Problem A : [CF1375E] Inversion SwapSort

不妨设 a 是一个排列。注意到交换原排列的两个位置等同于交换逆排列的两个值，对逆排列冒泡排序即可。

Problem B : [CF1391E] Pairs of Pairs

建出 DFS 树。如果有一个点的深度至少为 $\lceil \frac{n}{2} \rceil - 1$，则找到了路径。否则，

1. 设 $x = \lceil \frac{n}{4} \rceil$ (即点对的数量)，删去 DFS 序第 $n - x + 1$ 个点到根的路径 (不包括这个点)，这些删去的点都是 DFS 前 $n - x$ 个点，且删完之后剩的点数至少为 x，把这些点和 DFS 序后 x 个点一一配对即可 (两部分的点之间不存在边)。
2. 匹配同深度的点，则较深的点对中每个点至多向上连一条边。

Problem C : [CF1270G] Subset with Zero Sum

注意到 $1 \le i - a_i \le n$，从 i 连向 $i - a_i$ 得到一个基环树森林，其中任意一个环满足 $\sum i = \sum (i - a_i)$，即 $\sum a_i = 0$。

Problem D : [CF1383D] Rearrange

先考虑所有行列的最大值。将他们从大到小排序，对每一个数，如果它是行最大值，则需要放在新的一行；如果是列最大值，则需要放在新的一列。当加上其他数之后，我们仍然从大到小处理所有数，遇到非行或列的最大值的数时，只需要在满足上下分别是阶梯形的条件下任意摆放即可。

也可以先摆放所有行列最大值，再按照新建行列的顺序摆放其他数。

Problem E : [CF1379E] Inverse Genealogy

显然 n 必须是奇数，k 至多为 $\max\{0, \frac{n-3}{2}\}$。事实上，对于给定的奇数 n，在所有满足大小条件的 k 中，只有常数个 k 无解，因此枚举一棵子树大小和不平衡的人数递归构造即可，容易发现一个结点枚举的总数不超过较小子树的大小的常数倍，因此复杂度为 $O(n \log n)$。

Problem F : [CF1444D] Rectangular Polyline

有解当且仅当 $h = v$ 且两个集合分别能分成两个和相等的集合。假设分成了 X^+, X^-, Y^+, Y^-，不妨设 $|X^+| \le |Y^+|$，则分成三段，分别是 X^+Y^+, X^-Y^+ 和 X^-Y^-。以 X^+Y^+ 为例，将 X^+ 从大到小、Y^- 从小到大排序，就能保证得到的图形在 $(0,0)$ 和 $(\mathrm{sum}\,X^+, \mathrm{sum}\,Y^+)$ 的连线之下。

Problem G : [CF1329D] Dreamoon Likes Strings

依次写下原串中所有相邻相同的字符得到一个新串，则原串中的操作对应到新串要么是删除一个字符，要么是删除两个相邻且不同的字符 (其余操作显然不优)。新串删空后再用一步操作就能删空原串。删空新串的次数下界为 $\max\{\lceil \frac{n}{2} \rceil, s\}$，其中 n 是新串的长度，s 是新串中出现次数最多的字符的出现次数。这个下界容易达到，每次删出现最多的字符和相邻的一个字符即可。

Problem H : [CF1396E] Distance Matching

考虑每条边的经过次数，设其两边的子树大小分别为 $a, b\,(a + b = n)$，则其经过次数至少为 $a \bmod 2$ (其奇偶性当然与 a 相同)，至多为 $\min\{a, b\}$。对每条边求和，得到最小值为 L，最大值为 R，则一个必要条件是 k 在 L 到 R 之间且奇偶性与它们相同。

事实上，这也是充分条件。考虑进行一个转化：将原树的每个点连出一个叶子，形成一棵 $2n$ 个点的新树，接下来我们为每条边确定经过次数，再根据这些经过次数为每个点相连的边配对，得到的就是一个合法匹配。

一个点相连的所有边能完美匹配的充要条件显然是经过次数的最大值不超过总和的一半 (奇偶性在考虑每条边时已经满足)。因此问题变成了给定 k，构造一组满足要求的经过次数。

首先令每条边的经过次数是其上界，这显然满足条件。接下来每次让经过次数最大的边减少 2，仍然满足条件。这个过程可以用二分实现。

得到经过次数之后 DFS 即可完成构造。

Problem I : [CF1237H] Balanced Reversals

有解当且仅当 a 和 b 中 00 和 11 的数目分别相等。考虑依次对 $i = 2, 4, \ldots, n$ 构造 $a[1..i] = b[n - i + 1..n]$。假设我们要在前面添加 xy，则只需找到 $a_{j+1}a_{j+2} = yx$ 然后依次翻转长度为 $j, j + 2$ 的前缀即可。这种构造只在 a 中 01 和 b 中 10 的数目相等时有效。对于一般的情况，记 $\mathrm{balance}(s)$ 为 s 中 01 的数目减去 10 的数目。不妨设 $|\mathrm{balance}(a)| \ge |\mathrm{balence}(b)|$，则要找到 a 的一个前缀 p 满足 $\mathrm{balance}(p) = \frac{\mathrm{balance(a)+balance(b)}}{2}$ 并将其翻转。这是一定可以做到的，因为考虑 a 的长度

为 $0, 2, \ldots, n$ 的前缀，相邻两个之间 balance 至多改变 1，而 $\frac{\text{balance(a)}-\text{balance(b)}}{2}$ 在 0 和 $\text{balance}(a)$ 之间。