

1、石子合并(stone)

考虑两堆相邻但奇妙值符号相反的石子(0既可以视为正也可以视为负),无论合并时是谁减谁,新石子堆的奇妙值绝对值都等于原石子堆奇妙值绝对值之和,而谁减谁可以决定新石子堆的符号,那就不难构造一种顺序,使得每次合并都是在操作两堆奇妙值符号相反的石子,而最后一次合并奇妙值一定是一正一负,此时用正奇妙值减负奇妙值,那么最终答案就是所有奇妙值的绝对值之和.

注意一些特殊情况的处理,例如奇妙值全为正或全为负的时候,必须牺牲一部分绝对值之和,先选择一个两堆石子合并来得到一个不同的符号,此外 $n=1$ 的时候又不适用于全正全负的处理,都是需要注意的点.

2、铺地毯(carpet)

每个地毯可以看作一个两边均平行于坐标轴的矩形,两个这样的矩形相交得到的也是有同样性质的矩形,那么归纳可得任意个这样的矩形相交都仍然得到平行于坐标轴的矩形,但是矩形的交是不可逆的,也即我们不能先将所有矩形交起来再去掉其中一个来得到 $n-1$ 个矩形的交.

那么我们可以计算矩形序列的前缀交和后缀交(这些交都是矩形),对于第 i 个矩形,除了该矩形外其它所有矩形的交即为前 $i-1$ 个矩形的前缀交再交上后 $n-i$ 个矩形的后缀交,也即每 $n-1$ 个矩形的交都可以这样计算出来,而这些交会重复的部分一定是 n 个矩形的交,

所以单独进行去重即可.

3、优美的旋律(melody)

这个问题的解法很多, 这里介绍处理起来最简单的一种.

想判断一个字符串是否是由某个子串循环得到的, 只需利用KMP求出其next后判断 $n - \text{next}[n]$ 能否整除 n 即可, 若能则说明最小的循环节长度即为 $n - \text{next}[n]$, 枚举循环次数 $\frac{n}{n - \text{next}[n]}$ 的约数即可得到所有的循环节, 再计算优美度的最大值即可.

对于原问题, 从每个位置开始单独计算一次next数组就可以得到所有子串的循环节了, 直接枚举约数总复杂度是 $O(n^2\sqrt{n})$, 这里可以预处理所有数的倍数, 将复杂度优化为 $O(n^2\log n)$.

4、基站建设(station)

这个问题中所求解的值公式其实没什么特点, 所以我们应该考虑去枚举满足条件的子图, 直接枚举并利用Hash表判断边是否存在的复杂度为 $O(n^4)$, 但事实上我们可以观察得知满足条件的子图其实可以总结为共享一条边的两个三元环, 那么我们可以枚举所有三元环, 再记录每条边在哪个三元环上, 复杂度可以降低为 $O(n^3)$.

如何更快地枚举三元环呢? 枚举三元环既可以枚举三个顶点, 也可以枚举三条边, 我们注意到两种方法对于度数大小不同的顶点优势区间是不同的, 那么可以考虑根号分治, 对于度数小于 \sqrt{m} 的点, 我们枚举其两条出边再判断两条边到达的点之间有没有边, 而度数大于

\sqrt{m} 的点总个数不超过 $O(\sqrt{m})$ 个，直接枚举三个点即可，总时间复杂度为 $O(m\sqrt{m})$ 。