

A:开开心心233

签到题，无论二分还是解方程还是直接for循环枚举都能直接通过啦

B站关注 开开心心233 谢谢喵

B:Square Game

需要求 SG 值的 Nim 博弈。对一个固定的 n ，注意到 $n^2 + 1, n^2 + 2, \dots, n^2 + n$ 个石子的后续局面是完全一样的。所以对石子数上界 m ，真正需要考虑的状态只有 $n^2, n^2 + 1, 1 \leq n \leq \sqrt{m}$ 一共 $O(\sqrt{m})$ 种状态。对每个不同的状态 S ，后继状态也只有 $O(\sqrt{S})$ 种，直接用 mex 函数记忆化搜索 SG 值即可。这样总复杂度是 $O(\sqrt{m^2}) = O(m)$ 的。

C:室温超导

1到m依次考虑T的每一个后缀能合成的方案数，一个后缀T[k...m]贡献的方案数为前缀S[1...n-m+k]的所有本质不同子串数，减掉以T[k-1]结尾的本质不同子串数，因为后缀S中以T[k-1]结尾的子串和T[k...m]拼接所得到的串已被后缀T[k-1...m]重复计算。

对字符串S从左往右建立后缀自动机，统计每个字符结尾的本质不同子串数

$\text{sum}[S[i]] += \text{cnt}[i] - \text{cnt}[i-1]$ ，sum表示每个字符结尾产生的本质不同子串数，cnt[i]表示前缀S[1...i]本质不同子串数

最后判断下T的每一个后缀是否也能合成，复杂度 $O(N)$

D:Container Orders

对于每个order，我们可以把 h 二进制拆分，把所有orders算出一个 $\text{req}[i]$ 表示 2^i 的箱子需要多少个。然后从小到大枚举 i ，对于当前的需求，显然我们贪心地选择代价更小的箱子，而如果有当前用不完的箱子，我们也贪心地从小到大选择代价更小的箱子两两合并，相当于之后他们的重量为 2^{i+1} 了

E:yytree

先不考虑 $(-1)^d$

把 $(x + k * d)$ 分成两个维护 x 与 $k * d$

可以发现 x 很好维护，问题在于 $k * d$

可以发现对于询问 v 而言，它的祖先 u 上的修改对他的影响为

$$x + k * d = x + k * (\text{dep}[v] - \text{dep}[u] + 1)$$

因此可以把 x 和 $k * (-\text{dep}[u] + 1)$ 维护一下，再维护一下 $k * \text{dep}[v]$ 即可

再考虑 $(-1)^d$

u, v 都是奇数，那么他们两之间就是偶数，如果一奇一偶就是负数

因此在插入和查询的时候，都乘上系数 $(-1)^d$ 即可， d 为自身的深度

复杂度: $O((n + m) \log n)$

对于操作三，用set维护dfs序上操作1的区间，删除符合条件的操作1即可

F:necklace

本题考查图论（最短路）与计数

考虑将问题分解，因为稀有金属转换与具体的稀有金属序列无关，可以考虑先预处理出来所有稀有金属转换。

子问题1:已知一些稀有金属转换的权值，怎么得到最优的转换方式？

最短路的想法。

多源多汇最短路Floyd，时间复杂度 $O(c^3)$ 。(c为宝石的种类数)

子问题2:已知了两两之间转换的最小权值，对于给定的起点怎么计算最优的转换模式？考虑到问题的规模，可以直接对每个位置枚举转换为哪种稀有金属，时间复杂度 $O(n * c)$ 。

那么对每个起点求解一次，但这样时间复杂度太高 $O(c^3 + n^2 * c)$ 。

可以 $O(c^3)$ 预处理记录稀有金属对转换为哪种稀有金属最优，统计两个环中颜色的数目，然后可以 $O(1)$ 得到稀有金属对的答案

复杂度 $O(c^3 + n^2)$

G:套娃收纳

假设已经确定了一个方案，如果两个套娃只考虑外径可以交换位置，那么从这两个套娃开始的对应的两个后缀可以交换。如果一个在序列首位的套娃A可以与一个不在首位的套娃B交换位置，那么如果A的外径大于B，则交换之后得到的方案具有更小的总体积。

对外径从大到小考虑。设对已经确定了所有外径大于 r 的套娃，并且有 m_r 个序列尾部的套娃内径大于等于 r 。记 n_r 为外径为 r 的套娃数量。若 $n_r \leq m_r$ ，则有 $A_{m_r}^{n_r}$ 种方案将 n_r 个套娃接在已有序列尾部；若 $n_r > m_r$ ，则有 $A_{n_r}^{m_r}$ 种方案将 m_r 个套娃接在已有序列的尾部。

H:但工程经验非常不足

考虑每个知识点能贡献的能力值，我们希望给他分配更高的gcd。所以就从大到小枚举gcd，然后枚举当前所有的gcd的倍数，如果个数大于一个，就说明这些数字可以被安排成一个集合学习。复杂度是一个调和级数 $O(W \ln W)$

upd：但是这种贪心是错误的，因为没有考虑可能漏掉的1个数字的贡献

比如4 7 7 28，枚举到7就全贪完了，但 $4+4+7+7 > 3 * 7$

I:hard math

数位DP， $dp[2e5][10][2][2]$ 表示，因为不关心当前状态有哪些数字，只关心不同数字的数量，所以状态由 2^{10} 可以简化到10，但在dfs的时候维护 2^{10} 的状态就行了，后面两个状态就是判断上界和下界，因为上下界长度相同且没有前导0，记忆化递归写起来非常优雅。

tiger2005给出了一种非递归的计数dp写法，比赛队伍的代码中也看见了各种dp姿势都行。

J:radius

先考虑x轴，二分球体半径，每个点可被覆盖的球心在x轴上是一个区间，因此二分判断下有没有至少 $n/2$ 个区间有交即可，复杂度 $O(N * \log N * \log N)$ 。

K:tourist

本题考查动态规划，组合计数

没有必须经过的城市约束时，可以用dp，设 $f[i][j]$ 为 i 时刻在城市 j 的方案数，然后矩阵乘法优化

时间复杂度 $O(n^3 \log d)$

有必须经过的城市约束时，注意到城市较小，时间较大，可以用容斥来解决城市约束

先暂且不理 k 个重要城市，构建矩阵用容斥解决重要城市的问题，由于 k 最多只有 7，所以 2^k 枚举重要城市{可以经过/强制不能经过}，强制不能经过只需在构建矩阵时无视与这个城市相连的边就行了

最后 $ans = f_{\text{强制关闭0个重要城市}} - f_{\text{强制关闭1个重要城市}} + f_{\text{强制关闭2个重要城市}} - f_{\text{强制关闭3个重要城市}} + \dots$

时间复杂度 $O(n^3 * \log d * 2^k)$