浙江工业大学第十九届“杭银理财杯”大学生程序设计竞赛暨全国邀请赛

题目分析

奥斯卡和她的小伙伴们

2022 年 3 月 20 日

A. Grammy Wants to Earn Big Money

题目大意：dgg 会在周一到周五每天在杭州银行存一块钱，在周末学习理财知识。今天是周日，问包括今天的接下来 n $(n \leq 30\,000)$ 天 dgg 会存下来多少钱。

A. Grammy Wants to Earn Big Money

题解：直接模拟复杂度是 $\Theta(n)$，按照 $\mathrm{mod} 7$ 直接计算复杂度是 $\Theta(1)$。

B. Puzzle

题目大意：有两个长为 n $(n \leq 100\,000)$ 的数列 A, B，每次操作可以把 A 或 B shift 一位，问至少要多少次操作能让 A, B 对应位置相加之后等于数列 C。

题解：考虑一个数列 A 的，我们定义其 hash 为 $h(A)=\sum_{i=1}^{n} base^{n-i}A_i \bmod P$，假设两个不同的数列 hash 不同，那么我们发现数列 A,B,C 满足 $C=A+B \iff h(C)\equiv h(A)+h(B) \pmod P$。
那么我们预处理 B shift 若干位的 hash，用 std::map 维护，然后枚举 A shift 多少位，同时维护 $h(A)$，在 std::map 中查找 $h(C)-h(A)$ 来更新答案即可
注意根据概率我们选择一个 10^{18} 级别的大质数 P 可以避免 hash 冲突

C. Terrible Additive Number Theory Problem

题目大意：假设 P_i 是第 i 个质数.

求有多少个 x 满足 $x = \prod_{i=l}^{r} P_i = 2^k P_{r+1} - 1$, 其中 $l, r, k \in \mathbb{N}^+, 1 \le l \le r$,

且 $x \le n$, $(n \le 10^{18})$.

题解：10^{18} 以内满足条件的 x 只有 $2431 = 11 \times 13 \times 17 = 19 \times 2^7 - 1$，那么只要在 $n \geq 2431$ 时输出 1，其他情况输出 0。
这里介绍一种确定 10^{18} 之内的解的方法。首先如果是 3 个以上连续质数相乘，那么有 $P_l \leq 10^6$，只要枚举每个 10^6 之内的素数作为 P_l，然后暴力枚举往后乘几次即可。
如果左边只有一个质数那么是无解的，现在考虑 $P_i P_{i+1} = 2^k P_{i+2} - 1$ 的情况，这种情况我们枚举 k，因为 10^{18} 之内的质数最大间隔不超过 1500，那么在 2^k 左右枚举 10000 个数就可以保证枚举到所有可能的情况。
复杂度 $O(n^{\frac{1}{3}})$

D. Interstellar

题目大意：有 n 个空间站，dgg 现在在 1 号空间站，目标是到达 n 号空间站。第 i 个空间站的能量系数是 a_i。每次 dgg 可以从空间站 i 跳跃到空间站 $i < j \leq n$ 的代价是 $(j-i)^3 \times a_j$，求到 n 号空间站的最小代价。

D. Interstellar

题解：设 f_i 是从 1 号空间站到达 i 号空间站的最小代价，那么有

$$f_i = \max_{j=1}^{i-1} f_j + (i-j)^3 \times a_i$$

注意到如果 $(i-j)^3 \times a_i > \sum_{k=j}^{i} a_k$ 那么就可以不转移。

假设 n 和 a_i 同阶，最极限的情况下需要转移的有 $(i-j)^3 \le n(i-j)$，即 $i-j \le \sqrt{n}$，那么这么枚举的复杂度是 $\Theta(n^{\frac{3}{2}})$，有些枚举的方法复杂度是 $\Theta(n^{\frac{5}{3}})$ 都可以通过。

E. Recharge Cycle

题目大意：有四个角色，第 i 个角色的技能需要 c_i 能量，并且释放之后能为第 j 个人充能 $a_{i,j}$ 能量。初始都是满能量：问是否存在一种方法可以循环释放技能，并且循环的是一个排列。

题解：首先存在一种简单的解法，直接枚举循环的排列，容易证明循环如果能达到两轮那就可以循环，复杂度 $\Theta(8 \times 4!)$。注意到每个人在两轮循环之间恰好一定吃到四个人充能一次，所以只要判断 $\forall i, \sum_{j=1}^{4} a_{j,i} \geq c_i$ 即可，复杂度 $\Theta(4^2)$。

F. Tree Game

题目大意：给定一棵 n $(n \leq 300\,000)$ 个点的树，要求从其中选出一个 k 个点的子集染黑，满足至多有一个黑点的邻居有至少一个白点和至多一个黑点。

F. Tree Game

题解：首先如果一个黑点的黑邻居不超过一个，那么他是所在黑连通块的一个叶子。所以我们不妨让所有黑色点是一个连通块，那么要求就相当于要求黑连通块只有一个非原树叶子的黑连通块叶子。容易发现对树进行 dfs 过程中，已经访问过的节点构成的连通块只有一个不非原树叶子的该连通块的叶子，那么我们只要选择原树的一个叶子作为根进行 dfs，dfs 到 k 个点之后输出即可。注意 $n = 1$ 的 corner case。复杂度 $\Theta(n)$

G. Resource Calculator

题目大意：实现一个类似米游社养成计算器的程序

G. Resource Calculator

题解：模拟即可，有一些可以简化模拟的小 trick 比如可以把表格复制下来然后实现一个 parser 生成包含对应常量数组的 C++ 代码。另外一个需要注意的地方是经验书数量的计算。

H. Minimum Edit Distance

题目大意：给两个串 S, T，要找到一个串 R 使得到 S, T 的编辑距离之和最小。可以进行的操作有插入，删除和修改。

题解：注意到操作是可逆的，所以有 $d(S,S) + d(S,T) \leq d(S,R) + d(R,T)$，否则因为操作可逆那么我们顺着 $S \to R \to T$ 的操作路径可以得到一个更小的 $d(S,T)$，和定义相悖。所以我们任取 $R = S$ 或 $R = T$ 即可。复杂度 $\Theta(n)$

I. Rounding Master

题目大意：给一个目标 n 和一个次数 k $(1 \le n, k \le 10^{18})$，求一个最小的非负实数 q 使得从 1 开始每次乘 q 之后四舍五入到整数，进行 k 次操作之后 $\ge n$。

I. Rounding Master

题解：首先特判 $n=1$ 的 case，答案为 0.5。
之后讨论 $n \geq 2$ 的情况，注意到如果先让这个数变大，那么 $q \geq 1.5$，而且答案具有二分性，那么我们直接在 $[1.5, n-0.5]$ 上二分答案即可。复杂度 $O(\log^2 n)$。
需要注意的一点是，用 $r-l \leq \epsilon$ 来作为二分的结束条件有可能导致 Time Limit Exceeded，因为在 l, r 很大时可能导致数据类型的精度不够。可以采用判断相对误差的方式或固定二分的次数。

J. Mountain

题目大意：dgg 在爬山，可以用 1 点体力爬高度为 a，或者用 b 点体力跳高度为 c，如果当前体力小于 b 点且不为 0，那么还可以跳一次高度为 c，问 S 点体力最多爬多少。

J. Mountain

题解：最优方案一定是以下三个之一

- 一直花费 1 点体力爬高度为 a
- 一直花费 1 点体力爬高度为 a，直到还剩 1 体力跳一次高度 c
- 一直花费 b 点体力跳高度为 c，体力不足 b 之后不断花费 1 体力爬高度 a 到还剩 1 点体力，跳一次高度 c

分别计算取 $\max$ 输出即可。复杂度 $\Theta(1)$。

K. Pairing Game

题目大意：给一个长度为 n $(n \le 300\,000)$ 的括号序列，定义括号序列的价值是每一对括号包含的括号数量，每次删除一个括号之后输出括号序列的价值。

K. Pairing Game

题解：注意到括号序列构成一个树形结构，每次删掉一个括号的贡献 $\delta = -($ 在该括号到根路径上还存在的括号数 $+$ 该括号子树内还存在的括号数 $)$，那么可以求出 dfs 序用树状数组或线段树维护。
也可以直接维护括号序列，一对括号 (l, r) 包含的括号数是在它们之间的括号数，包含他们的括号数可以容斥计算，等于 (剩余括号数 $-$ 在 l 左边的右括号数-在 r 右边的左括号数)，也可以用树状数组或线段树维护。
时间复杂度 $\Theta(n \log n)$。

L. Race

题目大意：给一个 n 个点 m $(n, m \leq 200\,000)$ 条边的无向图，边有 k $(k \leq 30)$ 种颜色，有 q 个询问，每次询问是否存在一条从 S 到 T 的路径满足路径上边的颜色集合中，所有颜色出现次数相同。

L. Race

题解：如果 $S = T$，那么答案为 Yes。接下来讨论 $S \neq T$ 的情况，首先 S, T 要处于同一连通块，并且该连通块要包含所有颜色。我们注意到对于该连通块中任意一种颜色 c，我们可以在不改变 c 出现的奇偶性的情况下无限增加 c 出现的次数。假设原来的路径为 $S \to T$，那么考虑一条边 (u, v, c)，我们将路径修改为 $S \to x \to u \to v \to u \to v \cdots \to u \to x \to T$，可以发现每种颜色出现的奇偶性没有变，并且 c 这条边出现的次数可以随意增加。

L. Race

因为可以随意增加，那么问题变成了是否存在一条从 S 到 T 的路径使得每种颜色出现的奇偶性相同。
我们将颜色 k 对应为 2^k，那么改变奇偶性操作就变成了异或。对于每一个环，我们用类似的方式将环包含到一条路径中，不妨设环上的异或和为 C，原路径异或和为 x，从原路径到环的路径异或为 y，那么将这条环加入原路径之后，异或和变为 $x \oplus y \oplus C \oplus y = x \oplus C$。

L. Race

如果我们找出了所有环，其异或和构成的集合为 S，我们任意找一条 S 到 T 的路径，异或和为 x，那么任意一条路径都可以通过该路径经过若干个环得到，那么问题就变为是否存在 S 的一个子集 T，使得 $x \oplus (\oplus_{w \in T} w) = 0$ 或 $x \oplus (\oplus_{w \in T} w) = \text{allmask}$，其中 allmask 代表全为 1 的 mask。这个问题可以通过求出 S 的线性基来完成。
所有环的异或和集合和简单环的异或和集合是等价的，所以可以任意找一棵 dfs 树，其每条返祖边都构成了一个简单环。将这个环加入到该连通块的线性基即可。
复杂度 $\Theta(n + (m + q)k)$。

M. A Chinese Idiom

题目大意：给一个半径为 r_1 的西瓜，瓜皮厚度为 d。问一个更大的西瓜，半径为 r_2，西瓜皮至少为多厚才能使得西瓜皮比小西瓜的西瓜皮小。西瓜瓤和西瓜都视为完美球体。

M. A Chinese Idiom

题解：设答案为 d'，考虑极限情况即西瓜皮大小相等，有

$$r_1^3 - (r_1 - d)^3 = r_2^3 - (r_2 - d')^3$$

解方程直接输出即可。复杂度 $\Theta(1)$

谢谢！