公约数神庙

首先进行一些特判：

- 若 $x=y$，输出 `Shi`
- 若 $a[x]=1 \vee a[y]=1$，输出 `Fou`
- 若 $a[x]=0 \vee a[y]=0$，若 $[x,y]$ 之间有 $a[x \le k \le j] > 1$ 输出 `Shi`

定义所有包含质因子 p 的所有 $a[i]$ 是 p 类神庙，记录：

- $go[i][p] = a[i]$ 能走到的第一个 p 类神庙的位置

回答询问 (x,y) 时，枚举 $a[y]$ 的所有质因子 p，检查是否满足：

- $go[x][p] \le y$

预处理 1000 内的质数（168个），以及每个 $a[i]$ 的质因子，递推 $go[i][p]$ 时只在自己的质因子里枚举，小于1000的数最多有4个不同质因子，复杂度是 $O(n \times 4 \times 200 + q \times 4)$。

栈法师

栈法杖的数量是 1 或者 2。

$k=1$ 的情况平凡，不赘述。

对于 $k=2$ 的情况，做法很多，其中之一是：把弹出的数都放入栈法杖 A 中，栈法杖 B 仅作为中转，设下一个要输出的数为 x：

- 若 x 还在序列 a 中，则一直进行 $(1,1)$ 操作直到 x 进入栈法杖 A，然后进行一次 $(2,1)$ 操作将其输出；操作次数 = 当前序列 a 长度 - x 初始位置 + 1。
- 否则 x 在栈法杖 A 中，进行 $(3,1,2)$ 操作直到 x 是栈顶，进行一次 $(2,1)$ 操作将其输出，然后进行 $(3,2,1)$ 操作清空栈法杖 B；操作次数 = 当前 A 栈 size - x 初始位置之后 $< x$ 元素数量 - 1。

"x 初始位置之后 $< x$ 元素数量" 可以使用树状数组预处理。

总复杂度 $O(n \log n)$。

城堡考古

由于 m 很小，可以状压每一列的状态（1表示要伸到后一列），状态数为 $2^m=64$，列和列之间的转移可以用矩阵快速幂进行优化，注意：

- 需要写一个高精度的$10->2$进制转换（可以每次暴力做高精度除2，复杂度是对的），位数大概 $\times 3$
- 题目要求的是一个差分形式，需要矩阵多开一维记录前缀 sum

直接实现的话大概能跑65分。

按照网格状压DP的套路，有很多状态其实是到不了的，dfs 预处理合法且能够从 $s=0$ 到达的状态，发现只有20个（其实是一个组合数，m 列恰有 $\binom{m}{m/2}$ 个合法状态）。

复杂度变为 $O(20^3 \times len)$，这样就能通过了。

生命之树

以下简称原题中`注入生命露滴`为`染色`

【算法1】成链

对于一个点 u，需要在一个区间 $[pre[u], nxt[u]]$ 中染色一个点， 可以二分预处理出来。设 $dp[i]$ 代表在 i 点染色、搞定前 i 个点的最小花费，枚举上一个染色位置转移：

- $dp[i] = \min(dp[j]) + c[i], j < i$
- 转移条件是 (j, i) 中的点 k 满足 $pre[k] \leq j \vee nxt[k] \geq i$，即 $j \geq \max\{pre[k], nxt[k] < i\}$

从大->小枚举 j 就可以顺带维护出 $\max\{pre[k], nxt[k] < i\}$ 的值，复杂度 $O(tn^2)$。

【算法2】菊花

可以按 1 是否染色讨论。

若 1 染色，对于其他点 u，若 $d[u] < dist(1, u)$ 则需要染色。

否则，设 x 为 $dist(1, x)$ 最小的染色的点，那么对于其他点 u，若 $d[u] < dist(1, x) + dist(1, u)$ 则需要染色。

枚举 x 然后两种情况取最优即可，复杂度 $O(tn^2)$。

【算法3】$d, w = 1$

树形DP，设 $dp[u][0/1/2]$ 代表搞定 u 子树，根的状态为 $0, 1, 2$ 时的最小花费：

- $dp[u][0]$ 代表 u 没染色，且 u 的限制未满足
- $dp[u][1]$ 代表 u 没染色，且 u 的限制已满足
- $dp[u][2]$ 代表 u 染色，且 u 的限制已经满足

转移：

- $dp[u][0] = \sum \min(dp[v][1], dp[v][2])$
- $dp[u][1] = \sum \min(dp[v][1], dp[v][2]) + \max(0, \min(dp[v][2] - dp[v][1]))$
- $dp[u][2] = \sum \min(dp[v][0], dp[v][1], dp[v][2]) + c[u]$

答案为 $\min(dp[1][1], dp[1][2])$，复杂度 $O(tn)$。

【算法4】$w = 1$，所有 c 都相等

贪心，按（深度-限制距离）从大->小考虑所有点，若当前点限制仍未满足，将其 d 级祖先染色，同时标记其能满足哪些点的限制（这个信息可以 $O(n^2)$ 预处理）。

复杂度 $O(tn^2)$。

【算法5】

对于每个点，称距离其最近的染色点为配对染色点。

观察到：对于子树 u 的所有点，其配对染色点至多只有1个在子树外（若有多个可以只留距离 u 点最近的）。

设计动态规划 $dp[u][j]$ 代表搞定 u 子树、且 u 和 j 配对的最小花费，记 $f[i] = \min(dp[i][j])$，答案为 $f[1]$。

考虑转移，对于 $dist(i, j) \leq d[i]$ 的状态，考虑 u 的每个儿子 v：

- 若 v 子树的配对集合完全在子树内，贡献为 $f[v]$；
- 若 v 子树的配对集合除了 j 点都在子树内，贡献为 $f[v] - c[j]$；
- 若 v 子树的配对集合有在子树外、且不是 j 的其他点 j'，则不是最优解（子树外多余1个）

转移方程为 $dp[u][j] = c[j] + \sum \min(f[v], dp[v][j] - c[j])$，复杂度 $O(tn^2)$。