A

仅讨论 $k=4$ 的情况。

把每个人的参赛情况用一个 $[0,16)$ 中的整数 s 表示，再按照 $\text{popcount}(s)$ 从大到小贪心凑即可。具体地，可以依次按照如下的 popcount 组合凑：

$$\{4\},\{3,1\},\{3\},\{2,2\},\{2,1,1\},\{2,1\},\{2\},\{1,1,1,1\},\{1,1,1\},\{1,1\},\{1\}$$

时间复杂度 $\mathcal{O}(\sum nk)$。

B

枚举那个不受「度数 $\le k$」限制的点，发现我们需要求出：

1. 以 u 为根的子树中，选择一个包含 u 的连通块，使得每个点度数 $\le k$（特别地，u 的度数要 $< k$）的最大边权和；
2. 整棵树去掉以 u 为根的子树后，选择一个包含 u 的父亲 v 的连通块，使得每个点度数 $\le k$（特别地，v 的度数要 $< k$）的最大边权和。

可以使用换根 dp + 贪心求出，时间复杂度 $\mathcal{O}(n\log n)$。

C

考虑调整。注意到可以把 $(r,c_1),(r,c_2)$ 调整为 $(r,c_1+c_2-1),(1,r)$，所以我们可以证明对于每一组可行的 (s,c)，存在一组解满足除 $(1,1)$ 外出现次数都 ≤ 1。

而 $(1,1)$ 的作用是 $s\leftarrow s+1,c\leftarrow c+2$，所以我们只要求出 $2s-c$ 为定值时合法的 c 的最小值即可。枚举 r，不难在 $\mathcal{O}(s)$ 的复杂度内更新所有 (r,c) 的贡献。

时间复杂度 $\mathcal{O}(s\sqrt{s})$。

D

考虑从一个严格递增序列开始操作的过程，发现操作过程中每个数右侧小于它的数的个数始终为偶数，并且显然任意满足这个条件的序列都合法，所以这个条件是充要的。

考虑用分块维护这个信息。设 i 右侧有 r_i 个 $< p_i$ 的数。假设交换 p_x,p_y，那么 x,y 所在的块可以暴力重构；对于中间的块，影响是一个值域区间内的 $r\pm 1$。

于是可以想到每个块内部先排序，然后维护 r 的差分。这样的话发现需要对中间每个块做 `lower_bound`，而块内是可以线性重构的。直接做时间复杂度为 $\mathcal{O}(n\sqrt{n\log n})$，已经足以通过。

我们考虑去掉这个 `lower_bound` 的过程。发现修改和 `lower_bound` 其实就是单点加、前缀查。使用 $\mathcal{O}(n)-\mathcal{O}(1)$ 的分块即可做到 $\mathcal{O}(n\sqrt{n})$。但是这个做法需要 $\mathcal{O}(n\sqrt{n})$ 的空间，常数也不是很优秀，简单来说就是出题人在开 std 两倍时限的情况下卡不掉上面的做法。

也存在使用时间分块的 $\mathcal{O}(n\sqrt{n})$ 做法，出题人没有实现过所以不知道常数表现如何。