $O(n \log n)$，可以通过此题。

后记 本题赛时数据有问题，在此我们向所有受到影响的选手真诚道歉。出题人的标准程序在清空时只清空了数组的 $0 \sim n$ 位，但是使用的时候用的是 $1 \sim (n+1)$ 位，导致数值没有被完全清空，进而导致答案错误。出题人已经进行了深刻反省，这里对其马虎大意的行为再提出一次批评，希望以此警示其他的选手、出题人。另外本题略为卡常，这是因为要区分掉启发式合并的 $O(n \log^2 n)$ 做法，如果有选手受到影响我们一并道歉。