部分分训练指导

ddd

2018 年 10 月 5 日

本节课内容

解决一道题目时，依照不同的部分分写不同复杂度的算法是很简单的，这只是一个经验问题。今天要讲的是如何根据部分分（特殊情况），找到突破口，进而解决全部问题。

一般来讲，典型的部分分（特殊情况）的转化有：“n, m” ->“$n = m$”、“树” -> “链”、“图” -> “ 树”、“ 动态” -> “ 静态”、“在线” -> “离线” 等。

例题 1

给出一棵 n 个点的树，问树上有多少条路径，满足编号的范围是连续的。

- $1 \le n \le 3 \times 10^5$。

题解

直接考虑树上情况不方便，先考虑如果是一条链怎么办：假设这条链上第 i 个点的编号是 a_i，编号为 i 的点位于 b_i 处，也就是说 $a_{b_i} = i$。

编号范围连续可以有几种等价的表示，下面给出一种最简单的表示。

设 $f(l, r)$ 表示 $[l, r]$ 这段区间的长度减去这段区间中形如 $(i, i+1)$ 这种点对的出现次数，那么当且仅当 $f(l, r) = 0$ 时，$[l, r]$ 是一条合法的路径。

建立一个二维坐标系，横轴表示 l，纵轴表示 r，(l, r) 处表示 $f(l, r)$。那么只需要统计这个坐标系中 $1 \le l, r \le n$ 这 n^2 个点中有多少个 0 即可。

题解

接下来考虑如何快速得出所有点的 f 值。

对于 (a_i, a_{i+1}) 这个点对，其会对满足 $1 \le l \le i, i+1 \le r \le n$ 的点的 f 值产生 1 的贡献。

对于 $(i, i+1)$ 这个点对，其会对所有满足 $1 \le l \le \min(b_i, b_{i+1}), \max(b_i, b_{i+1}) \le r \le n$ 的点的 f 值产生 -1 的贡献。

这些贡献都是一块矩形区域，扫描线 $+$ 线段树维护即可。时间复杂度 $O(n \log n)$。

题解

接下来考虑如何推广到树上。

对于一条边 (u, v)，不妨设 v 是 u 的父亲，那么其会一个端点在 u 的子树内，一个端点在 v 的子树外（包括 v）的所有路径产生 1 的贡献。由于是对子树的贡献，容易发现如果用 dfs 序来表示，贡献也是若干个矩形。

对于一个点对 $(i, i+1)$，经过讨论也可以发现贡献也是 dfs 序上的矩形。

和链上维护方式相同，时间复杂度 $O(n \log n)$。

例题 2

Alice 和 Bob 在玩游戏，开始时有 n 堆石子，第 i 堆石子有 a_i 个，Alice 和 Bob 轮流取石子。Alice 每次必须恰好取 x 个，Bob 每次必须恰好取 y 个。谁不能取了谁输，Alice 先手，两人都以最优策略取，问谁会赢。

- $1 \le n \le 10^5$。
- $1 \le a_i, x, y \le 10^9$。

题解

首先考虑 $x = y$ 的情况：

对于第 i 堆石子，可以被取 $[a_i/x]$ 次。所有石子可以被取 $\sum_{i=1}^{n}[a_i/x]$ 次，只需要判断这个数的奇偶性即可。

更进一步，我们发现对于一堆石子 a_i，其加上 $k(x+y)$ 或减去 $k(x+y)$ 都不影响答案，即如果 $a_i = a_i \mod (x+y)$，答案不变。

经过归纳发现这个结论在 $n \neq m$ 时也成立。

题解

接下来考虑 $x < y$ 的情况：

由于 a_i 对 $x+y$ 取模后答案不变，所以将 a_i 对 $x+y$ 取模，那么 $0 \le a_i < x+y$，即：一堆石子至多只能被一个人取。又因为 $a_i < x$ 的石子没有贡献，不妨删掉，所以有 $x \le a_i < x+y$。

如果存在一个 i，使得 $x \le a_i < y$，那么先手必胜。因为先手总可以优先取 $a_i \ge y$ 的堆直到取完，这时他还有石子可以取，而后手已经没有可以取的石子了。

如果不存在这样的堆，那么所有堆都有 $a_i \ge y$。此时如果存在一堆满足 $a_i \ge 2x$，那么先手必胜。因为先手可以开场取这堆，从而构造出一堆满足 $x \le a_i < y$ 的石子。否则每堆石子都只能恰好被取一次，判断石子堆数的奇偶性即可。

题解

$x > y$ 的情况类似，请同学们自行推导。

例题 3

初始时你有一张卡片 (a, b) 以及三台机器，这三台机器的功能分别如下：

- 投入一张 (x, y)，生成一张 $(x+1, y+1)$。
- 投入一张 (x, y)，生成一张 $(\frac{x}{2}, \frac{y}{2})$。需保证 x, y 均为偶数。
- 投入一张 (x, y) 和 (y, z)，生成一张 (x, z)。

注意：机器工作完成后会将投入的卡片返还。

现在你希望得到卡片 (c, d)，问是否能在 10000 步内达成，如果能，输出方案，否则输出 -1。

- $1 \leq a, b, c, d \leq 1000$。

题解

首先考虑 $a = b$ 的情况：

容易发现如果 $a = b$，生成的所有卡片也是两数相等的。又因为显然可以用前两台机器将 (a, b) 变为 $(1, 1)$，所以只要 $c = d$，就一定可以。否则一定不行。

题解

接下来考虑 $a < b$ 的情况：

容易发现如果 $a < b$，生成的所有卡片也是第一个数小于第二个数的。我们可以使用前两台机器把 (a, b) 变为 $(1, x)$，其中 x 是偶数。那么 $x - 1$ 是两数之差可能的最小值。也就是说如果 $d - c < x - 1$，那一定无解。

如果有解，则显然需要满足 $(x - 1)|(d - c)$。假设 $d - c = k(x - 1)$，接下来只需要考虑如何将 $(1, x)$ 变为 $(1, k(x - 1) + 1)$。

题解

由 $(1, x)$ 得到 $(1, k(x-1)+1)$ 的策略:

先由 $(1, x)$ 得到

$(x, 2x-1), (2x-1, 3x-2), \cdots, ((k-1)(x-1)+1, k(x-1)+1)$,

再由机器三得到 $(1, 2x-1), (1, 3x-2), \cdots, (1, k(x-1)+1)$ 即可。

$a > b$ 的情况只需要交换 a, b 和 c, d 即可。

例题 4

一个 $n \times m$ 的网格，你可以给每个格子填一个方向（右或下），要求从左上角开始沿着给出的方向走，可以将所有格子恰好经过一次且返回起点。问有多少种定向方法。

网格是循环的，也就是说如果某一步走出了网格，你会到网格的另一边。

- $2 \leq n, m \leq 10^5$。

题解

注意到如果 (i,j) 处往右走，那么 $(i,j+1)$ 是从 (i,j) 过来的，也就意味着 $(i-1,j+1)$ 处不能往下走，所以 $(i-1,j+1)$ 也要往右走。同理往下走也一样。所以可以得到：网格的次对角线上的格子的方向一定相同。

考虑 $\gcd(n,m)=1$ 的情况，此时从任意一点出发，不断向右上方走，可以走遍所有点。即所有点的方向必须一样，这是显然不可以的，所以无解。

题解

接下来考虑 $n = m$ 的情况，此时恰好有 n 条次对角线，现在需要给这 n 条次对角线分配向右还是向下。

设这 n 条次对角线中有 i 个是向右走的，$n - i$ 个向下走的。那么走 n^2 步之后会走到 $(ni \mod n, n(n-i) \mod n) = (0,0)$，即无论怎么分配，走 n^2 步之后都会返回起点。

但是只保证走 n^2 步后返回起点还不够，要保证这 n^2 步经过的点都是不相同的。考虑从起点走 kn 步后会到哪里，因为 kn 步经过了 k 轮次对角线，所以往右走了 ki 步，往下走了 $k(n-i)$ 步，即到了 $(ki \mod n, k(n-i) \mod n)$ 这个点。我们需要保证当 $k < n$ 时，这个点都不为 $(0,0)$。

题解

即问多少个 i 满足 $ki \mod n = 0, k(n-i) \mod n = 0$ 当 $1 < k < n$ 时都不成立。

显然当且仅当 $\gcd(i, n) = 1$ 时满足。

所以枚举往右走的步数 i，只有 $\gcd(i, n) = 1$ 时有贡献，贡献为 $\binom{n}{i}$。故答案为：

$$\sum_{i=0}^{n} [\gcd(i, n) = 1] \times \binom{n}{i}$$

题解

接下来考虑 $n \neq m$ 的情况，根据之前讨论，如果有解那么 $\gcd(n, m) = g \neq 1$。

画图后发现此时的网格共有 g 条次对角线，和 $n = m$ 时使用相同的办法，可以发现当且仅当 $\gcd(i, n) = 1, \gcd(g - i, m) = 1$ 时对答案有贡献，贡献为 $\binom{g}{i}$。所以答案为：

$$\sum_{i=0}^{g} [\gcd(i, n) = 1, \gcd(g - i, m) = 1] \binom{g}{i}$$

例题 5

给出一个 n 个点的树，开始时树上有两只乌鸦和一只松鼠，位于不同位置。现在要进行若干论操作，每轮操作将分为三步：

1、你选择一个乌鸦飞到天上。

2、松鼠在树上随机移动，但不能经过另一只乌鸦。

3、在天上的乌鸦降落回树上（落的位置在第一步确定）。

如果某次降落后，乌鸦恰好在松鼠所在位置，那么松鼠被抓住。问如何操作乌鸦，使得在最坏情况下抓住松鼠的步数最少。

- $2 \le n \le 10^5$。

题解

首先考虑一条链的情况：一种策略是两只乌鸦借助边界一步一步卡松鼠的活动范围，还有一种策略是某只乌鸦先到某一点，使得松鼠的活动范围减小，之后两只乌鸦再借助边界卡松鼠。

树上也是如此：

一种情况是以某只乌鸦为根，之后两只乌鸦轮流将松鼠往子树中卡，此时答案就是根的松鼠方向的儿子的子树的最大深度。

另一种情况是某只乌鸦先跳到某个点上，并以该点为根，接下来同上一种情况。

对于第二种情况，需要枚举第一步前往的所有点，所以复杂度 $O(n^2)$。

题解

注意到第二种情况的答案就是 $\min_{i=1}^{n}(\max_{j=1}^{n} depth_j)$，也即 $\min_{i=1}^{n}(\max_{j=1}^{n} dis(i,j))$，显然这个数为直径除以二上取整。

时间复杂度 $O(n)$。

例题 6

给出 n 个数，在这 n 个数的 $n-1$ 个间隔中加入加号或乘号，问对于所有的 2^{n-1} 种情况，所有表达式的值的和为多少。答案模 10^9+7。

同时你还要支持 q 次修改操作：每次给你两个数 p b，表示把 p 处的值修改为 b，输出修改后的答案。

- $1 \le n, q \le 10^5$。

题解

首先不考虑修改操作：

用一个二元组 (a_i, b_i) 来表示一个表达式的值，含义为这个表达式最后一个加号前的结果为 a_i，最后一个加号之后的数的乘积为 b_i。那么如果在这个表达式后面加入一个数 c，将会产生两个新表达式 $(a_i + b_i, c)$ 和 $(a_i, b_i \times c)$。

设当前一共有 n 个二元组，那么加入一个 c 之后，将变成 $2n$ 个二元组。

设之前所有二元组的和为 (A, B)，那么加入一个 c 之后，将变成 $(2A + B, c(n + B))$。扫一遍并维护 A, B, n 即可。时间复杂度 $O(n)$。

题解

考虑转移 $(A, B, n) \to (2A+B, c(n+B), 2n)$。写成矩阵形式：

$$\begin{bmatrix} A & B & n \end{bmatrix} \times \begin{bmatrix} 2 & 0 & 0 \\ 1 & c & 0 \\ 0 & c & 2 \end{bmatrix} = \begin{bmatrix} 2A+B & Bc+nc & 2n \end{bmatrix}$$

所以答案即为若干个矩阵的连乘，修改 c 即修改某个位置的矩阵。线段树维护即可，时间复杂度 $O(n \log n)$。

Thank you!