牛客竞赛
AC.NOWCODER.COM

2024 牛客 暑期多校训练营 第七场

qcjj 粉丝团

HR.NOWCODER.COM

A - Maximum Subarray Sum

题目大意

- 长度为 N的数组 a，首先执行至多 M次元素交换（不要求相邻）操作，然后从中选择长度不小于 K的 $\lfloor \frac{N}{K} \rfloor$个不交叉子数组，求所有所选子数组的元素之和最大是多少
- $1 \leq K \leq N \leq 10^4, 0 \leq M \leq 20$
- $-10^5 \leq a_i \leq 10^5$

AC.

A - Maximum Subarray Sum

题解

- 这个题的出题思路是轮廓线 DP 缝合了一个 2018 中国大学生程序设计竞赛-女生专场——寻宝游戏（轮廓线背包）
- 两个套路的结合
- 现代的轮廓线 dp 不再局限于插头或者状态压缩，典型的例如 2019CCPC 秦皇岛 G，还有上面女生赛的题
- 轮廓线 dp 的核心逻辑是，如果某种操作可以表示成“选，不选”的"01"序列。
- 那么就可以以左上角为起点，构造一个只能往右或下走的二维模型，同时令 0 表示往右，1 表示往下

A - Maximum Subarray Sum

题解

- 解题时先不要考虑太多，首先想不考虑交换的前提下，如何用轮廓线表示一个合法的取数方案
- 首先在数组末尾补充一个"*"，表示终点，然后按照每行 k个元素依次把数组中的元素放到矩阵中，且仅有最后一列可以不满。
- 这样就构造了一个 n/k行的矩阵，且终点位于 $P(n/k, n\%k)$

A - Maximum Subarray Sum

题解

- 一条轮廓线可以表示一个合法的取数方案，当且仅当
- ①所有不选的区间，全部位于轮廓线向右移动的段上
- ②每一行只允许存在一段不选，且第一行与最后一行开始与结束的位置不能有孤立的所选段

A - Maximum Subarray Sum

题解

- 接下来考虑贪心，由于至多交换 m次，所以对于每一条轮廓线，物品数目大于 m是没有意义的，这一步可以提前用堆预处理好
- 用五维数组 dp[x][y][i][j][2] 表示当前轮廓线走到 (x, y)，交换出去 i个物品，交换进来 j个物品，当前所选/不选段是否连续
- 细节处理比较麻烦，具体看 std 代码吧
- $O(nm^3)$

A - Maximum Subarray Sum

解法 2

- 这里是验题人给出的另一种解法：
- 令 $f_{i,x,y,0/1}$ 表示当前 dp 到了第 i 个点，区间内有 x 被换出去了，y 个换进来的，前一个点是否被选中了。
- 定义，只选择若干个 $\lfloor \frac{n}{k} \rfloor$ 长度为 k 的区间作为答案，后面附属的数全都是和没有被选中的数地位相同。
- 注意到，一个合法的决策当且仅当在第 i 个点扔掉的数的个数是 $i \mod k$，因为首先它不可能存在大于等于 k 个数你没有选择，那么剩下的数也只能是 $i \mod k$ 个。

A - Maximum Subarray Sum

解法 2

- 此时会发现，只需要在 dp 的时候注意 $i \bmod k = k - 1$ 的时候一定要选中这个数和后面的 $k - 1$ 个成为一个区间就可以了。
- 转移是非常容易的，分成选中和不选，其中不选和选中但是是附属的转移能在 $O(1)$ 的时间复杂度内解决。
- 只有选中的时候，需要枚举这个区间内有多少个数出去了就可以，这部分复杂度是 $O(m)$。
  总的时间复杂度是 $O(n \times m^3)$。

B - Lottery

题目大意

- 每次从 $[0, i]$的整数中等概率产生随机数，共进行 N次，问随机数之和大于 $\frac{N \cdot M}{K}$的概率是多少
- $1 \leq T \leq 10^5$
- $1 \leq N \leq 10^5, 1 \leq M, K \leq 10^5, \frac{M}{K} \leq 2$

B - Lottery

题解

- 原本是 poly 入门题，但是在比赛头一天 q 说能做到总价值 $2N$，我们先假设总价值不大于 N，首先构造多项式生成函数
- $f(x) = \frac{1}{2}(1+x) \cdot \frac{1}{3}(1+x+x^2) \cdot \frac{1}{4}(1+x+x^2+x^3) \cdot \ldots \cdot \frac{1}{N}\sum_{i=0}^{N} x^i$
- 如果你不懂这个式子怎么写出来的，搜索“母函数——砝码问题”
- 等比数列求和，整理得
- $\frac{1}{N!(1-x)^{N+1}}\prod_{i=1}^{N+1}(1-x^i)$

B - Lottery

题解

- 继续拆成三部分的贡献乘积 $\frac{1}{N!}, \frac{1}{(1-x)^{N+1}}, \prod_{i=1}^{N+1}(1-x^i)$
- 其中第一部分 $\frac{1}{N!}$是一个与 x无关的常数项，最后再乘就行
- 第二部分和第三部分，在算法竞赛中属于多项式常识级别的知识，如果你不知道，你需要把它背过

B - Lottery

题解

- 第二部分 $\frac{1}{(1-x)^k}$，它表示 k阶前缀和的生成函数，其中 k的取值可以为任意整数，当 $k<0$时，它表示 $|k|$阶差分
- 当然，你可以通过万能的数论计算器，直接计算其在 x=0 处的多项式展开 https://www.wolframalpha.com/
- $1, C(k,1), C(k+1,2), C(k+2,3), ..., C(k+i,i+1)$
- 对于这个结果，有一个非常精妙借助二维走格子辅助推导的方式。如果你并不会计算什么多项式求逆/泰勒展开的话，可以使用此方法理解该多项式展开

B - Lottery

题解

- 第三部分 $\prod_{i=1}^{k}(1-x^i)$，一眼反应过来长得很像大名鼎鼎的五边形数 $\prod_{i=1}^{\infty}(1-x^i)$
- 在多项式领域的 x^∞不代表真正的无穷，只要其超过取值范围，任何数都可以等价无穷
- 例如取值范围是 $[0,10]$，那么 $11,12,13$都可以看成是 ∞
- 本题中由于取值范围是 $\frac{N \cdot M}{K}$，假设其是一个小于等于 N的数，此时在多项式意义下 $x^k=x^\infty$，即第三部分就是五边形数

AC.

B - Lottery

题解

- 接下来考虑 $[n+1,2n]$的部分，我们先把它当成五边形数处理，这样会多乘上 $(1-x^{n+1})\cdot(1-x^{n+2})\cdot\ldots\cdot(1-x^{2n})$
- 最直观的做法，把它用除法除掉，则这部分的多项式逆为 $(1+\sum_{i=n+1}^{2n}x^i)$，直接在卷积处理时加上这部分贡献即可

B - Lottery

题解

- 由于题目要求还要再求个前缀和，所以前缀和部分的 N取 $N+1$
- 对于五边形数，其在 N的范围内只有 $\sqrt{N}$项非 0，暴力计算的复杂度为 $O(\sqrt{N})$
- 时间复杂度 $O(T\sqrt{N})$

C - Array Sorting

题目大意

- 构造一个并行的排序算法，可以排序一个 10^4 的数组，要求并行比较次数不多于 200次。
- $1 \le N \le 10^4$

C - Array Sorting

题解

- q：3-smooth number shellsort 秒了
- 外网搜索 Sorting network，应该能找到很多算法，任意一种 $nlog^2n$复杂度的排序网络算法都能够通过本题
- 这里介绍一下双调排序算法，又称双调归并排序，其核心思路和归并排序差不多

C - Array Sorting

题解

- ①对于一个序列，如果它是单调序列，则它也是一个双调序列
- ②对于一个序列，如果满足先单调递增，再单调递减或者反过来先单调递减再单调递增，则它是一个双调序列
- ③循环位移下满足①或②的序列，是双调序列

C - Array Sorting

题解

- Batcher 定理：对于一个长度为 $2n$的双调序列 a，定义 $Min_i = min(a_i, a_{n+i})$， $Max_i = max(a_i, a_{n+i})$
- 则 Min序列的最大值小于等于 Max序列的最小值，且 Min, Max序列仍为双调序列
- 注意到定理中的这一句：Min**序列的最大值小于等于** Max**序列的最小值**

C - Array Sorting

题解

- 假设一开始有一个长度为 8的双调序列，我们做一次 $Max-Min$拆分，将 Min序列放到左侧，将 Max序列放到右侧。
- 此时左边 4个数就不大于右边 4个数，同时由于 $Max-Min$拆分不改变双调性质，所以可以继续递归处理 $[1,2],[3,4]$
- 然后再递归处理 $[5,6],[7,8]$，此时所有的数字均已排序完毕，这个排序的代价是并行 $O(logn)$的（串行模式下为 $O(nlogn)$）

C - Array Sorting

题解

- 那么接下来，需要解决的问题就只剩下两个了
- 一、初始不是双调序列怎么办
- 二、长度不是 2的幂次怎么办

C - Array Sorting

题解

- 想这么一个问题，如果我有两个长度相等的单调序列，我把它按照单调性相反左右拼接，能否得到一个双调序列
- 想第二个问题，数组中任意一个元素（长度为 1的序列），它是不是一个单调序列，数组中任意两个相邻元素（长度为 2的序列），它是不是一个双调序列
- 想第三个问题，我们已经有了一个算法，可以在 $O(logn)$的并行复杂度下，将一个双调序列变成单调序列。

AC.

C - Array Sorting

题解

- 假设现在有个长度大小为 8的数组，先按照大小为 2进行分组 $[1,2],[3,4],[5,6],[7,8]$。
- 显然任何长度大小为 2的序列都是双调序列，则对 $[1,2],[5,6]$升序排序，对 $[3,4],[7,8]$降序排序
- 排序后，拼接 $[1,2],[3,4]$得到 $[1,4]$拼接 $[5,6],[7,8]$得到 $[5,8]$
- $[1,4],[5,8]$都是双调序列，所以一个升序排序一个降序排序后拼接得到 $[1,8]$构造了一个大的双调序列

AC.

C - Array Sorting

题解

- 如果长度不是 2^k怎么办，可以自行发挥下想象力解决这个问题
- 比较简单的方式是在末尾补 inf，但是要注意 inf的实际位置，避免最后排错

D - Interval Selection

题目大意

- 氧气少年有一个长度为 $n(1 \le n \le 2 \cdot 10^5)$ 的序列，序列中的某个区间 $[l, r]$ 是好的，当且仅当 $a_l, a_{l+1}, \ldots a_r$ 中的每个元素在当前区间中恰好出现了 $k(1 \le k \le n)$ 次。
- 请你求出可以选择的好区间的数量。

D - Interval Selection

题解

- 考虑固定左端点，统计右端点。倒序枚举左端点，对于第 i 个元素：
- 它向右第 k 个与 a_i 相同元素的下标，记为 l，它向右第 $k+1$ 个与 a_i 相同元素的下标，记为 r。
- 记 $[l, r-1], [1, i-1]$ 为 i 的合法区间（因为对于区间 $[i, l], [i, l+1] \dots [i, r-1]$，$a_i$ 都出现了恰好 k 次；对于区间 $[1, i-1]$，由于这里是倒序处理，所以 $[1, i-1]$ 不在第 i 个元素的影响范围）。
- 对于以 i 为左端点的好区间，它对答案的贡献是所有合法区间的交集在 $i \dots n$ 的长度。
- 使用线段树实现即可。

E - Flowers

题目大意

- n 个点的树，第 i 个点有颜色 c_i $(1 \leq i \leq n)$
- 求有多少个连通子图，满足：
  1. 子图上度数为 1 的点的颜色均相同
  2. 子图上度数为 1 的点的数量不大于 k
- $1 \leq k \leq n \leq 10^5, 1 \leq c_i \leq n$，答案模 998244353

E - Flowers

关键词

- 虚树
- 全局平衡二叉树
- NTT

E - Flowers

题解

- 只要确定了连通子图的叶子，就确定了这个连通子图
- 可以独立考虑每种颜色，建虚树
- 颜色为 c 的虚树上，设 $f(u)$ 是一个多项式，$[x^j]f(u)$ 表示以 u 为根的子树下取 j 个颜色为 c 的叶子的方案数
- $$f(u) = \prod_{\text{child } v} f(v) + [c_u = c]x$$

E - Flowers

题解

- 设 $s(u)$ 多项式，$[x^j]s(u)$ 表示以 u 为根的子树下，连通分量的根度数为 1 的，有 j 个颜色为 c 的叶子（不包含根）的连通分量的数量
- $$s(u) = \sum_{\text{child } v} s(v) + [c_u = c]f(v)$$
- 最后答案是 $\sum_{j=1}^{k-1}[x^j]s(rt) + \sum_{j=1}^{k}[x^j]f(rt)$

E - Flowers

题解

- 以全局平衡二叉树的形式合并答案
- 首先对树做一个轻重链剖分
- 合并轻孩子结点时可以每次贪心选 size 最小的两个合并

E - Flowers

题解

- 重链上假设 $\mathrm{hc}(u)$ 是 u 的重孩子结点，$t(u)=\sum_{v\neq \mathrm{hc}(u)} s(v)+[c_u=c]f(v)$ 和 $g(u)=\prod_{v\neq \mathrm{hc}(u)} f(v)$

$$\begin{bmatrix} s(u) \\ f(u) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & [c_u=c] & t(u) \\ 0 & g(u) & [c_u=c]x \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s(\mathrm{hc}(u)) \\ f(\mathrm{hc}(u)) \\ 1 \end{bmatrix}$$

- 设 $w(u)=\mathrm{size}(u)-\mathrm{size}(\mathrm{hc}(u))$ 为 u 的权值
- 以加权的中点作为分界进行分治

E - Flowers

复杂度分析

- 以下分析重孩子结点分治的复杂度，轻孩子的合并可以类似分析
- 考虑序列 $w_1, w_2, \ldots, w_k$, 它们的和 $w = \sum_{i=1}^{k} w_i$
- 按以上策略分治时， w_i 在分治树上的深度不超过 $2 + \log_2(w/w_i)$
- 假设一个点到根经过的所有轻孩子结点是 $u_1, u_2, \ldots, u_k$， $k \le \log_2(n)$
- 该点所在的深度不超过

$$2k + \log_2\left(\frac{n}{w(u_1)}\right) + \log_2\left(\frac{w(u_1)}{w(u_2)}\right) + \ldots + \log_2(w(u_k)) = O(\log(n))$$

- 于是整棵全局平衡二叉树的高度是 $O(\log(n))$ 的
- 总复杂度 $O(n\log^2(n))$

F - Teleportation

题目大意

- 从正整数 x 可以一步跳到正整数 y，当且仅当存在另一个正整数 z，满足 $\frac{1}{x}+\frac{1}{y}=\frac{1}{z}$
- 给定 $3 \le x, y \le 10^9$
- 求一条路径 $v_0 = x, v_1, v_2, \ldots, v_m = y$
- 满足 $m \le 150, 3 \le v_i \le 10^{100}$ 且 v_i $(0 \le i < m)$ 可以一步跳到 v_{i+1}

F - Teleportation

通解形式

- 条件等价于 $(x+y) \mid xy$
- 假设 $\gcd(x,y)=g$，$x=ga, y=gb$
- $(a+b) \mid gab$
- 因为 $\gcd(a+b,a)=\gcd(a+b,b)=1$
- 所以 $(a+b) \mid g$
- 假设 $g=(a+b)d$，有通解 $x=a(a+b)d, y=b(a+b)d$

F - Teleportation

题解

- 考虑将 x 变成 2^k
- $2y+1 \Longrightarrow 2y(2y+1) \Longrightarrow 2(y+1)(2y+1) \Longrightarrow 2(y+1)$
- $j \geq 1$ 时，$(4y+1)2^j \Longrightarrow 4y(4y+1)2^j \Longrightarrow 4y(4y-1)2^j \Longrightarrow y2^{j+2}$
- $(4y+3)2^j \Longrightarrow (4y+3)(4y+2)2^j \Longrightarrow (4y+3)(4y+4)2^j \Longrightarrow (y+1)2^{j+2}$
- x 变为 2^k 的次数上界是 $3+\frac{3}{2}\lceil\log_2 x\rceil$
- 从 2^i 变为 2^j: $2^i \Longrightarrow 3\cdot 2^i \Longrightarrow 3\cdot 2^{i+1} \Longrightarrow \ldots \Longrightarrow 3\cdot 2^j \Longrightarrow 2^j$
- 总次数上界不超过 $2(3+\frac{3}{2}\cdot 30)=96$
- Bonus: 可以进一步构造上界不超过 80 的做法（也许还能更小？）

G - Frog Crossing

题目大意

- 按照简单几何图形（矩形）的方式给定一张平面图，求删除每一条边后的最大流。
- $1 \leq N \leq 10^5, 1 \leq M \leq 2 \times 10^5$
- $1 \leq x_i \leq N, 1 \leq y_i, l_i, v_i \leq 10^9$

G - Frog Crossing

题解

- 平面图删边最大流 → 平面图删边最小割 → 对偶图必经边最短路
- 不懂的话就搜一下“狼抓兔子”
- 第一步是平面图转对偶图，方法不正确的话处理起来可能会很麻烦

G - Frog Crossing

题解

- 注意到对于任何一行，“木板”与“两块板之间的水流——简称水板”；“青蛙自上而下移动”与“水流从左流到右侧”，在几何意义上互为对偶关系
- 以“水”为主体，“水板”是对偶图的点，“木板”是对偶图的边，边权为木板的权值，左侧无限远为源点 S，右侧无限远为汇点 T建图，则从 S到 T的最短路，就是平面图的最小割/最大流
- 细节处理上，相同“水板”形成的连通区域，可以用并查集，也可以建一条长度为 0 的边，注意“水板”不同于“木板”，斜角相邻也算相连

G - Frog Crossing

题解

- 接下来就是一个老生常谈的问题必经边最短路，分两种情况讨论
- 1、修改权值后导致最短路变短，则起点跑一次最短路，终点跑一次最短路，答案等于 $pre[u] + suf[v] + dis[u][v]$
- 2、修改后最短路不变，答案还是 $dis[s][t]$
- 每次查询 v_i改成任意值（修改权值后最短路变长）也是可做的，搜索下"删边最短路"
- $O(mlogn)$

H - Database

题目大意

- 氧气少年需要你模拟一个数据库表，具有固定列数 $n(1 \leq n \leq 1000)$，每列有唯一字段名
- 需要实现的 SQL 语句包括插入（`insert`）、选择（`select`）、删除（`delete`）、带条件集合的选择（`select_in`）和带条件集合的删除（`delete_in`）
- 需要实现的事务语句包括开始（`begin`）和结束（`commit` 或 `abort`）
- 语句数量 $1 \leq q \leq 3000$，每条语句长度不超过 2000，嵌套层数不超过 10

AC.

牛客竞赛 AC.NOWCODER.COM

H - Database

题解

本题着重考验选手的代码实现能力。实现的方式有很多，但请注意复制条目和查询的开销。以下是有关代码实现的提示。

- 对于事务语句：不建议在事务开始的时候复制整张表作为副本，而是在新增或已删除的条目做标记；
- 对于 SQL 语句：字符串可能很长，建议使用字符串哈希加快查询效率。

I - Fight Against the Monster

题目大意

- 需要造出若干台战争机器对抗怪兽。每台机器有”战斗”和”创造”功能
- 每台机器使用”战斗”功能会使怪兽血量减 1，但机器会丧失所有功能;
- ”创造”功能需要 $m(1 \le m \le 10^6)$ 台机器同时使用，在这之后会产生 $k(1 \le k \le m)$ 台新的战争机器，每台机器仅可使用一次”创造”功能
- 怪兽的初始血量为 $h(0 \le h \le 10^9)$，其血量不超过 0 时就会死亡。请计算最初至少需要多少台战争机器才能打败怪兽
- $1 \le T \le 2 \cdot 10^5$

I - Fight Against the Monster

题解

- 判断能否取得胜利，只需要判断战争机器能产生的最大伤害 sum 是否不小于怪兽血量 h 即可。已知使用"战斗"功能后机器会报废，所以优先使用"创造"功能增加机器数量，然后使用"战斗"功能即可得到最大伤害 sum。
- 所有机器能产生的最大伤害随着机器的初始数量增大而增大，要求出至少需要多少台战争机器，可以使用二分答案来解题。

I - Fight Against the Monster

题解

- 每次使用“创造”功能会使没有使用过“创造”功能的机器数量减小 $d = m - k$，进而得到 x 个机器使用“创造”功能的次数 $cnt = \frac{x-m}{d} + 1$，进而求出最大伤害 $sum = x + cnt \cdot k$，每次 check 时间复杂度为 $O(1)$。
- 另外，当 $h \leq m$ 或者 $m = k$ 时，我们可以直接得到答案 $\min(h, m)$。
- 单组测试用例的复杂度为 $O(\log h)$

AC.

牛客竞赛 AC.NOWCODER.COM

J - Ball

题目大意

- 给定一根长木棒和一个点 p，问是否存在一个旋转中心使得木棒在旋转过程中可以途径点 p。
- $1 \leq T \leq 10^4$
- $1 \leq l \leq 10^5, -10^5 \leq x, y \leq 10^5$

J - Ball

题解

- 贪心，只使用木棒的左右两端点作为旋转轴时覆盖到的区域一定包括了木棒上任意一点作为旋转轴。
- 可以使用动点方程/求导等方式证明。
- 这里提供一个直观的方式，也是常用工具。
- 首先打开函数计算器：https://www.geogebra.org

J - Ball

题解

- 输入如下三条公式
- $a = 100 - b, (x-b)^2 + y^2 = a^2, (x-b)^2 + y^2 = b^2$
- 设置动点范围为 $[0, 100]$，点击播放动点。
- 观察到对于圆上任意一点，收缩的趋势明显大于向右移动的趋势，因为差异过大，所以不用求导也可以直观的看出不优。
- 所以只用判断点是否在以左右端点为圆心，半径为 l 的圆内即可。
- $O(1)$

AC.

牛客竞赛 AC.NOWCODER.COM

K - Strings, Subsequences, Reversed Subsequences, Prefixes

题目大意

- 给定两个字符串 S, T，求 S中有多少本质不同的子序列，包含前缀 T并且要求该子序列的反串也同时包含前缀 T。
- $1 \leq N, M, \leq 10^6$
- $S_i, T_i \in [a - z]$

K - Strings, Subsequences, Reversed Subsequences, Prefixes

题解

- 首先先贪心，从 S的前缀中取一个子序列 T，找到能够包含该子序列且最短的前缀位置，然后反过来从后往前找能包含该子序列且最短的后缀位置。
- 如果这两个位置没有交叉，首先答案至少是从这两个位置切开，剩余部分本质不同的子序列数目，这是一个经典字符串 dp，你可以直接搜索“动态规划：本质不同子序列数”。

K - Strings, Subsequences, Reversed Subsequences, Prefixes

题解

- 例如 $S = "abegfhbca", T = "ab"$，此时贪心取到的最短前缀为 ab，最短后缀为 bca。把这两部分去掉，相当于求中间部分 $"egfh"$的子序列数目。
- 当然，如果只算不交叉部分，会发现答案漏了一些。正好丢掉了 T在所选子序列中交叉的情况，同时注意到 T一旦在正反串中交叉，就会使得该字符串变为回文串。

K - Strings, Subsequences, Reversed Subsequences, Prefixes

题解

- 而 T又是该回文串的一个前缀，所以对 T的后缀跑一个马拉车/哈希求回文中心，对于每一个回文中心，看对称过去之后的部分，能否在 S中取到，这里只需要考虑每个回文中心能构成最长的回文即可（因为回文中心相同时，实际上最后构成的子序列本质相同）。检查每一个后缀回文中心能否取到对应的子序列，加上其贡献即可。
- $O(N)$

牛客竞赛
AC.NOWCODER.COM

牛客暑期多校 2023 训练营

THANKS!

AC.NOWCODER.COM