CCPC Final 2024 题解

CCPC Final 2024 裁判组

May 24, 2025

Statements

给定一个字符串，定义它的一个子串是好的，当且仅当可以将原串划分成若干个该子串的前缀。

对于输入的字符串求出它的好的子串的数量，相等但位置不同的子串算多个。

$1 \le n \le 10^5, \sum n \le 2 \times 10^5$

Author: apiad; Solution: Kubic; Prepared by Kevin114514

Solution

一个子串是好的，当且仅当它和它的前缀在原串中出现的位置覆盖了所有下标。

对原串建出后缀树，在后缀树上 DFS，并且实时维护当前节点到根路径上的字符串覆盖的位置。

具体地，我们维护当前子树内的所有 endpos，并且维护相邻两个 endpos 之间的最靠前的还没有被覆盖的位置，以及它们到下一个 endpos 的距离。

Solution

进入一个节点的时候，我们首先把所有距离不超过当前节点的长度的位置覆盖。这一步可以每次暴力从线段树上找到距离最小的位置并且覆盖它。

向子节点 DFS 的时候，可以使用线段树分裂的技巧将维护 endpos 的线段树分成若干棵，传给子节点。

可以证明这样做的时间复杂度是 $O(n \log^2 n)$ 的。

Solution

线段树分裂的过程本质上是线段树合并的反向过程，这部分的复杂度为 $O(n \log n)$。

每次暴力在线段树上找到一个要覆盖的位置的复杂度为 $O(\log n)$，只需要证明这样的操作次数是 $O(n \log n)$ 级别的即可。

定义势能为线段树上还未被覆盖的位置数量。初始的势能不超过 n，每次分裂时，若相邻两段被分到不同子树，则会至多使势能增加 1，分到相同子树则不会增加势能。每次覆盖一个位置会让势能减少 1。

由于分到不同子树的相邻段数不会超过每个节点轻子树的大小和，故势能的增长量为 $O(n \log n)$ 级别，暴力覆盖的次数同样是 $O(n \log n)$ 级别。

Statements

给你一个序列 $a_1, a_2, \ldots, a_n$。q 个询问，你可以给每个位置加一，把区间 $[l, r]$ 变成唯一的最大子段和需要的步数。

$1 \leq n \leq 2 \times 10^5$

Author: Kevin114514 & apiad; Prepared by apiad

Solution

首先可以特判一些平凡的情况。如果这个序列往两边延伸，存在一个非负的后缀或者前缀，那么一定是不合法的。

对于剩下的情况，一定可以让它变得合法，因为你可以把每个数加到足够大。

假设初始序列的最大子段和为 M。首先我们证明，如果不考虑唯一的限制，那么答案为 $M - \sum_{i=L}^{R} a_i$。

Solution

不妨设存在一个区间 $[l, r]$ $(L \le l \le r \le R)$ 使得 $\sum_{i=l}^{r} a_i = M$。

- 因为往外延伸的和都是负的，所以不存在一个 MSS 与 $[l, r]$ 相交，只可能是包含或者在 $[l, r]$ 外部。
- 如果 MSS 不在内部，那么可以先把这个区间变成当前区间的 MSS，然后加两端，变成全局 MSS 即可。
- 因为两端的和都是负数，出现在区间之外的 MSS 不会对这段是否成为 MSS 造成影响。

Solution

固定区间内 $[l, r]$ 的数字不动，并且在前缀和序列上固定 $s_l = 0, s_r = M$，增加 l 前面的数相当于前缀和减一，增加后面的数相当于加一。

只需要从后往前依次把 s 往下调，从前往后依次把 s 往上调，所有数在 $[0, M]$ 之间，然后把 s_l 变成 0，s_r 变成 M 即可。

Solution

如果是唯一的，需要一些额外的分类讨论。令 $S = \sum_{i=L}^{R} a_i$。具体的：

- 如果存在 $L < l \le r < R$，并且 $\sum_{i=l}^{r} a_i = M$，需要把 s_l 调整成为 -1，把 s_r 调整成为 $M+1$，答案为 $M+2-S$。
- 如果存在 $L = l \le r < R$ 或 $L < l \le r = R$，并且 $\sum_{i=l}^{r} a_i = M$，那么只需要调大一个端点，答案为 $M+1-S$。
- 如果外部有 MSS，那么这段需要比 M 大，答案也为 $M+1-S$。
- 这段就是原序列的 MSS，答案为 $M-S$。
- 注意特判全都是负数或者单个数的情况。

Solution

可以用线段树查询区间 MSS，做到 $O((n+q)\log n)$。

也可以记录一下每个端点最近的 MSS，做到 $O(n+q)$ 的时间复杂度。

实际上也可以通过把 a_l 和 a_r 减一之后查询 MSS，做到类似的效果，并且避免一些分类讨论。

Statements

给一个图，初始每条边有一种颜色。

你可以做不超过 n 次操作，每次把选择一个点 x，把它的所有相邻边变成 y。问是否可行，并构造方案。

$n, m \leq 2 \times 10^5$。

Author: Milmon; Prepared by Milmon

Solution

考虑倒着操作，如果一个点的邻居的所有边都是同色的，那么之前可以是任意颜色的。

也就是可以把这些边的颜色变成空，继续对所有邻居边都是同色的或者空的点做这个操作。

如果无法继续进行操作，对比一下剩下的边和初始边的颜色即可。

类似拓扑排序，可以使用 map 或者 Hash 表维护一下一个点周围的不同颜色。

时间复杂度 $O(n+m)$ 或者 $O((n+m)\log n)$。

Statements

给你一个排列，问 $i < j < k < l$ 并且满足 $p_i \cdot p_k = p_j \cdot p_l$ 的对数。

$1 \leq n \leq 5 \times 10^4$

Author: apiad; Improved by zhoukangyang; Prepared by apiad

Solution

记 q_i 表示 i 所在的位置。

问题转化为 $p_i/p_j = p_k/p_l$，令 $p_i/p_j = a/b$，其中 $(a, b) = 1$。

那么这四个数可以被表示成 ac, bc, bd, ad，问题等价于统计多少对 a, b, c, d，满足 $(a, b) = 1$ 并且 $q_{ac} < q_{bc} < q_{bd} < q_{ad}$。

考虑对 $(a, b) = 1$ 反演，只需要对于每个 $x(1 \le x \le n)$，统计 x 倍数中，满足上述条件的即可。

Solution

根据反演，我们去掉了 $(a, b) = 1$ 的条件，只需要统计 a, b, c, d，并且 $q_{ac} < q_{bc} < q_{bd} < q_{ad}$ 的方案数。根据 $\max(a, b)$ 进行讨论。

- 如果 $\max(a, b) \le T$，那么只要预处理出 q_{bx} 的顺序，然后线性扫描一遍即可，时间复杂度为 $O(n/\max(a, b))$。假设 $\max(a, b) = M$，这部分时间复杂度为 $O(\sum_{M=1}^{T} M \cdot \frac{n}{M}) = O(nT)$。
- 否则会有 $\max(c, d) \le \frac{n}{T}$，考虑数对 (q_{xc}, q_{xd})，需要求出互相嵌套的对数，这部分需要数点，时间复杂度为 $O(n \log n/\max(c, d))$ 这部分时间复杂度为 $O(\sum_{M=1}^{n/T} M \cdot \frac{n \log n}{M}) = O(\frac{n^2 \log n}{T})$。

Solution

平衡两部分的时间复杂度，解决单个 n 的时间复杂度为 $O(n\sqrt{n\log n})$。

根据反演，对于每个 $x(1 \le x \le n)$，解决规模为 n/x 的问题，总的时间复杂度为 $O(\sum_{x=1}^{n}\left(\frac{n}{x}\right)^{1.5}\sqrt{\log n}) = O(n\sqrt{n\log n})$。

实现的比较好的 $O(n\sqrt{n}\log n)$ 也可以通过这个题。

Statements

平面上给你若干个矩形，问覆盖次数是 $m, 2m, \ldots$ 的面积。
$1 \le n \le 3 \times 10^5, m \ge \sqrt{n}$。
Author: nzhtl1477; Prepared by nzhtl1477 & ccz181078

Solution

考虑扫描线，记录一下当前每个位置的被覆盖次数。只需要支持区间加一，区间减一，查询所有 m 的倍数出现次数即可，

记 a_i 表示第 i 个位置的覆盖次数。对于序列分块，每块记录一下 max 到 min 之间的每个数的出现次数，然后在每块内暴力查询即可。

因为 $\sum |a_i - a_{i+1}| \leq 2n$，所以每一块的 $\max - \min$ 的和不超过 $O(n)$，每块查询可以做到 $O(\frac{\max - \min}{m})$，所以一次查询的时间复杂度不超过 $O(n/m)$。

总的时间复杂度 $O(n\sqrt{n} + n^2/m)$。

注意，如果每块重构时如果是 $O(\max - \min)$ 的，那么复杂度会退化成 $O(n^2)$。

Statements

数轴上 1 到 n 的位置有磁铁。每次可以选择一个磁铁操作，这个磁铁会消失，两边的磁铁会往外移动一格。你想进行恰好 k 次操作，并且使得最终磁铁仍然在 $[1, n]$ 之间。

给两个带 ? 的串 S 和 T，求有多少种替换 ? 的方式可以使得 S 和 T 可能是合法的初始状态和最终状态。

$n \leq 5000$。

Author: Kevin114514; Prepared by Kevin114514

Solution

考虑确定被拿走的磁铁的集合，此时没有被拿走的磁铁移动的方向和距离可以由它左侧被拿走的磁铁的数量确定。

容易说明拿走磁铁的顺序不影响实验过程，且两个不同的被拿走的磁铁集合不会对应相同的 (S, T)。

考虑 DP 状态 $f(i, j)$ 表示，确定了 S 的前 $i-1$ 位和 T 的前 $j-1$ 位，且若 S_i 是一个被保留的磁铁，则它最终的位置为 j。

Solution

若 S_i 是 0，则从 $f(i,j)$ 转移到 $f(i+1,j+1)$。
若 S_i 是 1 且被保留，则从 $f(i,j)$ 转移到 $f(i+1,j+1)$。
若 S_i 是 1 且不被保留，则从 $f(i,j)$ 转移到 $f(i+1,j+3)$。
初值为 $f(1,1-k)=1$，答案为 $f(n+1,n+k+1)$。
时间复杂度 $O(n^2)$。

Statements

给你一个正整数序列，划分成若干段，使得每段的乘积之和尽量大。

$n \leq 2 \times 10^5$。

Author: Gellyfish; Prepared by Gellyfish

Solution

首先如果总乘积超过 $2n$，那么把所有大于 1 的数字分成一段一定最优。因为分两段，和至多增加 $n-2$，而乘积会至少会减少 $n-2$。

对于剩下的数字，至多只有 $\log n + O(1)$ 个数不等于 1，可以直接枚举或者 dp 求出这些数的划分方式。

时间复杂度 $O(n)$。

Statements

在所有 m 个 1 长度为 n 的 01 串中，选 k 个，使得每一位 1 个数的最大值尽量小。构造方案。

$n \leq 20$。

Author: apiad; Prepared by apiad

Solution

答案下界显然为 $\lceil \frac{mk}{n} \rceil$。

本题有多种构造方法。

其中一种方法把所有循环同构的串分成一组，每次填入一组，所有位上的 1 的个数增量相同。

对于不能用整组填充的部分，我们可以一开始把 $1\ldots10\ldots0$ 的循环同构组保留下来用来填充多出来的部分。

依次用这个串以及它循环移位 m 位的结果填充。这样能保证每一位上面的 1 依次增加。

比如 $n = 6, m = 4$，那么可以用 $111100, 110011, 001111, 011110, 111001, 100111$ 依次填充。

Statements

给定一张简单图，请求出有多少种删去图中两个点的方式，使得得到的图中无环。

$3 \leq n \leq 5 \times 10^5, 0 \leq m \leq 5 \times 10^5$。

Author: Kevin114514; Prepared by Kevin114514

Solution

首先提取出图中所有包含环的点双联通分量。后文中点双特指包含环的点双。

若图中有 ≥ 3 个点双，则可以找到一个点双，使得其与至多 1 个其他点双相交。则要么选中其与另一个点双的交点，要么选中剩余所有点双的交点。

若图中有 2 个点双，则要么选中它们的交点，要么两个点双中分别各包含一个选中的点，转化为单个点双的情形。

若图中有 0 个点双，则答案显然为 $\binom{n}{2}$。

Solution

若图中有 1 个点双，只需要统计其中恰好有 1 和 2 个选中的点的方案数即可。

对于恰好选中 1 个点的情形，可以暴力枚举这个点计算答案。

对于恰好选择 2 个点的情形，首先找到点双中的一个环，特判点双是单个环的情形，这种情形的答案计算是简单的。

若点双不是单个环，则必然可以找到环上某个点 x，和一条从 x 开始，经过若干个相同点双但不在环中的点，回到一个环上的点 y 的路径，且 $x \neq y$。

Solution

此时，若点双中仅有找到的环和这条 $x - y$ 的路径，没有其他边存在，即这个点双是一个 3 条路径组成的杏仁图。这种情况可以使用简单的组合计数求出答案。

否则，类似地，可以找到杏仁图上某个点 z，和另一条从 z 开始，经过若干个相同点双但不在杏仁图中的点，回到一个杏仁图上点 w 的路径，且 $z \neq w$。注意这里可能有 $x = z$ 等情形，只保证 $z \neq w$。

此时可以发现，若 x, y, z, w 四个点（可能有重复）均没有被选中，图中一定存在一个环。

Solution

综上，除去两类可以组合计数的情形之外，我们均可以找到 $O(1)$ 个点满足其中必有至少一个点被选中，此时可以枚举这些点，转化为删去 1 个点的情形。

对于删去 1 个点的情形，可以对每个点统计，删去它之后图中剩余的边数 m' 和连通块数 c，若 $n - m' = c$ 则该点符合条件。

时间复杂度 $O(n)$。

Statements

问 n 个点的有标号 d-正则图个数模 p 的结果。T 组询问。

$0 \leq d < n \leq 10^{18}$, $T \leq 10^4$, $p \in \{2, 3, 5, 7\}$。

Author: apiad & JohnVictor; Improved by zhoukangyang; Prepared by apiad & zhoukangyang

Solution

考虑每 p 个元素分一组，这样会分出若干组以及 $[0, p)$ 个剩余元素。

对于每个组，考虑将其循环位移。如果位移了一位后，图变得不同了，那么我们就可以将位移了 $[0, p)$ 位后的图分为一组，这样分出的组大小都为 p。因此，我们只需要考虑位移一位后不同的图。

这就意味着组的每个点向组外连的边都是相同的。对于组内的点，需要满足对于每个 $1 \leq d \leq \lfloor \frac{p}{2} \rfloor$，所有 $(i, (i + d) \bmod p)$ 的边的存在性都相同。

Solution

接下来考虑再往上继续对这些大小为 p 的组来分组。我们可以再次做类似的循环同构操作。此时，我们要求组内 p^2 个点往外连的边都相同；这 p 个组内所连的边都相同；对于 p 个组间的边，同样要求差相同的点之间连边状态相同。

类似地，我们可以向上继续分组。最终，设 $n = \sum_{i \geq 0} a_i p^i$，$0 \leq a_i < p$，我们会有 a_i 个大小为 p^i 的组。

Solution

首先我们先观察组内的连接情况。我们的限制是，每一次向上分组中，不同组之间所构建的图都必须完全相同。假设构建的图满足每个点度数均为 d $(0 \le d < p)$，那么第 i 次分组会对每个点的度数贡献 $d \times p^{i-1}$。

这就意味着，假设满足差相同时连边状态相同、且每个点度数均为 d 的图的个数为 f_d，那么如果我们希望组内每个点度数为 $m = \sum_{i \ge 0} b_i p^i (0 \le b_i < p)$，方案数为 $\prod f_{b_i}$。

Solution

接下来我们考虑组之间的点的连边。根据之前循环同构带来的限制，两个组之间要么连满了边，要么没有连边。

对于一个大小为 p^i 的组内的点，我们可以发现，所在组大小小于等于 p^i 的点和它的连边产生的度数贡献小于 p^{i+1}，因此大小大于 p^i 的组产生的总贡献为 $p^{i+1}\lfloor\frac{n}{p^{i+1}}\rfloor$。而所有大小为 $p^j(j>i)$ 的组产生的贡献一定只会影响总度数的第 j 位，所以我们可以用这个来确定每个大小为 p^i 的组会连多少个大小为 p^j 的组。

Solution

接下来，我们再对于大小为 p^i 的那些组，分析大小 $\leq p^i$ 的所有组和这些组之间的连边。

我们先按 j 从小到大考虑 $j < i$ 的大小为 p^j 的组的连边。我们可以记录每个大小为 i 的点当前的度数，一个 p^j 大小的贡献就是给其中若干个点的贡献 p^j 的度数。接下来，我们考虑每个这些大小为 p^i 的组里，每个点度数的第 j 位。做完操作后，每个点的度数在 $[0, 2p^{j+1})$ 内。而我们后面影响的度数都是 p^{j+1} 的倍数，所以我们实际上只用记录一个 01 串。由于这些大小为 p^i 的组本质相同，我们可以只记录 1 的个数。

Solution

上面的过程只用考虑 n 的第 p^i 位，m 的第 p^i 位，n 的第 p^j 位，m 的第 p^j 位，开始时 1 的个数、出去时 1 的个数。所以我们只需要预处理这些。预处理是可以 $4^p \operatorname{poly}(p)$ 的。

预处理复杂度 $\mathcal{O}(4^p \operatorname{poly}(p))$，查询复杂度 $\mathcal{O}((p \log_p n)^2)$，std 可以通过通过 $p = 11$ 的数据。

Statements

给一个图，划分成两部分，使得两部分都连通并且点数模 3 余 2，判断是否可行。

$n \leq 5 \times 10^5, m \leq 5 \times 10^5$。

Author: apiad & Lynkcat ; Prepared by apiad

Solution

首先特判一下图不连通的平凡情况。

如果图连通，划分成两个连通块，一定会从一个点双连通分量划分开来。

建立圆方树，求出点双连通分量中，每个点在去掉这个点双之后的连通块大小（即圆方树上这个点对应的子树大小）模 3 的余数。

如果一个点双中有一个 2，那么把这个点单独切出来即可，剩下的仍然连通。

否则这个点双中只有 $0, 1$，如果只有一个 1，那么这个点双不可行。否则至少 4 个 1。因为是点双，所以可以双极定向，一定能划分成两个连通的部分，并且其中一个部分只有两个 1。

时间复杂度 $O(n+m)$。

Statements

交互题。你一棵形态和边权都随机的树。你可以查询两个点的距离，请你用不超过 $7n$ 次询问复原这棵树。

$n \leq 10^5$。

Author: rddccd; Prepared by rddccd

Solution

我们可以任选一个点 u，求出它到其他所有点的距离。然后继续选择另一个距离 u 最远的点 v，求出它到其他所有点的距离。

对于一个点 w，如果 $d(u,w)+d(w,v)=d(u,v)$，那么 w 在 u,v 的路径上。否则可以根据 $(d(u,w)+d(w,v)-d(u,v))/2$ 来算出 w 在哪个子树里。

对每个子树递归查询即可。对每个子树，根据之前的查询结果，你可以得到根节点到其他所有点的距离，可以省下一半常数。

因为随机生成的树的直径是 $\Theta(\sqrt{n})$ 的，所以任选一个点后，离它最远点的距离也是 $\Theta(\sqrt{n})$ 的。这个做法的期望查询次数毛估估是 $O(n \log\log n)$ 的。

Statements

有 n 个点，其中第 i 个点一步能走到 $[a_i, b_i]$，保证 $a_i \leq i \leq b_i$。

问从某个起点出发，走若干步回到这个点，中间经过的点集可能的方案数。

$1 \leq n \leq 50$

Author: Kubic; Solution: zhoukangyang; Prepared by Kubic

Solution

相当于求有多少个点导出子图强连通。

可以证明，一个子集 S 强连通当且仅当它们可以通过如下过程合并为一块：

- 初始 S 中每个点独成一块。
- 若相邻两块相互可一步到达，则将它们合并。

Solution

故令 $f_{l,r,u,v}$ 表示满足如下条件的子集 S 的个数：

- $l, r \in S, S \subseteq [l, r]$。
- S 对应导出子图强连通。
- $u = \min\limits_{i \in S} a_i, v = \max\limits_{i \in S} b_i$。

如果不合法，一定是若干个不能合并的块，可以用辅助数组求出若干个不可合并的块拼在一起的结果。

然后用总方案减去不合法的方案，可以得到上述的 dp 结果。时间复杂度视实现可能为 $O(n^6) \sim O(n^8)$。