模拟赛题解

花花

RDFZ

September 17, 2023

富贵竹

显然整个路径形如，某个时刻碰到一个角落，然后原路返回。

因此只用考虑第一次碰到角的时刻。

速度方向改变不好做，考虑碰壁的时候翻转矩形。这样的话就变成，整个平面铺满了 $n \times m$ 的矩形，然后从 $(0,0)$ 开始一直走。会碰到竖线的时刻是所有 $cur \bmod n = 0$ 的时候，碰到横线的时候是 $cur \bmod m = 0$ 的时候，第一次同时碰到显然是 $\mathrm{lcm}(n, m)$。

时间复杂度 $O(q \log n)$。

柳穿鱼

如果不是平方和，而只用计算和？

考虑对于每个连通块拎出来一个 dfs 树，任意的决定所有非树边是黑的还是白的，所有树边的状态就可以唯一确定了。确定方式就是从下往上构造，如果这个点度数是奇数，就把这个点和父亲的边的然黑，否则白。注意到根的度数一定是偶数。

因此方案数是 2^{m-n+c}，其中 c 是连通块个数。

柳穿鱼

平方和可以枚举两个边算一下贡献。计算这两个为黑色有多少个染色方案是好的。

case1：这两个边存在一个割边。假设第一个边是割边，图分成了两个连通块，考虑没有第二个边的连通块。这个连通块恰有一个点度数增加了 1，因此整个连通块的度数一定是奇数，所以一定无解。

case2：这两个边删掉之后，整个图的连通块个数增加了 1，此时方案数是 $2^{m-n+c-1}$。

case3：这两个边删掉之后，整个图的连通块个数不变，方案数是 $2^{m-n+c-2}$。

柳穿鱼

因此我们只需要分别统计三类点对的个数。其中一个方法是：考虑 dfs 树的每个非树边，给这个非树边和对应的树链上的边同时 xor 一个随机数。

所有权值为 0 的边是割边，权值相同的边属于 case2，剩余的属于 case3。正确性证明可以百度 DZY loves Chinese II。

用哈希表做到时间复杂度 $O(n)$。

雁河菊

题意是要求你最后所有数字覆盖了 1 到 max 的所有整数。

假设确定了最终的数字可重集合，为了最小化代价，一定是原本第 k 小的匹配最终第 k 小的。

于是可以考虑一个 dp，$dp_{i,j}$ 表示考虑了最小的 i 个数字，目前所有数字覆盖了 $[1,j]$，最小代价是多少。

考虑新加入一个数字 x，首先 x 最终的位置必须为 j 或 $j+1$。不能小于 j 是因为要保证大于等于之前的；不能大于 $j+1$ 是因为如果大于了，那么以后都会大于，就没有人 $j+1$ 了。具体来说我们分三类情况讨论：

雁河菊

1. x < j，由于之前的都已经到 j 了，所以如果你只向右移动到 j 是没意义的，因为你上一个完全没必要移动到 j。所以只有一个选择就是更新到 $dp_{i+1,j+1}$。
2. x = j，两个选择就是更新到 $dp_{i+1,j}$ 或 $dp_{i+1,j+1}$。
3. x > j，显然只会更新到 $dp_{i+1,j+1}$。

观察上述转移，只有唯一的情况是会转移到 j 的。因此我们需要支持四种操作：

1. 全局平移 1
2. 全局加绝对值函数
3. 单点查询
4. 单点修改

雁河菊

为了避免全局平移 1，我们可以不执行这个操作，而是对所有后续操作打一个偏移标记。绝对值函数可以拆成前缀后缀一次函数，三个操作可以用线段树简单实现。

但是注意到关键性质，所有操作的分界点（前后缀的分界点，查询，修改的位置），总移动量是 $O(n+m)$ 的。

因为有 $a_i >= a_{i-1}$，分界点在 $a_i - i$ 附近，$-i$ 是因为偏移量。

所以考虑维护一个指针 p，在 p 的右侧，每个点维护一个后缀加一次函数标记，左侧维护前缀加一次函数标记。p 可以 $O(1)$ 的向左或者向右移动。因此总时间复杂度为 $O(n+m)$。

雁河菊 - Sol2 by monstersqwq

先对原序列排序，设 $b_i = i - a_i$，结束条件转化为 $b_{i-1} \le b_i$，这是经典的 slope trick 模型，可以使用堆解决。

注意到由于 a_i 不降，有 $b_i \le b_{i-1} + 1$，因此堆中的最大值每次至多增大 1，可以维护一个桶和一个最大值指针做到线性。

樱草花

先套个盾，本体只需要用到哈希，std 使用的 SA 可以替代。就算 SA 也在提高组考纲里。以及这个题时间复杂度是 $O((n+q)\log n)$。

樱草花

考虑基本子串字典相关理论。

若 p, q 是字符串 w 的周期且 $p+q \leq |w| + \gcd(p, q)$，那么 $\gcd(p, q)$ 是字符串 w 的周期，下文会反复用到这个定理。

若 $2|u| = |w|$，u 在 w 里的出现位置形成等差数列。特殊的若，出现至少三次，那么公差为 u 的最小正周期。用上文证明即可。

樱草花

考虑求出 l_1 的前缀等于 r_2 的的后缀有哪些。每次考虑求出 $[2^k, 2^{k+1})$ 内的所有合法长度。于是可以找到 $(r_2 - 2^{k+1}, r_2]$ 内 $[l_1, l_1 + 2^k)$ 所有出现位置，这个形成了一个等差数列记为 a。同理求出 $(r_2 - 2^k, r_2]$ 在 $[l_1, l_1 + 2^{k+1})$ 内的出现位置形成的等差数列，记录为 b。

所有的合法长度为 a 与 b 的交。当 a 或 b 只有不超过两项可以暴力。否则的化，注意到 a 中任意一个长度和 b 中任意一个长度代表的字符串都有交，a, b 中最长字符串的交，可以得到这个字符串的周期为 a, b 公差的 gcd。也就证明了 a, b 公差此时一定相等。也就是做到了 $O(1)$ 等差数列求交。

更多内容可以搜索基本子串字典。

樱草花

用上文方法我们可以用 $\log$ 段等差数列表示出合法的 u, w。

先暴力枚举两段等差数列。记为 a，b，其中 $a_1, a_2, a_3 \cdots$ 表示 a 的最大项，次大项等等。令 $len = r_1 - l_1 + 1$，那么 $a_1 + b_1 \le len$ 的情况是简单的。具体来说，你可以先检验中间的 v 是否 reverse 之后是另一个区间的中间。剩余 a, b 将彻底独立，不妨只考虑 a。a_2 合法的条件是：a_1 长度为 $a_1 - a_2$ 的前桌的 reverse 和后缀相等。类似于回文串的查询，为了方便可以直接使用 SA 求 lcp，实际上分析周期可以得到一个哈希做法。求出最长的 lcp 之后除以公差即可得到合法的 a 的数量，乘以合法 b 的数量就是贡献。

樱草花

　　第二种情况比较麻烦，首先我们先排除一些平凡的情况，比如：a 或 b 只有不超过两项，或者 $a_3 + b_1 \le len$ 或者 $a_1 + b_3 \le len$，也就是暴力做两项能转化为第一种情况的。

　　剩余的情况有性质 a, b 的公差一样，并且是整个串周期。

　　具体证明大概是：若 a 的公差大于 b 的公差，考虑 $a_{1,2,3}$ 与 b_1 相交，考虑 a_1 交 b_1，这个串显然有 a, b 两者的周期，且长度大于等于两倍 a 的周期，也就是说名他有 $\gcd$ a, b 的周期。由于交的部分占用了完整的 a 的周期，所以 a 整个串的周期也可以变成 $\gcd$，反之同理。

　　于是只需要判断长度为极小的区间作为 v 是否回文，如果回文同样长度的区间都合法。

　　然后判断周期是否回文，因为有回文串的 border 充要条件是回文前缀的定理。所以所有的 $a_i + b_j \le len$ 都会合法。

樱草花

上述分类讨论都可以 $O(1)$ 做到。

于是获得了一个 $O(\log^2 n)$ 单词询问的做法。

回顾我们基本子串字典的等差数列划分方法，我们可以注意到：存在一项大于 $\frac{len}{2}$ 的等差数列不超过两个。

两个 $\leq \frac{len}{2}$ 的等差数列必定会进第一种简单的情况。因此考虑统一的处理。

首先判定最长的不超过 $\frac{len}{2}$ 的那个等差数列的最大项。如果 v 合法，那么考虑计算 a 的最长的前缀 reverse 等于后缀，对于 $\log$ 个等差数列统计一下等差数列交上这个长度大于等于多少的限制，还留存了几项。

a, b 分别统计最后乘起来即可。

剩余可能产生贡献的等差数列对数不超过 $O(\log n)$，可以采用上文做法。

总之时间复杂度 $O((n+q)\log n)$。

樱草花

一些实现细节

但事实上如果你写网上流行的哈希表在线基本子串字典可能会有点卡常。

原因是这部分 1log 跑的比后续统计答案 2log 还慢。

std 采用的实现方法是离线下来询问。枚举 2^k，扫描线序列统计每种长度为 2^k 的字符串在 $[i, i+2^{k+1})$ 的区间理出现的等差数列。需要支持等差数列 pushabck 和 popfront。总体来说非常快，甚至能跑 $5 \cdot 10^5$。

所有分类讨论可以以最暴力的方法执行，因为 std 也是这么写的。

樱草花

谢谢大家！