牛客竞赛
AC.NOWCODER.COM

2025 牛客

暑期多校训练营 2

小羊肖恩 (Little_Sheep_Yawn)

HR.NOWCODER.COM

概况

- 入门：B、F、I
- 简单：A、L
- 较简单：D、G、H
- 中等：E、K、M
- 较难：C、J
- 好像没有难题！

小羊肖恩 (Little_Sheep_Yawn)

I - Identical Somehow

题目大意

- 对给定的不相等整数 x, y，求一个使得 $(x \bmod k) + (k \bmod x) = (y \bmod k) + (k \bmod y)$ 的 k，不存在则输出 -1。要求 k 不超过 10^{18} 。
- $1 \leq x, y \leq 10^9, x \neq y$

题解

- 如果 x, y 都大于 1 ，那么 $(x \bmod 1) + (1 \bmod x) = 0 + 1 = 1, (y \bmod 1) + (1 \bmod y) = 0 + 1 = 1$ ，即输出 1 。
- 否则，不妨设 $x = 1$ ，则
  - 若 $k = 1$，则 $(x \bmod k) + (k \bmod x) = (1 \bmod 1) + (1 \bmod 1) = 0 + 0 = 0,$ $(y \bmod k) + (k \bmod y) = (y \bmod 1) + (1 \bmod y) = 0 + 1 = 1$ ，不满足要求。

题解 (cont'd)

- 若 $k>1$，则 $(x \bmod k)+(k \bmod x)=(1 \bmod k)+(k \bmod 1)=1+0=1$，而若 $(y \bmod k)+(k \bmod y)=1$，则 $y \bmod k, k \bmod y$ 必然一个 0 一个 1，因此满足整除关系，同时不相等，于是小的数关于大的数取模是 1，与 y,k 中不包含 1 矛盾。
- 因此，此时无解。

- 综上，只需判断 x,y 中是否有 1 即可，时间复杂度为 $\mathcal{O}(1)$。

F - Field of Fire

题目大意

- 一个长度为 n 的由 $0, 1$ 构成的环形数组，每单位时间，当前 1 的位置会向相邻的两个位置扩散，让它们都变成 1 。
- 目前可做的操作是把一个 0 变成 1 ，这个位置的 1 之后不会扩散。问操作一次后，T 时刻的最多的 0 的数量。
- 保证数组中至少有一个 1 。
- $1 \leq n \leq 5 \times 10^5, 1 \leq a_i \leq 10^{18}$

题解

- 先提取出数组的每一段 0 的长度，注意是环形数组，要把最前面的 0 和最后面的 0 拼起来。
- 如果有一段 0 长度是 x ，那么如果这段不操作，T 时刻的长度为 $\max(0, x-2T)$ ，否则，一定对其最边缘的位置进行操作，结果为 $\max(0, x-T-1)$ 。
- 因此操作带来的好处是多了 $\max(0, x-T-1)-\max(0, x-2T)$ 个 0 。这个式子关于 x 单调不减，所以取最长的一段 0 操作就行。
- 时间复杂度为 $\mathcal{O}(n)$ 。
- 这题也可以进一步求进行 k 次操作的最优结果。

AC.

牛客竞赛 AC.NOWCODER.COM

B - Bitwise Perfect

题目大意

- 一个长度为 n 的数组，一次操作会删去其中两个元素，并添加他们的异或。问单次操作是否不会减小 $\sum_{i=1}^{n} 2^{a_i}$ 。
- $2 \le n \le 5 \times 10^5, 1 \le a_i \le 10^{18}$

小羊肖恩 (Little_Sheep_Yawn)

题解

- 考虑删去的元素是 x, y ，新增的元素是 $z = x \oplus y$ ，则如果操作后数值减小，则 $2^x + 2^y > 2^z$ 。该式成立当且仅当 $\max(x, y) \geq z$ 。
- 注意到，如果 x, y 对应的最高位相同，则其异或后，最高位变为 0 ，上述不等式一定成立。
- 而总共只有 60 个不同的二进制位，因此，一旦数组长度达到 61 ，则必有两个数最高位相同，可以直接判断；否则，直接暴力枚举每一对元素是否满足要求即可。
- 时间复杂度为 $\mathcal{O}(n \log M)$ 。

A - Another Day of Sun

题目大意

- 一个长度为 n 且由 $-1, 0, 1$ 构成的数组，将每个 -1 替换为 0 或 1 。
- 问最终对于不同的替换方案得到的数组中，1 构成的连续段数量的和。
- 输出关于 $998\ 244\ 353$ 取模。
- $2 \leq n \leq 5 \times 10^5$

题解

- 如果在数组最前面添上一个 0 ，那么实际计数的就是 $[0,1]$ 作为子数组出现的次数。
- 考虑第 i 个位置和第 $i+1$ 个位置对最终结果的贡献。
- 只考虑这个子数组，$[0,1],[-1,1],[0,-1],[-1,-1]$ 都恰好有一种方案形成一个 $[0,1]$ 子数组，其他情况都不可能。
- 其他 $n-1$ 个位置（目前数组长度为 $n+1$ ）中的空格数量为 x_i ，因此如果是这四种情况，这个数组对最终答案的贡献是 2^{x_i} 。
- 对不同 i 求和即可，幂可预处理，时间复杂度为 $\mathcal{O}(n)$ 。

L - Love Wins All

题目大意

- 一个长度为 n 的排列 P，其中 n 是偶数，问有多少种形成 $\frac{n-2}{2}$ 个数对的方式，使得数对中出现了 $1 \sim n$ 中的 $n-2$ 个不同的元素，且对于其中的任一个数对 (x_i, y_i) 满足 $P[x_i] = y_i$ 或 $P[y_i] = x_i$。结果关于 $998\,244\,353$ 取模后输出。
- $4 \leq n \leq 5 \times 10^5$

题解

- 考虑将排列转为图，即将排列视作有 n 个点的无向图，并将 $i, P[i]$ 之间连一条边。
- 则这个图由一系列环组成。同时最终的方案等价于其中取 $\frac{n-2}{2}$ 条边，使得选择的边之间没有公共点（这里如果两条边都是 u, v 之间的认为是一种方案）。
- 相当于去掉两个点后，剩余的图完全匹配的方案数的和。

题解 (cont'd)

- 图中，奇环的数量一定是偶数，因为环的点数和是 n ，一个偶数。
- 如果奇环的数量不小于 4 ，则删去两个点后，一定还有奇环，而奇环完全匹配，因此方案数为 0 。
- 如果奇环的数量等于 2 ，则删去的两个点一定分别来自两个奇环。可以删去环上的任意点，剩余一条链的完全匹配唯一确定。考虑剩余的环：如果环的大小是 2 ，则只有 1 种方案；否则有 2 种方案。

题解 (cont’d)

- 如果奇环的数量等于 0，则删去的两个点都只能在一个环上。考虑枚举操作的环，假设环的大小为 n ，对环上的点进行黑白交替染色，则为了剩余的点可以形成最大匹配，一定选一黑一白两个点（因为匹配的两点一定一黑一白）。
- 同时，如果选了一黑一白，剩余的点一定能形成完全匹配，所以方案数是 $\frac{n^2}{4}$ 。最后需要乘以剩余的环的总方案数（统计有几个非二元环），再对各个环求和。
- 时间复杂度为 $\mathcal{O}(n)$ 。
- 也可以尝试本题的更难情况，即输入不保证是排列。

小羊肖恩 (Little_Sheep_Yawn)

G - Geometry Friend

题目大意

- 一个凸多边形和一个点，凸边形逆时针绕着点转，且每年转 1 弧度，求凸边形覆盖区域达到最大值的最早时间。
- $3 \le n \le 5 \times 10^5$

题解

- 凸多边形绕一个点覆盖的区域有两种可能——圆或圆环。
- 如果结果是一个圆环，我们考虑内径和外径。此时点一定在凸边形外侧。内径只需考虑图形中与点最近的一个点，这个点在凸多边形的情况下是唯一的。所以此时一定需要绕一周，答案为 2π 。
- 如果结果是一个圆，则点在凸多边形的边缘或内部。此时只需考虑最外侧圆何时完成覆盖。

题解 (cont’d)

- 最外侧的圆是由哪些点画出来的呢？离给定的点最远的一些点，这些点一定是多边形的顶点。
- 因此取出所有最远的点后，按照极角排序，取相邻点之间的最大角度即可。
- 时间复杂度为 $\mathcal{O}(n\log n)$ 。也可以依据给的点是逆时针排列的，进一步优化为 $\mathcal{O}(n)$ 。注意判断点是否在图形内最好使用整数判断。

H - Highway Upgrade

题目大意

- 一个 n 点 m 边的有向图，每条边分别由 u_i 连向 v_i ，边权 t_i ，操作一次边权减小 w_i 。q 次查询，每次查询进行 k 次操作时，1 到 n 的最短路。保证 1 能到达 n 。
- $4 \le n \le 10^5, 1 \le m \le 3 \times 10^5, 1 \le q \le 3 \times 10^5$

题解

- 假设我们已经找到了一条路径，那么我们应该怎么操作呢？我们一定选择其中 v_i 最大的边进行操作。
- 所以，操作的边只能有一条。
- 考虑枚举这条边 (u_i, v_i, t_i, w_i)，则在走这条边前一定走 1 到 u_i 的最短路，走这条边后一定走 v_i 到 n 的最短路。
- 于是可以用原图计算 1 到每个点的最短路，用反向边建图可以得到每个点到 n 的最短路。

小羊肖恩 (Little_Sheep_Yawn)

题解 (cont'd)

- 于是设从 1 开始到 x 的最短路长度为 $d_1(x)$ ，从 x 开始到 n 的最短路长度为 $d_n(x)$ ，则操作 k 次的最短路长度为 $d_1(u_i) + d_n(v_i) + t_i - w_i \times k$。
- 因此，查询的相当于对每一个 k 求上述函数的最小值。因为上面的函数都是关于 k 的一次函数，也就是一系列的直线，于是可以维护一个这些直线形成的可行域（类似于凸包），在分割点序列中进行二分即可。
- 时间复杂度为 $\mathcal{O}(n + m\log n + q\log m)$ 。

D - Donkey Thinks ...

题目大意

- n 个物品，有 (s_i, h_i, d_i) 三个性质，求在满足 $\sum s_i \le V$ 的情况下，$\sum h_i - (V - \sum s_i)(\sum d_i)$ 的最大值。
- $1 \le n \le 10^5, 1 \le V \le 500$

题解

- 主要难以处理的是两项的乘积 $(V-\sum s_i)(\sum d_i)$，但其中 $V-\sum s_i$ 部分数值很小，所以考虑枚举 $S=\sum s_i$ 。
- 此时，每个物品的价值变为 $h_i-(V-S)d_i$ ，就固定了，变成了常规的背包问题。
- 此时再进行背包 DP，求恰好占据 S 单位空间的物品的最大价值。

题解 (cont’d)

- 而如果完全遍历物品会超时。考虑占据空间为 S 时，如果选择空间为 s 的物品，最多选择其中价值最大的 $\lfloor \frac{S}{s} \rfloor$ 个，可以使用 ‘nth_element‘ 函数进行快速选择来解决。
- 这样，选择的总物品数量等于 $\sum_{s=1}^{S} \lfloor \frac{S}{s} \rfloor = \mathcal{O}(S \log S)$ 。
- 进行背包 DP 的时间复杂度就等于 $\mathcal{O}(n + S^2 \log S)$ （包括前面的快速选择物品），关于不同 S 求和，结果为 $\mathcal{O}(nV + V^3 \log V)$ 。
- 因此，总复杂度为 $\mathcal{O}(nV + V^3 \log V)$ ，后半部分常数较小，可以通过。

K - K-th Memorable Sub-string

题目大意

- 对于一个长度为 n 的 01-字符串，如果其存在两个不重合的前缀和后缀完全一致，且间隔不超过 k，则称为好串。动态查询 q 次，每次查询第 k_i 小的串。
- $1 \leq n \leq 10^4, 1 \leq q \leq 5 \times 10^5$ ，且输出总长不超过 10^7 。

题解

- 因为这题的 k 可能很大，所以答案集中的字符串数量可以达到 $\mathcal{O}(n^2)$ 量级。还好 $n \leq 10^4$ 并且空间充足，允许我们使用 $\mathcal{O}(n^2)$ 时间 / 空间的算法。
- 我们先找到所有的满足条件的子串。我们可以对于每一个 i ，对于满足 $j > i$ 的 j 求 $s[i...n], s[j...n]$ 的最长公共前缀，进而可以得到前缀后缀分别以 i, j 开头的子串的可能结尾位置。
- 只需逆序遍历 i ，即可很方便地用 $\mathcal{O}(n)$ 空间完成这部分预处理。

题解 (cont'd)

- 找到了所有可行子串后，如何找第 k_i 大呢？你可以使用后缀数组进行细致处理，但也有很初级的做法。
- 你可以将所有子串插入字典树。这看似插入了 $\mathcal{O}(n^2)$ 个长度为 $\mathcal{O}(n)$ 的串，但你可以同时插入所有起始位置相同的串，这样这些插入就合并到一起了。相当于插入一个最长串时，在中间的某些位置也新增标记。这也保证了占据的空间是 $\mathcal{O}(n^2)$ 的。

题解 (cont'd)

- 接下来找第 k_i 大的逻辑类似于 Trie 上二分。
- 先看当前节点标记数量是否不小于 k_i，如果是就不用往下找了。
- 否则，先将 k_i 减少当前结点的数值，再看 0 那一侧的子树是否字符串数量不小于 k_i ，如果是，下一个字符是 0 。
- 否则，将 k_i 减去 0 一侧的字符串数量，再走向 1 一侧的子树。
- 这种做法的主要的目标是说明，字典树中包含了字符串间的字典序信息。
- 时间复杂度为 $\mathcal{O}(n^2 + \sum |ans|)$ 。

M - Miscalculated Triangles

题目大意

- 三角形三边长度 a, b, c 不超过 s ， 且异或和不超过 l ， 两个三角形不同当且仅当无论怎么旋转、平移都不能重合。问这样的三角形共有多少个。
- s, l 用二进制表示给出，且二进制表示长度不超过 10^5 。

题解

- 一眼就是很典型的数位 DP，但状态需要设计且偏多。
- 用 $DP[idx][a<b][b<c][c<s][a\oplus b\oplus c<l][(a>>idx)+(b>>idx)-(c>>idx)]$ 表示从高到低，考虑到第 idx 位时，将 a 是否严格小于 b，b 是否严格小于 c，c 是否严格小于 s，$a\oplus b\oplus c$ 是否严格小于 l，$a+b-c$ 在更高位时计算得到的结果作为状态，在该状态下的的 (a,b,c) 方案数。
- 状态转移枚举新的一位的 a,b,c 分别的取值即可。

题解 (cont'd)

- 但如果不加限制，$(a >> idx) + (b >> idx) - (c >> idx)$ 可能很大。但我们最终的目标是到 2^0 这一位时，这个数至少为 1 ，所以过大和过小的数值是不必要的，因为后面的位不会影响到最后的判断。
- 事实上，只要这个数值超过了 2 ，后面就不可能变成负数了。这个数值小于了 -1 ，后面就不可能变成正数了。所以只需大于 2 时，取和 2 的最小值，小于 -1 时，直接不作考虑即可。这样，对于这一状态而言，不同的情况就只剩下 $-1, 0, 1, 2$ 四种了。

题解 (cont'd)

- 最后统计答案时，保留所有最后一个状态大于 0 的所有 DP 结果，就可以得到满足条件的且满足 $a \le b \le c$ 的三元组。
- 接下来考虑翻转产生的不同的三角形，事实上只需再把满足 $a < b, b < c$ 的所有结果加总到答案中就行。
- 综上，总状态数是 $\mathcal{O}(n)$ 的，常数比较大，再算上转移，总复杂度是 $\mathcal{O}(n)$ 的。

E - Effective Numbers

题目大意

- 求 $[l, r]$ 中有多少个数，加上它的一个质因子等于完全平方数。
- $2 \le l < r \le 10^{18}$

题解

- 很经典的是，只需求解问题——小于等于 x 的数中有多少个满足条件，设有 $f(x)$ 个，则返回 $f(r)-f(l-1)$ 即可。接下来对于整数 V 求解该问题。
- 假设一个数加上其质因子 p 后，变成了完全平方数 x^2 。则 p 也是 x^2 的因子，进一步，也是 x 的因子。
- 所以 $x^2>x^2-p\geq x^2-x>(x-1)^2$ 。因而对于一个数而言，加上其质因子最多只可能得到一个完全平方数。

题解 (cont'd)

- 所以，如果有 $x^2 \le V$，那么 x^2 减去任何一个 x 的质因子都满足要求，且不会跟其他的平方数减去质因子的结果重复。
- 因此，如果设 x_0 是平方不超过 V 的最大的数，则我们要求不超过 x_0 的数的质因子个数的和，再加上满足 $(x_0+1)^2 - p'$ 形式的数的个数（因为这个数仍然可能不超过 V，但是 $(x_0+2)^2 - p''$ 就一定至少是 $(x_0+1)^2$ 了，就一定不满足要求）。
- 后面这部分是很容易的，只需对 x_0+1 进行质因数分解即可。接下来考虑前面的部分，即求不超过 x_0 的数的质因子个数的和。这里 x_0 的范围并不大，最大只有 10^9。

题解 (cont'd)

- 考虑每个质因子 p 出现的次数，显然是 $\left\lfloor \frac{x_0}{p} \right\rfloor$，所以对于不超过 10^7 的 p，可以直接这么求出答案。
- 对于更大的 p，注意到 $\left\lfloor \frac{x_0}{p} \right\rfloor$ 的数值会比较小，因此考虑枚举这个数值。
- 在这个数值等于 val 时，可以确定一个 p 的范围，我们相当于求这个范围内的质数个数，可以用 min_25 筛等多种方式进行解决。
- 设取的分界位置是 P，则时间复杂度为 $\mathcal{O}(\sqrt{M}/P + \frac{M^{3/8}}{\log M} + P)$，前半部分是素数筛与枚举小质数，后半部分是用 min_25 筛的使用。在 P 取 $M^{1/4}$ 时最优，因此复杂度为 $\mathcal{O}(\frac{M^{3/8}}{\log M})$。

小羊肖恩 (Little_Sheep_Yawn)

C - Colorful Tree

题目大意

- 一棵树，以 1 为根，由 i 结点等可能地选取 $1, 2, \ldots, i-1$ 中的点进行连边形成，每个点等可能地选取 $[l_i, r_i]$ 中的正整数作为其颜色。处理 q 次查询。查询 1：修改一点的颜色区间。查询 2：求 u 点对应子树的极大同色联通块个数的期望和方差。
- $1 \le n \le 5 \times 10^5, 1 \le q \le 10^5$

题解

- 称一条边为异色边，当且仅当其端点颜色不同。极大连通块的数量等于异色边的数量加 1。
- 设 i 子树对应的边集为 $e_1, e_2, \ldots, e_k$ ，设随机变量 Z_i ，其为 0 当且仅当 e_i 为异色边，否则取 1 。
- 则设子树的极大同色连通块数量为 Y ，则我们的目标是求随机变量 $Y = 1 + \sum_{i=1}^{k}(1 - Z_i)$ 的期望和方差。

题解 (cont'd)

- 关于期望，我们根据期望的可加性，只需算出每一项 $\mathrm{E}Z_i$ 即可，这个数值等于这条边是同色边的概率，这是非常容易计算的。
- 关于方差，注意到 $\mathrm{Var}(Y)=\mathrm{Var}\left(1+\sum_{i=1}^{k}(1-Z_i)\right)=\mathrm{Var}\left(\sum_{i=1}^{k}Z_i\right)$ ，可以有多种理解方式用于求解——
  - 使用公式 $\mathrm{Var}(X)=\mathrm{E}(X^2)-\mathrm{E}^2(X)$ ，则注意到 $\mathrm{Var}(Y)=\mathrm{Var}\left(\sum_{i=1}^{k}Z_i\right)=\mathrm{E}\left(\left(\sum_{i=1}^{k}Z_i\right)^2\right)-E^2\left(\sum_{i=1}^{k}Z_i\right)$ ，后者前面算过了，所以看前者。完全平方计算后，得到 $\sum_{i,j}\mathrm{E}(Z_iZ_j)$ 。

题解 (cont'd)

- 如果 $i=j$ ，则 $\mathrm{E}(Z_iZ_j)=\mathrm{E}(Z_i)$ 。
- 如果边 i,j 没有任何公共点，则两个随机变量相互独立，有 $\mathrm{E}(Z_iZ_j)=\mathrm{E}(Z_i)\mathrm{E}(Z_j)$ 。
- 如果边 i,j 有公共点，则 $\mathrm{E}(Z_iZ_j)$ 算的是边 i,j 都是同色边的概率，即三个点同色的概率，也是好求的。

- 更方便的方法是，直接使用

$$\mathrm{Var}\left(\sum_{i=1}^{k} Z_i\right)=\sum_{i=1}^{k}\mathrm{Var}\,Z_i+2\sum_{1\le i<j\le n}\mathrm{Cov}(Z_i,Z_j)。$$

  - 若边 i,j 没有公共点，则随机变量独立，协方差为 0 。
  - 否则可以通过 $\mathrm{E}(Z_iZ_j)-\mathrm{E}Z_i\mathrm{E}Z_j$ 计算协方差，方式同上。

题解 (cont'd)

- 总结上述过程，关于子树，需要计算的主要包括——$\mathrm{E}Z_i$ 以及对于有公共点的边计算 $\mathrm{E}Z_iZ_j$ 。
- 而只需将这些数值记录到对应边的最靠近根节点 1 的结点，就可以用子树求和对整个子树的该变量进行统计了。子树求和有许多处理方式，比如用 DFS 序把子树转化为一个区间，就变成了单点更新区间求和的问题，就可以很方便地使用树状数组进行维护。

题解 (cont'd)

- 修改结点的时候可以怎么更新呢？事实上只需要暴力找其附近距离不超过 2 的点进行计算即可。
- 从期望的角度来看，结点 i 的度数的期望是 $\frac{1}{i}+\frac{1}{i+1}+\cdots+\frac{1}{n}\le 1+\frac{1}{2}+\cdots+\frac{1}{n}\sim\mathcal{O}(\log n)$，所以平均意义上而言，这种随机树的生成方式满足点的度数不会太大。
- 所以每次更新时，直接暴力维护跟当前点有关的 $\mathrm{E}Z_i,\mathrm{E}Z_iZ_j$，并在根节点汇总更新的数值后，再更新。寻找两层，所以复杂度是 $\mathcal{O}(\log^2 n)$ 的。
- 求期望和方差的部分直接使用树状数组即可，时间复杂度为 $\mathcal{O}(\log n)$。
- 时间复杂度为 $\mathcal{O}((n+q)\log^2 n)$。

小羊肖恩 (Little_Sheep_Yawn)

J - Just Curve It

题目大意

- 有一个操作序列，每个操作是线性函数 $kx+b\ (k\le 1, k+b\ge 1)$ 和 $\sqrt{x}$ 之一。查询 q 次，每次可能执行修改或查询。
- 修改操作——使得某个下标变为另一种操作。
- 查询操作——给定一个输入 $x\ (1\le x\le 100)$ ，取一个操作子数组 $[l,r]$ ，其中操作为 $op_0, op_1, \ldots, op_{r-l}$ ，设 $a_0 = op_0(x), a_1 = op_1(a_0), \ldots, a_{r-l} = op_{r-l}(a_{r-l-1})$ ，求 a_{r-l} 和 $\sum_{i=0}^{r-l} a_i$ 。相对误差与绝对误差的较小值不超过 10^{-4} 。
- $1\le n\le 10^5, 1\le q\le 10^5$

小羊肖恩 (Little_Sheep_Yawn)

题解

- 一些相对基础的观察：
  - 数值永远保持在 1 以上。
  - 只做 $\sqrt{x}$ 后数值会不断逼近 1 。
- 所以，可以猜测（当然熟悉的朋友应该已经知道是咋回事了），尽管 $\sqrt{x}$ 实在是难以处理，但可能多做几次，之前的数就不重要了。
- 具体地，如果一开始的输入是 $[l, r]$ 中的一个数，在经过一系列上述的可能操作后，只要操作中有足够多的 $\sqrt{x}$ ，最后结果就几乎跟初始值无关。

题解 (cont'd)

- 这件事其实是好证明的。我们考虑一个区间 $[l, r]$ 经过操作后，区间长度怎么变化。
  - 如果经过线性变化，则 $f(r) - f(l) = k(r - l) \le r - l$，所以区间长度不会边长。
  - 如果经过 $\sqrt{x}$，则 $f(r) - f(l) = \sqrt{r} - \sqrt{l} = \frac{r-l}{\sqrt{r}+\sqrt{l}} \le \frac{r-l}{2}$，所以区间长度至少减半。
- 因此，即使是在上面非常粗略的估计下，每经过一次 $\sqrt{x}$ 操作，区间长度仍然减半，因此只要 $\sqrt{x}$ 的次数足够多，此时的输出就跟输入没有关系了。

AC.

题解 (cont'd)

- 上面只计算了“绝对误差”，关于“相对误差”——
  - $\frac{f(r)}{f(l)} = \frac{kr+b}{kl+b} = \frac{r+b/k}{l+b/k} \leq \frac{r}{l}$，即相对距离减小。
  - $\frac{f(r)}{f(l)} = \frac{\sqrt{r}}{\sqrt{l}} = \left(\frac{r}{l}\right)^{\frac{1}{2}}$，即相对距离减半。
- 以上都说明了，只要经过足够多次的 $\sqrt{x}$ 操作，左右边界的相对和绝对差别会很小。

题解 (cont'd)

- 于是考虑用线段树维护操作信息。考虑前 $\mathcal{O}(\log \epsilon^{-1})$ 次 $\sqrt{x}$ 操作，两个相邻的 $\sqrt{x}$ 操作之间有一些线性操作。我们维护这些线性操作，计算可得它们会把 x 变成 k_1x+b_1（维护这个 k_1,b_1），以及中间遍历的数的和 k_2x+b_2 。同时取一个代表元素 x 维护其在该区间经过所有操作后得到的数值，及其第 $\mathcal{O}(\log \epsilon^{-1})$ 次 $\sqrt{x}$ 操作后，每一次操作得到的数的和。
- 时间复杂度为 $\mathrm{O}(n\log\epsilon^{-1}+q\log n\log\epsilon^{-1})$ 。

牛客竞赛 AC.NOWCODER.COM

牛客暑期多校 训练营

THANKS!

AC.NOWCODER.COM