Solution

递增路径(increase)

签且原。

按边权从大往小扫描，设 f_u,g_u 分别表示只考虑当前加入的边，以 u 作为起点，Alice 或 Bob 先手时操作的轮数，则加入边 (u,v) 时，只需令

- $f'_u \leftarrow \max(f_u, g_v+1)$
- $g'_u \leftarrow \min(g_u, f_v+1)$
- $f'_v \leftarrow \max(f_v, g_u+1)$
- $g'_v \leftarrow \min(g_v, f_u+1)$

注意单独讨论一下 $g_u=0$ 或 $g_v=0$ 的情况。

最终 $f_1,f_2,\ldots,f_n$ 就是答案。时间复杂度为 $O(n+m)$。

机房惨案(network)

> Codeforces 825 G Tree Queries

妙题，想起来可能比石头游戏难一点。但是很好写。

考虑将第一台被 JC 的电脑当做根，先 DFS 预处理每个点到根路径上的编号最小值，记 x 到根路径上的路径最小值是 a_x。

然后修改 x，对 x 子树内的答案是没有影响的，而对于 x 子树外的点，假设这个点为 y，用来更新的答案是 y 到 x 路径上的编号最小值，考虑到我们至少已经有了 a_y 的贡献，所以我们可以将 y 到 x 路径上的编号最小值转化为 $\min(a_x, a_y)$。

所以对于修改，我们维护一个全局被 JC 的电脑的 a_x 的 $\min$ 的变量 mn，每次修改对 a_x 取 $\min$。

然后对于询问，我们的答案就是 $\min(a_x, mn)$。

时间复杂度 $O(n+q)$。

石头游戏(stone)

一个显然的最优策略：对于每一个局面，一直选择第一个 $a_i = i$ 的位置进行操作可以操作尽可能多次。

因为这样操作不会破坏其他已经可以操作的位置的性质，同时也会让前面的各堆石头更加接近操作。

显然只有后面的石头堆可能影响前面的石头堆。不妨直接倒着 DP。对于第 i 堆石子，我们只关心后面的石子堆给了这一堆石子多少石头。

故记 $f(i, j)$ 表示从后往前处理到第 i 堆石头，后面的这些石子堆一共提供了 j 次 $+1$ 操作的方案数。

接着我们枚举第 i 堆石子的初始值 x，如果可以操作$(x \le i)$，那么操作次数即为 $\lfloor \frac{x+j}{i} \rfloor$，继续转移即可，操作次数不超过 n^2 次，暴力枚举状态转移的时间复杂度是 $O(N^4)$。常数 $\frac{1}{4}$，直接跑即可。

简单计数(count)

30分： $O(n^2m^2)$

记 $f[i][j]$ 表示做到第 i 个音符，最后一个音符为第 j 种的方案数。

根据题意，我们可以直接枚举，将第 k 个音符到第 i 个音符定为第 j 种音符（需要满足限制条件 $i-k+1\le a_j$），则有$f[i][j]=\sum_{k=i-a_j}^{i-1}\sum_{x\ne j}f[k][x]$。

50分： $O(n^2m)$

如何优化式子呢？我们记 S_i 表示前 i 个音符的方案数和，则有$S_i=\sum_{k=1}^{i}\sum_{x=1}^{m}f[k][x]$
$S_i-S_{i-a_j-1}=\sum_{k=i-a_j}^{i-1}\sum_{x=1}^{m}f[k][x]$。

于是我们就可以简化一下 30分 的式子，如：
$f[i][j]=\sum_{k=i-a_j}^{i-1}\sum_{x\ne j}f[k][x]=S_i-S_{i-a_j-1}-\sum_{k=i-a_j}^{i-1}f[k][j]$。

70分： $O(n(n+m))$

我们再来优化一下式子，记 $g[i][j]$ 表示前 i 个音符，末尾均为 j 的方案数，则有：
$g[i][j]=\sum_{k=1}^{i}f[k][j]$。

于是我们可以简化一下 50分 的式子，如：
$f[i][j]=S_i-S_{i-a_j-1}-\sum_{k=i-a_j}^{i-1}f[k][j]=S_i-S_{i-a_j-1}-(g[i][j]-g[i-a_j-1][j])$

100分： $O(n^2+m)$

由于 $a_i\le n$，若 $a_i=a_j$，则 $f[*][i]=f[*][j]$，再优化一波，该记 $f[i][j]$ 为做到第 i 个音符，最后一个音符为是 $a_i=j$ 的方案数，再改一改式子就 OK 了。

工团运输(synd)

Subtask 3

写出距离和 $dis = \sum |(x-x_i)(y-y_i)|$。

把坐标按照 x 排序，得到这个式子:

$$\sum_{i=1}^{p}(x-x_i)|y-y_i| - \sum_{i=p+1}^{n}(x-x_i)(y-y_i)$$

其中 $x_p \le x \le x_{p+1}$。

然后对与任意一个确定的 $x, y_q \le y \le y_{q+1}$，这个式子是一个自变量为 y 的一次函数 $f(y) = kx + b$。

如果 $k > 0$，y 取 y_q 最优，如果 $k < 0$，y 取 y_{q+1} 最优，如果 $k = 0$，$y = y_q$ 是最优之一。

故答案 $y \in y_i$。同理 $x \in x_i$.同时这里也说明了：最优解一定是整数。

然后我们枚举 x, y，逐一带入计算，时间复杂度为 $O(n^3)$。

Subtask 4:

可以打表发现 x 一定时，这是关于 y 的单峰函数，而设 $f(x)$ 为 $\min_{y\in\{y_i\}} \sum |(x-x_i)(y-y_i)|$，可以发现 $f(x)$ 也是单峰函数，然后三分套三分做了。

证明：容易发现 $\frac{\partial^2 dis}{\partial y^2} = C$，$C$ 为一常数，故在 x 一定时 dis 是关于 y 的单峰函数。然后 $\lim_{y\to\infty} dis = \infty$，所以这是下凸函数，然后关于 x 同理，然后就证完了。

注意开 `double` 而不是 `long long` 。由于不输出具体距离，所以说可以认为 `double` 精度是够用的。

有一个乱搞。注意到在数据较大的时候，一般情况下答案会比较靠近坐标的中位数。并且我不知道怎么卡。枚举可。