

T1 数数题

[PA 2020] Malowanie płotu

暴力 DP 是简单的，我们记 $f_{i,l,r}$ 表示第 i 个区间是多少，转移如何做到平方呢？考虑正难则反，与 $[l,r]$ 不相交 $[l',r']$ 的要么 $r' < l$ ，要么 $l' > r$ ，前缀和一下就行了。

如何进一步优化状态？注意到我们实际上只关心 l,r 中的一个就可以转移了，所以我们可以记 $s_{i,l}$ 表示 $\sum_{r=l}^n f_{i,l,r}$ ，由于对称性这同时也是 $\sum_{l=1}^r f_{i,l,r}$ 的翻转。然后还需要记录 s 前缀和，转移基本同上。

T2 造树题

[CF773D] Perishable Roads

考虑生成树形态，由于是计算路径 **min** 之和，所以只需要关注瓶颈就行了。可以发现生成树一定是长成末尾是根，开头挂着一个菊花的一条链，链上的边权递增。

考虑费用提前计算，树的代价和就是链上所有边权的长度加上菊花的大小乘最小边权，即 $d_i + mn(n-1-x)$ ， x 为链长。所以我们把所有边权同时减去一个 mn ，然后在新图上求最短路就行了。

用 dijkstra 做到 $O(n^2)$ ，当然如果你习惯堆优化会多带一个 \log ，没有卡这个东西。

T3 博弈题

IOI2023 集训队互测 九王唱

发现这个博弈形态太奇怪！如何寻找切入点？

虽然我们不确定谁会留下来，但是我们知道有些数一定会死：某个人最不喜欢的数一定不会留下来，否则轮到他如果剩下了这个数他一定能毙了这个数。

把这个过程类推下去，我们可以发现这个题实际上的博弈结果就等于：从 n 到 1 每个人都拿走他最不喜欢的数，最后剩下的数。

直接模拟是 $O(n^3)$ ，但是可以注意到这个过程有决策单调性（不是 DP 中的那种，但是也可以这么说），一个人拿走的数在开始点往后的时候排名只会增加，所以拿个指针维护一下，当开始点穿过这个数时重新暴力扫描即可。

T4 直线题

[EC Final 2021] Vison Test

突破口在于考虑固定直线的斜率 K ，那么截距 B 就需要满足不等式 $\max(a_i - Ki) \leq B < \min(a_i - Ki) + 1$ 。当 $a_i - Ki$ 数组的极差要严格小于 1 时，取 $B = \max(a_i - Ki)$ 就是字典序最小的解了。

接下来考虑什么样的 K 能让极差小于 1 。一个很牛的想法是注意到在 K 增大的时候，最小值的位置总是在往右边移动，最大值的位置总是往左边移动；而当我们想要减小极差时，如果最小值在最大值左边，那么只有增大 K 才能减小这个极差。这说明这个 K 是有可二分性的，在 Stern-Brocot 树上二分即可。SBT 二分需要倍增优化跳直链，根据一些经典结论 SBT 倍增 check 的总次数是在 $O(\log V)$ 级别的。

现在考虑询问区间 $a_i - Ki$ 的极值，这是区间凸包问题。不过我们不能带很多 \log ，所以不能直接上线段树做 $O(n \log n) - O(m \log^2 n)$ 的做法。

一个十分好写的做法是，考虑用回滚莫队维护凸包，由于斜率已经有序所以直接正常拿栈维护，平常怎么建凸包就怎么建。这里的回滚莫队当然不需要回滚，其实就是分块，将左端点在同一个块内的一起处理，每次暴力建出左端点到块右边界的凸包，然后排序右端点增量构建出右边界到右端点的凸包，这样询问就只有一个 \log 了。

当然另一个 polylog 做法是，类似猫树一样优化那个线段树的区间凸包做法，由于斜率有序所以直接弄出凸包栈的操作树在上面倍增询问即可。