2022 东北四省赛
题解

Prepared by Beijing University of Posts and Telecommuncations

May 22, 2022

A. Encryption

- 这是一个递归函数，我们考虑递归的最深一层，也就是 $f(t_m) = t_m$, 其实我们可以发现该函数描述的问题，是从后往前看询问串的每一个字符，如果当前的答案串在 s 中出现的次数为偶数，则将当前遍历到的字符加到答案串的前面，为奇数则加到后面。
- 先对 s 串求一下 SA, 始终维护着当前答案串，在 SA 中的区间。考虑加到后面的情况，只需要在原来区间的左右端点分别向内进行二分，即可维护加到后面后新答案串在 SA 中的区间。对于加到前面的情况，其实有很多处理办法，这里说一种简单的处理方法，就是先将 s 串的所有后缀按首字母分一下类，每一类中在按去掉首字母后的字典序排序一下，
- 然后每次我们就可以在当前枚举到的字符类别后缀里直接进行二分，找到新的答案串在 SA 中的区间。
- 使用 $O(n\log n)$ 的倍增算法构造后缀数组，时间复杂度 $O((|s| + \sum|t|)\log|s|)$。
- SAM 也可以做，但可能会超出本题时空限制，且解法可能会更加麻烦些，这里就不介绍了。

B. Capital Program

- 本题做法较多，大家可以赛后去看一下其他的通过代码。这里提供一种：
- 考虑如果 $k = 1$，我们肯定选择树的直径的中点作为首都，现在要多选一些点则是要尽量选，则可以贪心的选和目前首都有边的点中子树深度最深的。
- 求出树的直径中点（若有多个，则任取一个）*root*，以 *root* 为根进行 *dfs*，求出每个点的子树的深度，然后用优先队列维护当前与首都块有边的点中子树最深的点即可。
- 时间复杂度：$O(n \log n)$。

C. Segment Tree

- 首先，如果 $k \geq m$，那么显然能覆盖线段树所有的点，答案为 $2m-1$。
- 一棵线段树的形状一定是一个深度为 w 的满二叉树，然后选择某些深度为 w 的叶子结点延伸出左右儿子。
- 对于这样的二叉树 T，设第一次操作 T，得到了 x 个新点。那么第二次得到的新点个数必定小于 x，因为根节点在第一次被统计了，而第二次没有。而对于任意的这样的二叉树 T，左右儿子的深度差最多不超过 1。因此，肯定存在一个最优解，使得操作是在左右子树中轮换的。

C. Segment Tree

贪心

- 设 i 为满足 $2^i \le k$ 的最大值。
- 对于深度在 $\le i$ 的节点，一定都能被覆盖。
- 而以深度为 $i+1$ 的节点为根的子树形态最多两种，且 $2^{i+1} > k$，也就是不是所有的子树都能被选到。由上述性质，每个子树最多选一个点。
- 统计出以深度为 $i+1$ 的节点为根的两种子树形态的方案数。这两种形态的深度最多差一，那么贪心选择即可。
- 时间复杂度 $O(T \log m)$。

- 设 $f(m,k)$ 表示 k 次操作大小为 m 的线段树得到的最大值。
- 由上述性质，操作是在左/右子树中轮换的。那么就是左右子树几乎平分 k 次操作。当 m 为奇数时，左右子树的大小分别是 $\frac{m-1}{2}, \frac{m+1}{2}$。由贪心可知，如果 k 也为奇数，那么将 $\frac{k+1}{2}$ 次操作分给 $\frac{m+1}{2}$。
- 于是就有：$f(m,k) = 1 + f(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{k}{2} \rfloor) + f(\lfloor \frac{m+1}{2} \rfloor, \lfloor \frac{k+1}{2} \rfloor)$。
- 一个数 n 每次除以二，可以选择上取整或者下取整。除了 i 次以后，不同的取值最多为两种。
- 因此，有用的状态总数不超过 $2(\log m + \log k)$。使用记忆化 DP 即可。
- 时间复杂度 $O(T\log m)$。

D. Game

Lemma

设 $f(x)$ 表示数字 x 在区间 $[l, r]$ 的出现次数。
先手必败当且仅当 $\forall x, f(x) \equiv 0 \pmod 2$。

证明:

若 $\forall x, f(x) \equiv 0 \pmod 2$，则先手必败。

- 使用归纳法。首先，当数字全为 0 时，先手必败。
- 当满足上述条件时，轮到 A 操作，设 A 操作第 a_k 堆石子并分配给第 a_j 堆 b_j 个石子，其中 $j \in [l, r], b_j \geq 0, \sum b_j = a_k$。
- 轮到 B 操作时，可以证明必定存在一种方案使得 B 操作完后，还是满足所有的 $f(x)$ 均为偶数。但总石子数减少了。
- 因此，此结论成立。

若 $\exists x, f(x) \equiv 1 \pmod 2$，则先手必胜。

- 只需证明，此时存在一种操作，使得所有的 $f(x)$ 变为偶数。
- 设当前序列为 $(a_1, a_2, \cdots, a_n)$。不妨设 $0 < a_1 \le a_2 \le \cdots \le a_n$
- 若 $n = 2k + 1$，则我们先完整地取走 a_n，然后对任意 $i (1 \le i \le k)$，在第 $2i - 1$ 堆上放置 $a_{2i} - a_{2i-1}$ 个石子。
- 这样做是可行的，因为 $\sum_{i=1}^{k} (a_{2i} - a_{2i-1}) \le \sum_{i=1}^{2k+1} (a_i - a_{i-1}) < a_{2k+1}$。
- 得到的状态是 $(a_2, a_2, a_4, a_4, \ldots\ldots, a_{2k}, a_{2k})$，所有 $f(x)$ 均为偶数。

若 $\exists x, f(x) \equiv 1 \pmod 2$，则先手必胜。

- 只需证明，此时存在一种操作，使得所有的 $f(x)$ 变为偶数。
- 设当前序列为 $(a_1, a_2, \cdots, a_n)$。不妨设 $0 < a_1 \le a_2 \le \cdots \le a_n$
- 若 $n = 2k$，从第 n 堆上取走 $a_n - a_1$ 个石子，易知 $a_n - a_1 > 0$。
- 然后对任意 $i(1 \le i \le k-1)$，在第 $2i$ 堆放置 $a_{2i+1} - a_{2i}$ 个石子。
- 类似 n 为奇数的证明，这样做是可行的。
- 得到的状态是 $(a_1, a_3, a_3, a_5, \ldots\ldots, a_{2k-1}, a_{2k-1}, a_1)$，所有 $f(x)$ 均为偶数。

- 因此，若所有数的出现次数为偶数，则先手必败。
- 使用莫队，则可以很方便地统计出每个数的出现次数，和是否出现奇数次的出现次数了。
- 时间复杂度 $O(n\sqrt{m})$。

E. Plus

Observation

当 $n \geq 3$，可以找到唯一的一对 $p = 2, q = 3$，否则无解。

- 若 p, q 都是奇素数或都是 2，$p^q + q^p$ 一定是大于 2 的偶数，不可能是素数。
- 不妨设 $p = 2$，注意到 q 是奇数，$2^q = (-1)^q = -1 \pmod 3$，若 $q \neq 3$，则 $q = 1 \pmod 3$ 或 $q = 2 \pmod 3$，则 $q^2 = 1 \pmod 3$。可以得到 $p^q + q^p = -1 + 1 = 0 \pmod 3$，当 $q = 3$ 时，$p^q = q^p = 2^3 + 3^2 = 17$。
- 注意开 long long。

F. Tree Path

- 首先对于每个询问二分答案 r，然后判断最小的权值是否可能小于等于 r。
- 若最小的权值不小于 r，则说明所有权值小于等于 r 的路径都包含 u，即这些路径的交集一定不为空且包含询问的顶点。
- 然后可以注意到路径的交依然为路径，而求两条路径的交可以做到 $O(1)$，或者 $O(\log n)$ （与实现求解 lca 算法的复杂度级相同）。
  考虑要求两条路径的交集 $(p, q), (u, v)$，可以分类讨论且方法很多，这里提供一种方法参考：

两条路径相交有三种情况:

- 不相交
- 交集部分完全落于 (p, q) 以 lca 为分割的某一半内
- 交集穿过 (p, q) 的 lca

容易发现这三种情况中，若将 $(p, q), (u, v)$ 两两求 lca，则得到的 4 个结果中，一定存在两个结果为 $lca(p, q, u, v)$，而剩余的两个结果 (无论是不是 $lca(p, q, u, v)$) 满足:

- 若 $(p, q), (u, v)$ 相交, 则这两个结果即求得的交集路径的两端
- 若 $(p, q), (u, v)$ 不相交, 则这两个结果相等，且这个结果指向的顶点一定在 lca 较高的那条路径上，且是 lca 较低路径的 lca 的某个祖先（即，这个顶点不在 lca 较低的路径上）

- 最后二分部分可以直接在线段树来维护二分结构。
- 对于本题，由于每次一定删去权值最小的路径，故可以用倍增来替代二分结构。
- 若用单次查询复杂度为 $O(\log n)$ 的算法求 lca，则时间复杂度 $O((n+k)\log n + (m+k)\log k \log n)$。
- 使用 ST 表 +RMQ 求 lca，则时间复杂度 $O(n\log n + k\log k + m\log k)$。

G. Hot Water Pipe

- 单独考虑水管中某个单元中的水温，其随时间的变化曲线为是一个以 $(T_{max} - T_{min} + 1)$ 为周期的函数。
- 因此可以方便地根据时间计算出该单元的水温，而不需要时刻维护每个单元中的水温。
- 若所有操作的用水总量不超过 n ，那么使用队列，维护当前操作对应的时间，即可在 $O(n + m)$ 的时间内计算出答案。
- 为了处理用水总量超过队列总量的情况，对于每个用了 k 单位体积的操作，在队列末端加入 k 单位体积，并记录相应周期函数在零时刻的温度。由于这些水的温度一致且连续，故合并成一个元素处理即可。
- 因此，队列里需要记录的数值为：当前这一段的长度，已经当前这一段相应的周期函数在零时刻的温度。
- 队列元素最多有 $n + m$ 个。在处理操作的过程中，队列中的每个元素要么被弹出，要么导致该次操作结束。势能分析可知，时间复杂度为 $O(n + m)$ 。

先不管字典序的限制，先求最小值。显然这是一个费用流模型：

- s 向 $i(0 \in [0,9])$ 连边，容量为 i 在字符串的出现次数，费用为 0。
- i 向 $i+1$ 连边，$i+1$ 向 i 连边。容量为 ∞，费用为 1。
- i 向 t 连边，容量为 m，费用为 0。

跑费用流即可找到最小值。

对于字典序的限制，我们从小到大依次枚举第 i 个数能否填上 j，然后还是类似地，求若干遍费用流。

需要做 $O(n\Sigma)$ 遍费用流，过不去。($其中 $\Sigma = 10$)

H. Digit String

优化-模拟费用流

- 考虑优化费用流的复杂度。
- 这个相当于经典的老鼠进洞模型。数轴上每个点有若干只老鼠和若干个洞。让老鼠进一个洞花费的代价是他们的距离。问最小代价和。
- 使用两个堆来记录当前的老鼠和洞。堆中的元素 (x, y) 表示 x 个老鼠或洞，选一个花费 y 的代价。以 y 为关键值排序。
- 遇到老鼠 a 时，考虑贪心地选择前面的洞 b，以及注意后面的洞可能会来抢这只老鼠，因此添加一个价值为 $-b-2a$ 的老鼠。
- 遇到洞 b 时，选择代价最小的老鼠 a，产生的贡献是：$b+a$。
  - 考虑后面有老鼠抢走这个洞，添加一个代价为 $-a$ 的洞。
  - 考虑后面有个洞来争夺这只老鼠，同理。

时间复杂度 $O(n\Sigma^2 \log \Sigma)$。

I. Generator

- 设 $f[i]$ 表示当前显示的数字为 i，期望多少次第一次显示 1。
- 当 $n \geq 2$ 时，有：$f[n] = \frac{0+f[2]\cdots+f[n]}{n} + 1$，移项得：$nf[n] = (0 + f[2] + \cdots + f[n]) + n$。
- 同理 $(n-1)f[n-1] = (0 + f[2] + \cdots + f[n-1]) + n - 1$。两式相减并整理得：$f[n] - f[n-1] = \frac{1}{n-1}$。
- 令 $n = 2$，解得 $f[2] = 2$。
- 因此 $f[n] = 2 + \frac{1}{2} + \cdots + \frac{1}{n-1} = 1 + (1 + \frac{1}{2}\cdots + \frac{1}{n-1})$。求 $1 + \frac{1}{2}\cdots + \frac{1}{n}$ 是一个经典问题。
- 令 $T_n = \sum_{k=1}^{n} \frac{1}{k} - \ln n$，可以证明 $\lim_{n\to\infty} T_n$ 存在，设为 γ，称为欧拉常数，其近似值为 0.5772156649。可使用打表算出。
- 当 $n \leq K$ 时，$O(n)$ 去算。当 $n > K$ 较大时，可用 $1 + \frac{1}{2}\cdots + \frac{1}{n} \approx \gamma + \ln n$ 近似。这里取 $K = 1e7$ 即可。
- 由于 $n \leq 10^9$，使用分段打表的方法也可通过。

Observation

一定存在一种最优方案，使得其中没有任何一件 paper 是在做其他的 paper 中间开始的（中间的含义里不包括开始和结束)。

- 假设存在一种方案是完成时间最优的方案，但其中有若干件 paper 是在做其他的 paper 中间开始的，
- 先找到最后的这种情况的 paper，即前面存在正在做的 paper 若干，则我们可以将这些包含该 paper 的若干 paper 一起挪到后面来做。
- 这样不影响总时间，且对再后面的 paper 也没有影响。
- 以此类推，可以继续往前找最后的这种情况的 paper。
- 综上，此结论成立。
- 有了这个结论，现在的问题就转化成了一个背包问题。

J. Papers

给定一个容量为 n 的背包，m 个物品，第 i 个物品重量为 i，价值为 a_{i-1}。问装满背包的最小价值。

Observation

设 $\frac{a_{i-1}}{i}$ 取到最小值的位置是 i。则当 $n > m^2$ 时，则一直取这个 i，直到 $n \le m^2$。

- 设第 j 个物品选了 b_j 次。
- 如果存在某个序列 b'，使得 $b'_i \le b_i$，且 $\sum_{j \ne i} j \times b'_j$ 为 i 的倍数，那么则可以用若干物品 i 代替之，不会使答案更劣。
- 假设存在某个 k，使得所有装满容量为 k 的背包的最优方案中，和物品 i 没有关系。
- 设 $\sum_{j \ne i} j \times b_j = k$ 为一种最优的选法。此时，不存在这样的 b'。
- 由鸽巢原理，$\sum_{j \ne i} j \times b_j$ 最大为 $m \times (i-1) < m^2$。
- 当 $n < m^2$ 时，使用完全背包 DP 计算。当 $n \ge m^2$ 时，一直选 i 即可。
- 时间复杂度 $O(m^3 + T)$。

K. Maze

- *bfs*，记 $d[i][j][k][0/1/2/3]$ 表示现在在位置 (i,j) 已经连续在 $0/1/2/3$ 方向上移动了 k 次。
- 然后直接进行 *bfs* 即可。
- 时间复杂度：$O(n^2m*4)$.

L. Polygon

- 签到题。
- 设 mx 为所有线段的最大值，判断是否有 $\sum_{i=1}^{n} a_i - mx > mx$ 即可。
- 时间复杂度 $O(n)$ 或 $O(n \log n)$。

M. Spiral

- 设 ω 为 k 的一次单位根 $cos\frac{2\pi}{k} + i\sin\frac{2\pi}{k}$， $d = \lfloor \frac{k}{2} \rfloor$，则答案为以下所有数的和：

$$
\begin{array}{ccccc}
1 & 1\omega & \cdots & \cdots & 1\omega^{d-1} \\
2\omega^{d} & 2\omega^{d+1} & \cdots & \cdots & 2\omega^{2d-1} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
s\omega^{(s-1)d} & s\omega^{(s-1)d+1} & \cdots & \cdots & s\omega^{sd-1} \\
(s+1)\omega^{sd} & \cdots & (s+1)\omega^{sd+i} & x\omega^{sd+i+1} &
\end{array}
$$

- 其中 $\sum_{j=1}^{s} d \times j + (s+1) \times (i+1) + x = n$。
- 分成两部分计算。

M. Spiral

Part One

$$
\begin{array}{ccccc}
1 & 1\omega & \cdots & \cdots & 1\omega^{d-1} \\
2\omega^{d} & 2\omega^{d+1} & \cdots & \cdots & 2\omega^{2d-1} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
s\omega^{(s-1)d} & s\omega^{(s-1)d+1} & \cdots & \cdots & s\omega^{sd-1} \\
(s+1)\omega^{sd} & \cdots & (s+1)\omega^{sd+i} & x\omega^{sd+i+1} &
\end{array}
$$

- 第一部分是除最后一行的和。
- 设 $S = \sum_{i=1}^{s} i(\omega^{d})^{i-1}$，则答案为 $S \times (1 + \omega + \cdots + \omega^{d-1})$。
- S 的计算是高考数学数列题。$\omega^{d} \times S = \sum_{i=1}^{s} i(\omega^{d})^{i}$，$S = \sum_{i=1}^{s} i(\omega^{d})^{i-1}$。
- 两式相减后，通过等比数列求和即可算出。

$$
\begin{array}{ccccc}
1 & 1\omega & \cdots & \cdots & 1\omega^{d-1} \\
2\omega^{d} & 2\omega^{d+1} & \cdots & \cdots & 2\omega^{2d-1} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
s\omega^{(s-1)d} & s\omega^{(s-1)d+1} & \cdots & \cdots & s\omega^{sd-1} \\
(s+1)\omega^{sd} & \cdots & (s+1)\omega^{sd+i} & x\omega^{sd+i+1} &
\end{array}
$$

- 第二部分是最后一行的和。这个就是一个简单的等比数列求和。
- 由于精度的问题，最后需要输出 $x+y$ 来减小误差。
- 时间复杂度 $O(T)$。

Thank you!