1.树（tree）

从小到大枚举距离，每次删去 G 中为当前距离的边。

发现在边删完之前，直径一直都在同一联通块。

考虑寻找每个点最后一次被删的边，发现是到距离它最远的点的边。

发现距离每个点最远的点一定是直径的某一端，那么这就确保了只要有边存在，它们都一定可以连到直径上。

于是找到每个点最后一条边的长度，差分计算答案即可。

2.宠物狗方阵（dog）

枚举 d，再枚举前缀取的连续段数 k，那么后缀会取 $\lfloor \frac{n}{d} \rfloor - k$ 个段，总共有 $\lfloor \frac{n}{d} \rfloor$ 个分割方式。

用字符串哈希维护连续段，维护出每种相同段的出现次数。我们可以使用简单组合数学算出方案数。

再用集合哈希维护集合去除掉相同的分割方式即可。

注意要使用要哈希表维护去掉 $\log$。

计算时容易动态维护出方案数。总复杂度 $O(n\ln n)$。

3.波特方阵（matrix）

算法 1

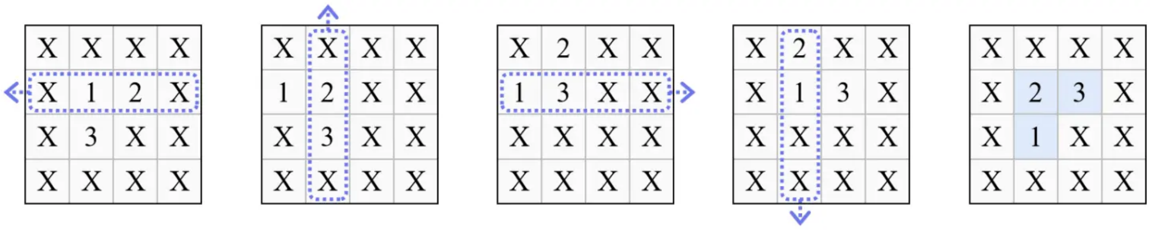
直接爆搜出每种状态，期望得分 15。

算法 2

$p_i = i$ 时，发现每列是独立的。单独拿出来，问题变成了序列上的。我们把属于 q 的同一置换环的分成一组，每组的方案数即为循环节长度，总方案即为每组方案的 lcm。最后再将所有列的方案相乘即可。结合算法 1 期望得分 25。

算法 3

当 p, q 都为大环，且 n, m 为奇时，先判掉 n 或 m 为 1 的情况。我们把矩形按置换环上顺序重排，观察发现若将每行首尾拼接成序列，其逆序对数的奇偶性是不变的。大胆猜测我们可以得到所有逆序对奇偶性相同的矩形。所以当矩形的数互不相同时，答案为 $\frac{(nm)!}{2}$，否则为 $\frac{(nm)!}{\prod cnt_i}$，其中 cnt_i 为 i 的出现次数。证明的话我们发现可以做出如下操作：



不难发现可以通过这种轮换得到所有逆序对奇偶性相同矩阵。结合前面期望得分 35。

算法 4

当 p,q 都为大环但 n,m 不全为奇时。我们发现上面的轮换无论 n,m 奇偶性还是不会改变逆序对奇偶性的，那我们可以通过先操作一次改变逆序对奇偶性然后得到所有矩形。结合前面期望得分 50。

算法 5

把 p,q 按置换环拆开成若干组矩形。对于每组矩形，设他们行列属于的置换环为 u,v，矩阵大小为 $n\times m$。分类讨论计算贡献：

- 若 n,m 其一为 1，使用算法二即可。
- 否则若存在两个相同的数，贡献为 $\frac{(nm)!}{\prod cnt_i}$。
- 否则，我们先将贡献除以 2，再考虑少乘的。设 c_x 表示置换环 x 被操作的奇偶性（同时包括 p,q 的置换环），若 n,m 都为偶数，则矩形的奇偶性取决于两端点点权的异或值。我们将 u,v 连边，边权为 $c_u\oplus c_v$。对于有奇数的情况，我们将无影响的那个点设为 0，同样连边即可。建完图后，对于每个连通块，我们可以对于一个 dfs 树上的所有边任意钦定边权，非树边的在钦定完树边之后权值也就确定了，所以其贡献为 2^{S-1}，S 为连通块点数。

实现可做到 $\mathcal{O}(nm)$，期望得分 100。

4.工厂（factory）

算法 1

对于每个问题单独考虑

$t=1$ 直接转成 $t=0$

不难发现有个很显然的网络流做法：

构造二分图，所有机器人为左部点，所有容器为右部点。

源点向每个左部点连 c_i 的边，所有右部点向汇点连 a_i 的边。

第 i 个左部点向区间 l_i,r_i 的右部点连流量无限的边。

这样，跑个最大流就是答案，期望得分 15。

算法 2

由于最大流等于最小割，所以不妨尝试求最小割。

这个图的最小割相当于选择一些总权值尽量小的机器人和容器，使得不存在 $l_i \le j \le r_j$ 且 i, j 都不选。

于是有一个 n^2 的 dp。

从前往后枚举每个点，设 $f_{i,j}$ 为当前考虑前 i 个点，最后一个没有被选的点位置为 j，最小的代价和。

转移的时候考虑当前的这个点选还是不选，然后枚举所有以这个点为右端点的区间，更新这些区间的贡献。

可以发现，转移的过程相当于单点修改，区间加，区间最值，套个线段树即可 $O(n \log n)$ 维护，期望得分 $25 \sim 30$。

以上是超纲内容，可忽视，只是便于理解。

算法 3

以从左往右为例。

对于每个点，发现在所有包含这个点且有剩余量的区间中，优先从右端点最近的区间得一定最优，右端点远的区间可以给更右边的点。

发现每个点，令此时这个点的剩余容量为 c；右端点最近的包含这个点的区间为 $[l, r]$，剩余量为 v；满足条件 $c > 0, v > 0$。

则会贡献 $\min(c, v)$，$c \leftarrow c - \min(c, v)$，$v \leftarrow v - \min(c, v)$。

可直接拿堆维护，一样是 $O(n \log n)$，期望得分 $25 \sim 30$。

算法 4

由于 $t = 1$，所有区间都覆盖点 x。

优先给 $1, n$ 两端，最后考虑给 x，这样一定也是最优的。

考虑包含 x 的极长放满的点的连续段，那么对于一个区间有两种情况。

1. 一个区间没有被这个连续段完全包含，则这个区间一定没有剩余，且会全部给连续段两端以外的部分，贡献为这个区间的总量。

2. 一个区间被这个连续段完全包含，则这类区间的总贡献一定是这个连续段的总量，因为每个这样的区间都没位置多给，也不能少给。

若连 x 这个点都没有放满，则所有区间都为第 1 类。

所以答案为两类的和。

考虑这样算有哪些不合法：

1. 一个第 1 类区间把连续段两端以外的部分放满了导致多出剩余量，贡献会小于区间的总量，算出的答案会更大。

2. 第 2 类区间无法把连续段放满，这类区间的总贡献会小于这个连续段的总量，算出的答案同样会更大。

一种方案可以取得 ans 时，考虑这种情况时包含 x 的极长放满的点的连续段，根据贪心，发现贡献和刚好为 ans。

考虑枚举这个连续段，将所有的情况取 $\min$，则取得 ans，结合上述期望得分 $40 \sim 45$。

算法 5

令包含 x 的极长给满的点的连续段为 $[l, r]$。

不妨对于每个 l 维护每个 r 的答案。

可以发现，每一个区间产生的贡献都可以扫描线维护。

而包含 x 的答案即为 $\min\limits_{i \le x, x \le j} f_{i,j}$，可以扫描线套历史最值线段树 $O(n \log n)$维护，结合上述期望得分 65。

算法 6

考虑 $t = 0$ 区间的贡献。

对于有交点的区间，与 算法 5 同理。

对于与 $[l, r]$ 没有交点的区间，可以发现这些区间要么右端点小于 l，要么左端点大于 r，发现类似 算法 3。

以 x 左端为例。

发现贪心部分一样，将区间 $[l, r]$ 贡献只加入 $\forall\limits_{r<x} x$ 即可。

而这类区间无法给到连续段，不会破坏 算法 4 的原理。

这两边的贡献只需用 上述算法 或 算法 2 正反跑一边即可。

直接在线段树中动态加入两边的贡献，总复杂度 $O(n \log n)$，期望得分 100。