A

前面部分分给一些暴力枚举前 i 个，每个序列和分别为多少。

或者发现了由于答案只和 A_i, B_i 差值有关，于是只计算 $|A_i - B_i|$ 是多少而不具体考虑 A_i, B_i 是多少。剩下的再通过插板计算方案数即可。

首先，$|A_i - B_i| = \max(A_i, B_i) - \min(A_i, B_i)$ 。

然后原式就变成

$$\begin{aligned}\sum_{i=1}^{n}|A_i - B_i| &= \sum_{i=1}^{n}(\max - \min) \\ &= \sum_{i=1}^{n}(\max + \min) - \sum_{i=1}^{n} 2\min \\ &= 2m - 2\sum_{i=1}^{n}\min\end{aligned}$$

令 $s = \sum_{i=1}^{n}\min$，肯定有 $0 \le s \le m$ 。那么最后答案就是 $(2m-2s)^2$。

那么确定每个 $\min(A_i, B_i)$ 对 s 的贡献，有 $\binom{n+s-1}{s}$ 种不同的贡献方式（插板法，n 个非负整数和为 s 的方案数）。

剩下还差 $m - s$ 需要分别分配给 A 和 B 以便各自序列和能为 m 。但是不能同时增大某个 A_i 和 B_i，否则 $\min(A_i, B_i)$ 会发生改变，所以对于一个 i 只能增加其中之一。

那么考虑到最终的序列有三类位置，一种是 A_i 严格大于 B_i ，一种是 $A_i = B_i$，一种是 A_i 严格小于 B_i 。若枚举三类的个数分别进行插板法的话可以解决，但是复杂度较高。

所以只枚举 $A_i > B_i$ 的位置个数，假设有 x 个，方案数为 $\binom{n}{x}$，那么要把 $m-s$ 分配给这 x 位置，每个位置至少被分配 1 个。这也可以用插板法。

而对于剩下的位置，就是将 $m-s$ 分配给这 $n-x$ 个位置，但是某个位置可以不分配（对应着 $A_i = B_i$ 的情况），也用组合数就好了。

时间复杂度 $\mathcal{O}(nm)$ ，最后答案就是：

$$\sum_{s=0}^{m}\sum_{x=0}^{n}\left((2m-2s)^2\binom{n+s-1}{s}\binom{n}{x}\binom{m-s-1}{x-1}\binom{n-x+m-s-1}{m-s}\right)$$

B

对于前 60 分，容易用矩阵表示每个时刻的转移，一个向量 v 表示每个点在当前时刻的期望人数，每个时刻需要的值就是 v_n，直接 $O(n^3T)$ 求答案。额外的 20 分用矩阵优化最后一大段转移即可。

对于大段的转移，发现除了加入新的人的 m 个时刻以外，其它时刻的 v 都是前一个时刻的 v 乘上固定的转移矩阵 M。注意到求矩阵成向量的某一项是 $O(n)$ 的，所以预处理一下 M 的 $1 \sim B$ 次幂，对每 B 个位置直接计算一下 $(M^i v)n$ 然后更新 v，复杂度变为 $O(n^3B + n^2\frac{T}{B} + Tn + mn^2)$，取 $B = \sqrt{\frac{T}{n}}$ 可以通过，可以适当调一下 B 的值。可能有其它优化方式可以通过本题。

C

当 $k = 1$ 的时候，只需要模拟一遍然后用 map 或者别的去重看一下有多少个点即可。

设完整读取完一次后的坐标为 (a,b) ，第一次经过的所有点为 (x_i,y_i)，那么第二次经过的所有点就是 (x_i+a,y_i+b) 。同理，第 k 次的点集为 $(x_i+(k-1)a,y_i+(k-1)b)$。

也就是说对于单独的一个点，它会按照上述方式成直线移动，我们只需要考虑哪些点可能会重叠。

显然，若 $(x,y)=(x'+ia,y'+ib)$ ，那么这两个点是有可能重叠的。于是我们把所有的满足可能重合的点放在一起，即 $(x \bmod a, y-\lfloor \frac{x}{a}\rfloor b)$ 相同的放在一起。然后按照他们原本的坐标大小排序即可。

然后把放在一起的按 x/a 的大小关系排序，相邻两个的贡献为 $\min\{(x_2-x_1)/a,k\}$。

对于 $b=0$ 可能看起来要直观点，也是一个很好的提示了。

D

测试数据中的 $[l,r]$ 比下发样例中的更长。

测试点 $3\sim 6$ 预计可以通过基于值域预处理的快速 GCD 或者 binary GCD 做到 $O(V+nq)$ 或小常数 $O(nq\log V)$，后者可能会因为评测机性能被卡掉，两者均可用于分块做法的常数优化。

分块做法：设块长为 B。难点在于预处理 $f_{i,x}$ 表示第 i 块中元素全为 x 时的答案。记 $F_i=\frac{x\cdot b_i}{\gcd^2(x,b_i)}$，因为只求 F_i 的最小值，所以可以枚举 $d|x,d|b_i$，则 $F_i=x\cdot\min_{d|x,d|b_i}\frac{b_i}{d^2}$，在块内枚举一下 d 计算 $\min_{d|b_i}b_i$，再对于 $d|x$ 的 x 结算贡献即可。这部分复杂度 $O(\frac{n}{B}V\log V)$，可以精细实现到 $\log\log V$ ，但没必要。其余部分大体实现上没有困难，适当卡常应该可以通过本题。完整代码参考 std 文件夹中的 pal2.cpp。

RMQ 做法：操作 1 是区间推平，考虑颜色段均摊，完整段段内维护答案，查询的时候单独查询两边的散段，中间整段可以挂到线段树上查询。于是问题转为若干次给定 l,r,x 求 $\min_{i\in[l,r]}\frac{b_i}{\gcd^2(x,b_i)}$，操作 1 和操作 2 的散段都是询问这个问题。类似地枚举 $d|b_i$ 统计 $\min_{d|b_i}b_i$，这个 RMQ 问题可以对每个 d 开个 ST 表来统计，所有 d 的 ST 表中的总元素为 $\sum\sigma(b_i)$，并不大。查询时枚举 $d|x$ 去对应的 ST 表完成查询，操作 1 再额外去线段树上更新一下。跑得很快，常数不大。