2023 SYSU School Contest

Solution

October 14, 2023

A. SUN YAT-SEN University

题意

给你一个字符串，求出有多少个子串的子序列包含“sysu”。
$|S| \leq 10^6$

A. SUN YAT-SEN University

枚举这个子串的起始位置 l，考虑 r 什么时候开始满足条件。
从 l 出发找到第一个 s 的位置 a_1，然后找到 a_1 之后的第一个 y，$\cdots$。
记录 $next_{i,j}$ 表示从位置 i 开始找到第一个 j 的位置，然后依次跳到第一个合法的 r 即可。
时间复杂度 $O(|S|)$。

B. Lowest Common Ancestor

题意

给你一棵 n 个点的树，对于所有的 k，询问随机 k 个点的 lca 深度的期望模 998244353 的结果。
$n \le 5 \times 10^5$。

B. Lowest Common Ancestor

统计 lca 深度的期望，可以算每个点 u 作为 lca 的祖先的贡献。也就是 $\binom{siz_u}{k}$。
对于每个 k，答案就是 $\sum_{u=1}^{n} \binom{siz_u}{k}$。
直接算，复杂度是 $O(n^2)$ 的，需要进一步优化。
考虑有多少个点 u，满足 $siz_u = i$，记为 a_i。答案可以写成 $[x^k] \sum a_i (1+x)^i$。
设 $f(x) = \sum a_i (1+x)^i = \sum f_i x^i$，那么 $f_i = \sum a_j \binom{i}{j}$
用 FFT 求解，复杂度 $O(n \log n)$。

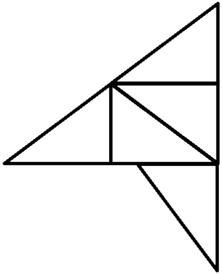
C. Triangle

题意

给你 n 个边长为 $3, 4, 5$ 的直角三角形，问是否能在不重合的情况下拼成一个轴对称的图形。

C. Triangle

首先 n 等于偶数一定可以，只需要两两配对拼成一个长方形即可。
然后考虑 n 为奇数的情况，当 $n = 5$ 时，考虑如下构造：



同时，当 n 为大于 5 的奇数时，在对称轴上放若干个由两个三角形组成的长方形即可。

D. Cross the Storm

题意

给你一个图，i 向 $i+1$ 连有向边，0 向 i 连无向边，有边权。
第 i 天随机断掉 $1 \sim i$ 内的某个区间的有向边，问从 1 到 i 的期望最短距离。
$n \leq 5 \times 10^5$

D. Cross the Storm

先考虑某一天（第 k 天），假如没有摧毁任何一条边，那么这个最短路可以通过提前预处理得到。需要注意的是有 k 种情况都是不会摧毁任何边的。
假如有某个区间的边被摧毁了，那么最短路一定形如：先从 1 出发走一段海路，然后在某个位置 x 走空路去空岛，然后再从空岛走空路去到某个点 y，然后从这个点一直走到 k。
记 w_i 的前缀和为 s_i，假如被摧毁的是 l 到 r 岛屿之间的所有航路，那么 x 一定是 $i \in [1, l]$ 中 $s_{i-1} + W_i$ 最小的 i，类似地，y 是 $i \in [r, n]$ 中 $s_{n-1} - s_{i-1} + W_i$ 最小的 i。
对于左端点为 l 的摧毁区间，x 肯定都是 $i \in [1, l]$ 且 $s_{i-1} + W_i$ 最小的 i，只需要从左到右扫的时候维护出每个 l 对应的 x，计算一下贡献即可。
y 的维护是类似的，按 $-s_{i-1} + W_i$ 的值维护一个单调递增的栈即可。栈中每个 y 可以贡献它到它上一个位置这一段，贡献是一个等差数列，很好容易计算。
时间复杂度 $O(n)$。

E. Travel

题意

给你一个有向无环图，有点权 a_i。一条路径合法当且仅当连续三个点的点权和大于 k，问是否存在一条合法的从 1 到 n 的路径。

$n, m \leq 3 \times 10^5, k \leq 10^9, a_i \leq 10^9$。

E. Travel

注意到，对于某一条合法路径 $a_1, a_2, ..., a_k$，这条路径要接着延伸下去的话，对于 a_k 后面的点，其实有用的信息只有 $a_k + a_{k-1}$ 和 a_k 的大小。
对于所有以 u 为终点的路径，设每条路径的 $a_k + a_{k-1}$ 和 a_k 分别为 x, y，假如对于路径 A, B，存在 $x_A \geq x_B, y_A \geq y_B$，那么沿着 B 路径接着走能走到的点，A 路径也一定能走到，所以可以直接丢掉路径 B。
于是考虑按拓扑序处理，对于当前点 u，丢掉没用的路径，剩下的路径按 x 升序排列后，y 一定恰好是降序排列。假如存在边 $u \rightarrow v$，那么寻找一条 y 尽可能大且合法的路径贡献给 v 即可（记为 $y_{\max}$），v 就获得了一条 x, y 为 $y_{\max} + a_v, a_v$ 的路径，二分寻找这个路径即可。
时间复杂度 $O(n \log m)$。

F. Four K3

题意

给你一个图，求出与一个 K_3 的每条边再分别长出一个 K_3 的图同构的子图个数。
$n, m \leq 10^5$。

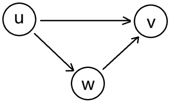
F. Four K3

首先，通过将 n 点按度数排序然后建有向图的方法，可以求出在 $O(m\sqrt{m})$ 内求出包含每条边的 K_3 个数 c_i。
然后我们对上述过程再来一遍，计算出每个三元环的 $\sum(c_x-1)(c_y-1)(c_z-1)$ 的和。
但是，这样会算到不合法的方案。不合法的方案一共有两种，分别是有两个点重了，三个点重了，都需要减去。
三个点相同的方案是一个 K_4，可以使用 $O(m\sqrt{m}+\frac{m^2}{w})$ 的做法求出所有 K_4。
两个点相同的方案是一个 K_4 和 K_3 共享一条边的情况，可以在统计 K_4 的时候统计出经过每条边的 K_4 个数 d_i，然后算出 $d_i\times(c_i-2)$ 的和就好了。
总的时间复杂度 $O(m\sqrt{m}+\frac{m^2}{w})$，其中 w 指 bitset 的位数。

F. Four K3

求经过每条边的 K_3 个数

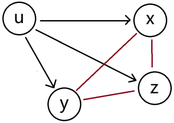
将每个点按照度数从小到大排序。新建一个有向图，每个点只会往在度数排名数组中比自己小的点连边。
这样就可以保证每个点的连出去的点不超过 $\sqrt{2m}$，因为 $\sum_{i=1}^{n} deg_i = 2m$，假设点 u 连出去 k 个点，连出去的点的度数都至少为 k，因此有 $k^2 \leq 2m$。



对于每个点 u，枚举 u 的每条出边的另一个端点 v，在 v 节点打上标记为这条边的标号。
然后再重新枚举点 u 的每条出边 (u, w)，然后再枚举 w 的出边的另一端点，如果该端点被标记过，则找到一个三元环。
时间复杂度 $O(m\sqrt{m})$。

F. Four K3

求经过每条边的 K_4 个数



还是按照度数排序，由上可知我们枚举两次出边的总边数是 $O(m\sqrt{m})$ 的，且每个点的出度不超过根号 $2\sqrt{m}$。
还是枚举 u，对于 u 的不超过 $2\sqrt{m}$ 个点 $a_1, a_2, \cdots, a_k$，需要统计这些点之间的三元环个数。
那么，对于每个 a_i，使用 bitset 统计出哪些 a_j 与之相邻，记为 b_i。枚举每条红色边 (x, y)，算出 $b_x \& b_y$ 的二进制 1 的个数即可知道经过每条边和每个点的三元环个数。
这样就可以算出这样一个结构中每条边的 K_4 个数。
bitset 大小是根号的，每次操作的复杂度是 $O(\frac{\sqrt{m}}{w})$。
因此总的复杂度是 $O(m\sqrt{m} + m\sqrt{m} \times \frac{\sqrt{m}}{w}) = O(m\sqrt{m} + \frac{m^2}{w})$。

G. Polynomial

题意

n 个数 c_i。对于每个 $i \geq j$ 的 (i,j)，问有多少个多项式满足系数均为非负整数且小于 c_i 且 $f(c_i) = c_j$。
m 次修改，每次修改一个 c_i 的值。输出每次修改后所有 (i,j) 的答案的和。

G. Polynomial

问题等价求对于所有 $i \neq j$，有多少个多项式满足 $f(c_i) = c_j$.
注意到要求多项式系数 $\in [0, c_i)$，所以满足 $f(c_i) = c_j$ 的多项式的系数，其实就是 c_j 在 c_i 进制下每一位的值。
如果 $c_i, c_j > 1$，那么因为 c_j 在 c_i 进制下表示是唯一的，所以只有一个满足条件的多项式。
如果其中一个为 0 或 1，那么就需要进行讨论。

- $c_i = 0$ 时，由于 $a_k \geq 0$ 且 $a_k < c_i = 0$，因此一定无解。
- $c_i = 1$ 时，只有 $c_j = 0$ 时才存在一个多项式，否则不存在。

设 x, y, z 分别为 0，1，> 1 的 c_i 个数，那么答案就是 $z \times (n-1) + y \times x$。
修改的话简单维护一下 x, y, z 即可。时间复杂度 $O(n+q)$。

H. Polygon

题意

n 根棍子，第 i 根棍子长度为 a_i。
问能否用其中的 k 根不同的棍子拼成一个简单多边形。
$n, k \leq 3000$。

H. Polygon

k 根棍子能拼成一个简单多边形，当且仅当较小的 $k-1$ 根棍子的长度之和大于最长的棍子的长度。

首先将 a_i 从小到大排序。

枚举最长的棍子 i，对于剩下的棍子，显然要贪心的选择比 i 小的最长的 $k-1$ 根。

使用暴力或者前缀和等方法判断即可。

I. Dislike

题意

定义一个排列的权值为所有区间最大值之和。
一个排列是好的当且仅当此排列经过若干次轮换后，权值取到所有排列的最小值。
问长度为 n 的好排列个数。

I. Dislike

先说结论，答案就是 $n \times 2^{n-2}$。
首先考虑什么样的排列的 G 值是最小的（也就是等于 K），或者说，如何调整一个排列，能使得其 G 值变小。

Theorem

如果一个 $1 \sim n$ 的排列中 n 不在第一位或最后一位，那么将其移到第一位或最后一位一定能使 G 值变小。

Proof.

考虑所有包含 n 的区间，将其分成两类，A 类为 n 在区间的端点处，B 类为 n 不在区间的端点处，显然 A 类区间一定有恰好 n 个。将 n 移到第一位或最后一位后，A 类区间数量不变，所以对 G 值贡献不变，而 B 类区间变成 0 个。具体来说，原本不包含 n 的区间的贡献都不变，而原来的 B 类区间相当于把 n 拎走了，那么这些区间的最大值一定会变小，所以 G 值一定严格变小了。□ □

I. Dislike

所以要使 G 值最小，首先要让 n 放在第一位或最后一位，剩下 $1 \sim n-1$ 可以递归考虑。那么放完之后，整个排列形如一个山谷，即从 1 处断开，左边是一个单调递减序列（可能为空），右边是一个单调递增序列（也可能为空）。
所以如果没有 rotate 操作，考虑 $2 \sim n$ 每个数在 1 的左边还是右边，就可以得到答案为 2^{n-1}。
有了 rotate 时，设将一个序列 rotate $0 \sim n-1$ 次得到的序列们构成的集合称为一个**循环**。则对于一个 G 值最小的排列（且 n 在序列最左边），他所在的循环中一定恰好有两个 good 的排列，即它本身以及它 rotate 一次后得到的排列。它本身显然不用说，它 rotate 一次后得到的排列，仅仅是把 n 从最左边移到了最右边，不难发现 G 值不发生改变，所以也是 good 的。
于是不妨先统计有多少个 n 在最左边且 good 的排列，由上可知为 2^{n-2}，再乘上一个 n，也就是他所在循环的大小，就是答案了。
注意 $n = 1, 2$ 需要特判。

J. Count

题意

定义一棵 n 个点的树的权值为 $\sum_{1\le i<j\le n} dis(i,j)$，其中 $dis(i,j)$ 表示 i,j 之间的最短距离。
给定 n，求所有有根树的权值形成的集合。
$n \le 20$。

J. Count

考虑将 $\sum_{1\le i<j\le n} dis(i,j)$ 转换成对于每条边的贡献。
每条边的贡献是 $siz_u \times (n - siz_u)$，其中 siz_u 表示断掉这条边后该边其中一个节点的子树大小。
考虑 DP，设 $f_{i,j}$ 表示 i 个节点的树中，$i-1$ 条边的 $siz_u \times (n - siz_u)$ 之和为 j 是否合法进行转移。
可以使用 bitset 优化转移。

K. Nim X2

题意

n 堆石子，两个人轮流操作。每次选一堆的若干个石子并拿走，拿走后，每一堆石子个数变为原来的 2 倍。
谁先拿走最后一颗石子谁获胜。
求最优策略下谁会获胜，或者是平局。

K. Nim X2

如果 $n = 1$，显然先手必胜。
如果 $n = 2$，已知 $n = 1$ 时先手必胜，所以谁都不会想拿完一堆石子，于是会一直保持 $n = 2$，于是就平局了，也就是输出 `draw`。
但是如果石子数是 $1, 1$，那么先手必然会拿完一堆，于是后手必胜。除此之外都是平局。
$n > 2$ 时则必然平局，因为经过若干轮变成 $n = 2$ 时，石子数一定不会是 $1, 1$，所以一定平局。

L. Function

题意

记函数 a 为满足对于所有的正整数 x，有 $a(a(x)) = dx$，且 $a(1), a(2), \cdots$ 单调递增且字典序最小的函数。
给定 d, x，求 $a(x)$ 的值。
10^5 组数据，$d, x \le 10^9$。

L. Function

考虑贪心，首先有 $a(1)=2$，对于下一位 x 来说，如果还没确定，那就令 $a(x)=a(x-1)+1$ 以保证字典序最小。
对于每一个确定的 $a(x)$，$a(a(x))$ 的值是确定的，为 $d\times x$。
可以发现：

Theorem 3. *Let d be an integer ≥ 4. Define the sequence $\mathbf{a}=(a(n))_{n\geq 0}$ by*

$$a(0)=0, \quad a(1)=2, \quad a(d-1)=2d-3$$

and for all $k\geq 0$, $i\in[0,d^k-1]$, we have

$$\begin{aligned}
a(xd^k+i) &= (x+d-2)d^k+i, && \text{for all } x\in[2,d-2],\\
a(xd^k+i) &= (x-d+2)d^{k+1}+di, && \text{for all } x\in[d,2d-4],\\
a((2d-3)d^k+i) &= (d-1)d^{k+1}+i, &&\\
a((2d-2)d^k+i) &= (d^2-d+1)d^k+i, &&\\
a((2d-1)d^k+i) &= (d-1)d^{k+1}+a(d^k+i), &&\\
a(d(d-1)d^k+i) &= (2d-3)d^{k+1}+di, &&\\
a((d^2-d+1)d^k+i) &= (2d-2)d^{k+1}+di, &&\\
a(xd^k+i) &= (d^2+x-2)d^k+i, && \text{for all } x\in[d^2-d+2,d^2-2],\\
a((d^2-1)d^k+i) &= 2(d-1)d^{k+1}+a((d-1)d^k+i). &&
\end{aligned}$$

可以通过找规律等方法，找到 $a(n)$ 的值。
相关论文：On integer sequences whose first iterates are linear。