NOIP 模拟赛 1

矩形印章 (seal)

暴力：枚举每个可能的位置是否需要盖印章，使用状压枚举或搜索都行，搜索加剪枝写得优秀点可以通过。

正解：暴力贪心判断，每次找到图纸最上面最左边的那个需要被染黑的，把印章的最上面最左边的和他对上，暴力 check 一遍有没有印到图纸外面、留白的地方或之前印过的地方。

铁路距离 (railroad)

暴力：考虑 $\operatorname{joy}(x, y)$ 如何计算，可以二分，然后 bfs 校验，时间复杂度大概是 $O(n^4 \log)$ 的；

优化后的暴力：类似于全源最短路，使用 Floyd 优化，时间复杂度 $O(n^3)$。

正解：至多只需要经过直径的两端点就能满足所有点对的要求，计算每个点距离直径两端点距离的较大值 d_i，那么 $\operatorname{joy}(x, y) = \min(d_x, d_y)$，排序计算贡献即可。

最优方案 (plan)

这道题的难度其实是 T4 的难度。

分析性质，当 $a_{l\sim r}$ 均为 0 时，答案显然为 `0 0`，接下来不考虑这种情况，设 $w=\max_{i=l}^{r}\{a_i\}$，那么第一问的答案一定是 $2^{\lfloor\log_2 w\rfloor+1}-1$，且第二问的答案不超过 2。只需要对于 w 先操作 $i=2^{\lfloor\log_2 w\rfloor}$ 变成 $2^{\lfloor\log_2 w\rfloor}$，再操作 $i=0$ 变成 $2^{\lfloor\log_2 w\rfloor+1}-1$。

那么，现在考虑，只需要能够判断第二问答案是否为 $0,1$，就能解决该问题了。

第二问答案是否为 0 是好做的，只需要求出区间或是否是第一问的答案即可。

第二问答案是否为 1 不太好做，发现区间内的所有 a_i 可以分为两种：第一种是怎么修改都不影响区间或的；第二种是可能会影响区间或的。

可以发现，第二种的数至多不超过 $\log_2 a_i$ 个（二进制位中有某个 1 是区间独有的），而第一种的所有数，需要进行一些性质上的分析：

- 对于一个数操作一下后，其二进制表示会变成中间一段连续 1 的形式（且末尾位置恰好是操作时选择的那个二进制位，且另一端点是 b_j 从 i 开始往高位找到第一个 1 的那个位置）；
- 那么记录 $f_{0\sim 29}$，f_i 表示最低位的 1 从 i 开始的最大值，那么同样只需要记录 $\log_2 a_i$ 个值。

所以考虑分治，每次处理跨过分治中心的所有区间，左右两边分别求出可能的第二种的所有数，以及第一种的贡献 $f_{0\sim 29}$（考虑插入一个数会有哪些影响，可以 $O(\log a)$ 修改），再暴力检验：对于第一种数的修改直接枚举判断，对于第二种数，需要枚举其中一个数再枚举操作哪一位，总时间复杂度是 $O((n+m)\log^2 a)$，理论上是通过不了的，但可以获得几乎满分了。

最后正解需要进一步地分析性质：

- 考虑找到初始区间的或的最低位的一个 0，设在第 k 位，将一个数变成一个二进制下区间的一段 1 之后，由于末尾 1 一定是操作时选择的那一个 1，那么显然，我们会贪心地选择第 k 位操作（不仅是第一类数，第二类数同样的）；
- 考虑第二类数操作第 k 位的结果，容易发现和原来相比，只会有恰好一个原来的 1 被保留到最后了（虽然这个性质似乎没有什么用）；
- 所以，我们只需要找到满足其二进制下从第 k 位开始往高位的第一个 1 至少是第 q 位（q 是可以通过初始区间或求出来的），可以考虑对于所有询问的 k 离线下来，然后从左到右扫描右端点 r，找到前 $\log_2 a+1$ 的检验一遍（因为第二类数至多 $\log_2 a$ 个），可以使用简单的桶、指针/链表维护。

总时间复杂度：$O((n+m)\log a)$。空间复杂度可以做到线性。

树边定向 (tree)

首先肯定先转化为对所有边定向的问题，无需变成排列。

可以先设计一个 dp：设 $f_{u,x,y}$ 表示 u 子树内，有 x 个点能够走到 u，u 能走到 y 个点，直接 dp，复杂度为 $O(n^5)$，不太知道得几分，没写过。或许可以进行一些优化降低复杂度。

接着，我们需要讲一个重要的性质：任取树上一条路径，其边的方向，至多只会变化一次。如下图

英文看不懂？我真不想翻译一遍了，听我讲吧……

这样有了这个，就能得到，最终方案一定是找到一个点作为根，然后根的所有子树要么全向上，要么全向下。

然后，就要让向上的子树大小之和和向下的子树大小之和尽可能接近，可以用背包解决，时间复杂度：$O(n^3)$。如果使用 bitset 优化的话，可以做到 $O(\frac{n^3}{\omega})$

然后继续分析性质，发现这个根一定是选取树的重心，时间复杂度优化到 $O(\frac{n^2}{\omega})$。

> *其实到这一步，善于~~观察~~打表的同学基本上都是能够发现的。*

然后，你会发现，你需要完成一个物品体积为 $a_1, a_2, \cdots, a_k$，求出最终能够凑出哪些体积。

由于 $\sum a_i = n - 1$，所以你会发现，可能会有很多相同的 a_i，你把他们放在一块做一个二进制拆分，就能够通过了（这个复杂度是可以证明的，是 $O(\frac{n\sqrt{n}}{\omega})$ 的）。

另外，你也可以使用一种方法，就是如果发现相同的 a_i 超过两个了，那就完全可以把其中两个合并成一个 $2a_i$，直到不能进行操作为止，每个 a_i 至多出现两次，这样可以证明剩下的 a_i 只有 $O(\sqrt{n})$ 个，总时间复杂度是 $O(\frac{n\sqrt{n}}{\omega})$。