1 拆分

1.1 对于部分分 $k=0$

相当于算整数拆分数，使用五边形数可以在 $O(n\sqrt{n})$ 或 $O(n\log n)$ 的时间内计算。

1.2 对于部分分 $1\le k\le 2, p=1$

众所周知，当 $k=1,p=1$ 时，答案相当于计算使用奇数拆分的方案数 n；当 $k=2,p=1$ 时，答案相当于计算使用模 5 余 1 或模 5 余 4 的数拆分的方案数。

以 $k=2,p=1$ 为例，考虑生成函数

$$\prod\frac{1}{(1-x^{5k+1})(1-x^{5k+4})}$$

可以发现对每一项取 ln 后的系数不为 0 的位置只有 $\frac{n}{5k+1}+\frac{n}{5k+4}$ 项，暴力计算求 ln 的结果后 exp 即可。

时间复杂度 $O(n\log n)$。

1.3 对于部分分 $k>0$

考虑枚举拆分出来的数的个数 m，那么这时的方案数就是用 m 个数拆分 $n-\frac{m(m-1)}{2}\times k$，也相当于用不超过 m 个数拆分 $n-\frac{m(m-1)k}{2}-m$。

将拆分方案对偶考虑，可以发现用不超过 m 个数拆分 n 的方案数就是用不超过 m 的数拆分 n 的方案数。

由于在 $k>0$ 时 m 是 $O(\sqrt{n})$ 级别的，在枚举 m 的时候用背包维护用不超过 m 的数拆分每个数的方案数即可。

时间复杂度 $O(n\sqrt{n})$。

1.4 标算

合并 $k>0$ 与 $k=0$ 的算法即可。

时间复杂度 $O(n\sqrt{n})$。

2 集合

2.1 算法 1（子任务 1,2）

注意到答案一定是整数，用 bitset 记下集合，暴力合并。

时间复杂度 $O(NQ\frac{M}{w})$。

2.2 算法 2（子任务 1,2,3）

优化算法 1，用线段树记录区间集合。

时间复杂度 $O((N+Q)\log N \times \frac{M}{w})$。

2.3 算法 3（子任务 1,2,3,5）

考虑线段树，用 vector 或数组暴力记录下区间集合并的结果。

建树是 $O(N\log N)$ 的，但是询问一次的最差复杂度是 $O(\min(N,M))$。

时间复杂度 $O(N\log N + Q\min(N,M))$。

2.4 算法 4（子任务 1,2,3,4,5）

考虑莫队，用线段树维护当前集合的交。

时间复杂度 $O(N\sqrt{Q}\log M + Q\log N)$。

2.5 标算

任意一个 S_i 集合都可以表示成 $A_i - B_i$ 的形式，其中 A_i 和 B_i 均为闭区间。

例如，$[2,4]\cup[8,9]$ 可以表示为 $[2,9]-[5,7]$；$[2,6]\cup[5,8]$ 可以表示为 $[2,8]-\emptyset$。

那么，(S_i 的交) 即为 (A_i 的交) 减去 (B_i 的并)。

显然，A_i 的交依然是一个闭区间。

接下来转化问题。考虑横轴为值域，竖轴为标号的二维平面。对于每个集合 i 画上竖轴标号为 i，横轴区间为 B_i 的横线表示限制。那么，一次询问 l,r 相当于求最小的横坐标，满足在 (A_i 的交) 的范围内且竖直方向上 $[l,r]$ 范围内没有线。

对于这个问题，可以对标号扫描线，用线段树维护解决。

时间复杂度 $O((N+Q)\log N)$

也有其他类似的一个 log 或两个 log 的做法。

3 网格

3.1 暴力

使用不同的暴力算法可以通过 5-40 不等的分数。

一种比较优秀的暴力为，每次枚举可填颜色最少的位置的颜色，在此基础上加上剪枝或随机，可以获得不错的分数。

3.2 对于部分分 $m = n$

直接构造即可，$coler_{i,j} = (j - i) \bmod n + 1$

3.3 假算法？

每次枚举一种颜色，提取可以填这种颜色的位置，跑最大匹配求出填最多位置的方案，填好这些位置，然后枚举其他颜色。

该做法没有正确性保证，在随机数据下有一定错误概率。

加上随机化，如果最终没有填满就再跑一次，这样该算法就难以被 hack。

时间复杂度 $O(mn^{2.5})$，拼上部分分可以得到 70 分。

3.4 标算

给每个位置赋权值 $val_{i,j} = (j - i) \bmod n + 1$。

每次枚举一种颜色，提取可以填这种颜色的位置，跑稳定匹配，填好方案中的位置，然后枚举其他颜色。

在稳定匹配中，对于行 i，$val_{i,j}$ 越大的列 j 优先度越高，对于列 j，$val_{i,j}$ 越小的列 i 优先度越高。

不难证明，该算法一定可以构造出一组解。

时间复杂度 $O(mn^2)$