Solution

平均数(ave)

纯签，越后面加入的数字权重越大，排个序贪心就好。这道题今天的任务就是把 100 分送到各位手上。

序列（sequence）

不难，而且暴力给了好多。

先搞个暴力，显然我们只需要知道上一个元素就可以了。

我们记：$f_{i,j}$ 表示确定前 i 个数字，第 i 个数字为 j 的方案。

显然只有 j 的大于 1 整数倍转移不了。暴力转移的时间复杂度均摊为 $O(\log m)$。

最终我们可以得到一个 $O(nm\log m)$ 的做法。这个有 50。

想要拿到更高的分数就要换思路了，除了在状态上记录上一个数字，我们其实还可以直接分段转移，即将状态记为 f_i，每次枚举一段。

不过段与段之间我们没法确定，因此只能尝试不管，也就是考虑容斥，进一步，我们发现，不合法的二元组 (a_i, a_{i+1}) 满足 $a_{i+1} \geq 2a_i$，这样的段数长度不超过 $\log M$。因此考虑转移时直接枚举一段不合法的位置进行容斥。

我们记 $f(S)$ 表示钦点位置集合 S 不合法的方案数(数对(a_i, a_{i+1})不合法我们认为是 $i+1$ 这个位置不合法)，其中 $S \subseteq \{2, 3, \ldots, n\}$。那么最终的答案就是：

$$\sum_{S\subseteq\{2,3,\ldots,n\}} (-1)^{|S|} f(S)$$

我们记 $f_{i,0/1}$ 表示目前已经考虑了前 i 个位置，$|S|$ 为奇/偶数的方案数。我们直接事先预处理长度为 k 的生成连续 $k-1$ 个不合法位置的方案数 g_k。

g_k 的预处理大概需要 $O(m\log^2 m)$。f 的转移复杂度为 $O(n\log m)$。

最终复杂度为 $O(n\log^2 m)$。

环上排序（ring）

满足的条件是存在条件，众所周知，这种条件不好搞，建议先找逆命题(也就是容斥)，即 $\forall i \in [1,n], a<b<i \vee i<a<b$。

下一步，想办法去掉数字，留着数字 dp 状态大概率是记不下来的。哪怕是写 40 分暴力，你也得想办法把排列枚举变成二进制状压。

显然，被操作过的数字都是比 i 小的，不妨把整个环上的点记为黑白两种颜色，最开始所有的点都是白色的，随后我们每次选择一个白色的点，交换该点周围相邻的两个点。容易发现这个过程就是原问题的过程。

因此我们有了一个 $O(n2^n)$ 的状压做法了。注意逆命题限制的含义变成了，每次我们找到一个点，要求两边都是白点或者两边都是黑点。

容易发现在这种情况下，一个被染黑的店无论如何都无法变白，换句话说，黑点将整个环已经拆分成了若干不相关的段。

注意到偶数段其实是无解的。因为该段内的任意操作都只会生成一个奇数段和一个偶数段，而最短的偶数段，即两个白点，是无法操作的。

这就意味着如果 n 为奇数(任意操作一次后就变成一个奇数段)，答案就是 $n!$。至此就获得了 50 分。

奇数段显然可以 dp，我们记 f_n 表示一个长为 $2n-1$ 的奇数全 0 段变成全 1 段的方案数，考虑枚举第一次操作的位置：

$$f_n=\sum_{i=1}^{n-1}\binom{2n-2}{2i-1}f_if_{n-i}$$

最终的答案就是 $n!-n\times f_{\frac{n}{2}}$。

平面（ds ）

先转换成序列问题：

将所有点按照 x 坐标从小到大（从左到右）排序， x 坐标相同时按 y 坐标从大到小（从上到下）排序，那么问题就转化为在某个下标区间中保留权值（也即 y 坐标）在一段区间内的数，求子序列中后缀最大位置的个数（位置 p 大于位置 q，当且仅当 p 上的权值大于 q 上的 权值，或p, q 上的权值相等且 p 在左边）。

在本题中我们认为 n, q 同阶。

算法一

对于每次询问，找到 x 坐标不大于 R 的序列中最后位置 v 和 x 坐标不小于 L 的序列中最前位置 u，从 v 到 u 扫描一遍，如果遇到更大的位置答案就加一，最后输出答案。时间复杂 度 $\mathcal{O}(n^2)$。

算法二

将 $[1, n]$ 均匀分为 $\sqrt{n}$ 块，对于每一个询问，我们同样找到 x 坐标不大于 R 的序列中最后位置 v，并设 L_0 为不超过 R 的最后一个块端点并找到 x 坐标不小于 L_0 的序列中的最后位置 u。使用算法一做法求出这一段的答案，并将 R 整合到块端点 L_0 上，更新询问的 D 限制。

再对于每一块，暴力将 U 从低到高扫描，把点加入维护答案位置的单调栈中。当扫描线与一个询问的上边界重合时，在单调栈二分能进行贡献的区间进行贡献即可。

时间复杂度 $\mathcal{O}(n\sqrt{n})$。

算法三

考虑利用楼房重建的 $\mathcal{O}(n \log^2 n)$ 线段树做法。将询问离线按 U_i 从小到大排序，使用线段 树维护序列区间$[u, v]$ 的最大值以及区间后缀最大位置的个数，边插入新点边进行区间询问， 可以得到复杂度保持 $\mathcal{O}(n \log^2 n)$ 的做法。

算法四

对于每个询问，找到在它的两个区间限制约束下的最大位置，这显然是第一个算进答案的数，这个问题可以使用 $\mathcal{O}(n \log n)$ 复杂度的数据结构+按 L 从大到小的扫描线得到。

接下来，我们考虑分治。求出分治中点，对于不跨越中点的询问，我们递归到子问题处理。 对于所有跨越中点的询问，我们考虑直接计算其在左部的贡献，再递归到右部处理。为此，我们需要找到其在右半边的最大位置。使用并查集维护前缀，从右到左扫描删点即可。

对于左半边的所有数，找到中点前不大于它的最大位置，向其连边。求询问贡献时，从其 原本的左端点开始跳，跳到最后一个不小于其右半边最大值的位置，跳的次数就是左半边的贡献，同样可以使用带权并查集+扫描线优化。

小细节：排序可以基数排序，注意缩小值域。

时间复杂度 $\mathcal{O}(n \log n\ \alpha(n))$。

算法五

按 y 坐标大到小顺序枚举点，大小相同从左到右枚举。当枚举的数的 y 坐标不高于一个询问的 U 时，我们就将该询问放入维护集合之中，当枚举的数 y 坐标低于一个询问的 D 时，我 们把它取出计算答案。

再考虑放入一个数时会对当前维护集合里的询问产生什么影响，它会把所有 L 不超过该点 x 坐标的询问的答案加 1，并且把这些询问的 L 推到这个位置。

那么我们使用势能线段树维护，在加入区间时把区间挂到其右边界的位置上，线段树维护每个区间的 L 的最大值， L 的次大值和增量标记。时间复杂度 $\mathcal{O}(n \log n)$。

树和森林（lct）

记树的大小为 size_i，第 i 棵树上的所有点到点 j 的距离和为 $\mathrm{dis}_{i,j}$，点 u 与点 v 间的距离 $d(u,v)$。

对于所求答案，包括每棵树内部的和树与树之间的，对于前者我们可以直接 dp 求出。

对于后者，若只有两棵树，应连接两棵树中值最大的。

若有三棵树，我们可以枚举三棵树的相对位置，设第一棵树与第二棵树通过 (x,y) 相连，第二棵树与第三棵树通过 (u,v) 相连，则树与树之间的答案为：

$$\mathrm{size}_2\mathrm{dis}_{1,x}+\mathrm{size}_1\mathrm{dis}_{2,y}+\mathrm{size}_1\mathrm{size}_2+\mathrm{size}_3\mathrm{dis}_{2,u}+\mathrm{size}_2\mathrm{dis}_{3,v}+\mathrm{size}_2\mathrm{size}_3+\mathrm{size}_3\mathrm{dis}_{1,x}+\mathrm{size}_1\mathrm{dis}_{3,v}+[d(y,u)+2]\mathrm{size}_1\mathrm{size}_3$$

其中 x,u 应为树 1 和树 3 中 dis 值最大的点。

对于树 2，当确定 y 后，我们只要求出 $\mathrm{size}_3\mathrm{dis}_{2,u}+\mathrm{size}_1\mathrm{size}_3 d(y,u)$ 的最大值，便可得到答案，而这个也是可以通过 dp 在 $O(n)$ 的时间内求出的。

子任务二：

经分析观察，我们可以得到结论：对于任意的连通块，只要连通块中黑点的个数为偶数，就一定可以通过删去若干条边使得连通块满足度数的限制，否则一定无解。

证明：

先证明后无解情况：如果连通块中黑点的个数为奇数，由于任意连通块的度数和一定为偶数，因此必然无解。

对于结论的前面部分，我们使用归纳法：

1. 若树只包含 2 个黑点，此时我们可以只保留他们之间的边来满足度数要求。
2. 假设对于大小小于 k 的任意连通块结论成立，现在我们有一棵 k 个点的树，则有以下几种情况：
    1. 如果树中不存在白点，此时 k 必然是偶数。任选一个点 u 作为根，设点 u 有 p 个儿子，其中有 q 个儿子的子树大小为偶数。
        1. 若 $q>0$，可以删去与一个子树大小为偶数的儿子的连边，得到两个大小 $<k$ 的且含有偶数个黑点的连通块。
        2. 若 $q=0$，可知 p 为奇数，那么点 u 自身的度数限制已经满足。而每棵子树等价于，删掉与 u 的连边之后，将根的颜色取反。这样每棵子树都变成了含有偶数个黑点且大小 $<k$ 的连通块。
    2. 如果树中存在至少 1 个白点。我们任选一个白点 u 作为根，设且 u 有 p 个儿子，其中有 q 个儿子的子树中含有偶数个黑点。
        1. 若 $q>0$，可以删去与一个子树中含有偶数个黑点的儿子的连边，得到两个大小 $<k$ 的且含有偶数个黑点的连通块。
        2. 若 $q=0$，可知 p 为偶数，那么点 u 自身的度数限制已经满足。而每棵子树等价于，删掉与 u 的连边之后，将根的颜色取反。这样每棵子树都变成了含有偶数个黑点且大小 $<k$ 的连通块。

这样我们就可以得到一个算法：首先判断是否有解，然后枚举每条边，判断其是否可以被删去，即去掉该边后各个连通块中黑点个数的奇偶性不发生改变。由于边与边之间互不影响，故只需去掉所有可以被删去的边，便满足了字典序的要求。