2019 计蒜之道 复赛 题解

外教 Michale 变身大熊猫

考虑每个数在几个最长上升子序列上

维护一下 f_i, g_i 表示从左至右以 i 的最长上升子序列长度为 f_i，数目为 g_i

从右至左再维护一个一样的东西 $f_i^{'}, g_i^{'}$

如果 $f_i + f_i^{'}$ 等于最长上升子序列的长度$+1$，就说明 a_i 在 $g_i g_i^{'}$ 个最长上升子序列上

这个东西显然可以用树状数组容易地维护

复杂度 $O(nlog_2 n)$

个性化评测系统

枚举听牌，判断能不能胡，爆搜即可，每次枚举一个顺子删掉，最后判断剩下的牌是不是都三张或者两张或者没有，注意到各个类型的牌互不影响，可以分开搜索，提高效率。

个性化学习之石子游戏

考虑计算sg函数

尝试打表，发现$sg(x) = log_2(lowbit(x))$

用上一题的方式计算

然后考虑计算异或和为0的方案数，可以用FWT快速计算

复杂度$O(Llog_2 L)$

“星云系统”

设串长为 n，则只需删掉 $n-k$ 个字符。

用一个单调栈维护，依次将字符串的每个字符插入，如果当前删掉的字符不足$n-k$个并且栈顶元素$>$插入的元素，那么删掉栈顶，直至删掉的字符达到$n-k$或者满足单调栈的性质。

最后取栈里前 k 个字符输出即可。

撑起信息安全“保护伞”

前驱：找一个尽量靠后的位置pos，使得pos这个位置上是一个左括号，且$pos-1$这个位置上是一个右括号。则前驱中$pos-1$之前与原串相同，$pos-1$这个位置是左括号，pos这个位置是右括号，pos之后的位置在保证合法的前提下尽量使右括号靠前。

后继：找一个尽量靠后的位置pos，使得pos这个位置是一个左括号，$pos+1$这个位置是一个右括号，**且交换这两个位置得到的括号序列依然合法**。则后继中pos之前的位置与原串相同，pos这个位置是右括号，$pos+1$这个位置是左括号，$pos+1$之后的位置在保证合法的前提下尽量使左括号靠前。

多重积分

把大的正方体分割成各边长度都为1的小正方体

$$
\begin{aligned}
&2^m\iiint\limits_{\Omega}\cdots\int gcd(\lfloor x_1\rfloor,\lfloor x_2\rfloor,\dots,\lfloor x_m\rfloor)\prod_{i=1}^{m}x_idx_i\\
&=2^m\sum_{x_1=1}^{n}\sum_{x_2=1}^{n}\cdots\sum_{x_m=1}^{n}gcd(x_1,x_2,\dots,x_n)\prod_{i=1}^{m}\int_{x_i}^{x_i+1}x_idx_i\\
&=\frac{2^m}{2^m}\sum_{x_1=1}^{n}\sum_{x_2=1}^{n}\cdots\sum_{x_m=1}^{n}gcd(x_1,x_2,\dots,x_n)\prod_{i=1}^{m}(2x_i+1)\\
&=\sum_{k=0}^{m}2^kC_m^k\sum_{y_1=1}^{n}\sum_{y_2=1}^{n}\cdots\sum_{y_m=1}^{n}gcd(y_1,y_2,\dots,y_m)\prod_{i=1}^{k}y_i\\
&=\sum_{k=0}^{m}2^kC_m^k\sum_{d=1}^{n}d^{k+1}\sum_{y_1=1}^{\lfloor\frac{n}{d}\rfloor}\sum_{y_2=1}^{\lfloor\frac{n}{d}\rfloor}\cdots\sum_{y_m=1}^{\lfloor\frac{n}{d}\rfloor}\prod_{i=1}^{k}y_i[(y_1,y_2,\dots,y_m)=1]
\end{aligned}
$$

令$S_0(n)=n,S_1(n)=\frac{n(n+1)}{2}$

继续化简上式

$$\sum_{k=0}^{m}2^kC_m^k\sum_{d=1}^{n}d^{k+1}\sum_{y_1=1}^{\lfloor\frac{n}{d}\rfloor}\sum_{y_2=1}^{\lfloor\frac{n}{d}\rfloor}\cdots\sum_{y_m=1}^{\lfloor\frac{n}{d}\rfloor}\prod_{i=1}^{k}y_i\sum_{t|(y_1,y_2,\ldots,y_m)}\mu(t)$$
$$=\sum_{k=0}^{m}2^kC_m^k\sum_{d=1}^{n}d^{k+1}\sum_{t=1}^{\lfloor\frac{n}{d}\rfloor}t^k\mu(t)\sum_{y_1=1}^{\lfloor\frac{n}{dt}\rfloor}\sum_{y_2=1}^{\lfloor\frac{n}{dt}\rfloor}\cdots\sum_{y_m=1}^{\lfloor\frac{n}{dt}\rfloor}\prod_{i=1}^{k}y_i$$
$$=\sum_{k=0}^{m}2^kC_m^k\sum_{d=1}^{n}d^{k+1}\sum_{t=1}^{\lfloor\frac{n}{d}\rfloor}t^k\mu(t)S_1^k(\lfloor\frac{n}{dt}\rfloor)S_0^{m-k}(\lfloor\frac{n}{dt}\rfloor)$$

令$T=dt$，原式=

$$\sum_{k=0}^{m}2^kC_m^k\sum_{T=1}^{n}S_1^k(\lfloor\frac{n}{T}\rfloor)S_0^{m-k}(\lfloor\frac{n}{T}\rfloor)\sum_{t|T}t^k\mu(t)\frac{T^{k+1}}{t^{k+1}}$$
$$=\sum_{k=0}^{m}2^kC_m^k\sum_{T=1}^{n}S_1^k(\lfloor\frac{n}{T}\rfloor)S_0^{m-k}(\lfloor\frac{n}{T}\rfloor)T^k\sum_{t|T}\mu(t)\frac{T}{t}$$
$$=\sum_{k=0}^{m}2^kC_m^k\sum_{T=1}^{n}S_1^k(\lfloor\frac{n}{T}\rfloor)S_0^{m-k}(\lfloor\frac{n}{T}\rfloor)T^k\varphi(T)$$
$$=\sum_{k=0}^{m}2^kC_m^k\sum_{i=1}^{n}i^k\varphi(i)S_1^k(\lfloor\frac{n}{i}\rfloor)S_0^{m-k}(\lfloor\frac{n}{i}\rfloor)$$

交换i,k的次序继续化简，得到

$$\sum_{i=1}^{n}\varphi(i)\sum_{k=0}^{m}C_m^k2^ki^kS_1^k(\lfloor\frac{n}{i}\rfloor)S_0^{m-k}(\lfloor\frac{n}{i}\rfloor)$$

用二项式定理，得到原式=

$$\sum_{i=1}^{n}\varphi(i)(i\lfloor\frac{n}{i}\rfloor^2+i\lfloor\frac{n}{i}\rfloor+\lfloor\frac{n}{i}\rfloor)^m$$

m在指数上，可以用$phi(p)$降幂

这样可以得到一个$O(nlog_2p)$的做法

大熊猫的“树”?

设 $f[i][j]$ 表示大小为 j 的二叉树，所有 (子树的)*i 大小的和。

当 i为 0 时，表示所有大小为i的二叉树的大小和，也就是$Ca(i)*i$，$Ca(i)$表示卡特兰数的第i项。

当 i 不为 0 时，有$f[i][j]=f[i-1][j]+\sum_{k=0}^{j-1}2*f[i][k]*Ca[j-k-1]$，含义是枚举左子树大小，右子树有Ca(j-k-1)种方案，枚举右子树大小同理，所以乘2，这样只统计了非自己的子树，所以还要加上自己这颗子树大小的和，就是$f(i-1,j)$。

这样暴力dp即可得到一个复杂度为$O(n^2*k)$的做法。

考虑用生成函数优化，设 $[x^j]F_i$ 表示$f[i][j]$，$[x^i]C$ 表示 $Ca[i]$，可以列出递推式
$F_i=F_{i-1}+2xF_iC$

，其中 $C=(1-\sqrt{1-4x})/(2x)$ 。

整理得

$$F_i=F_{i-1}*(1-4x)^{1/2}$$

$$F_0=xC'=(1-4x)^{-1/2}-1/(2x)+(1-4x)^{1/2}/(2x)$$

也就是

$$F_i=(1-4x)^{i/2}*((1-4x)^{-1/2}-1/(2x)+(1-4x)^{1/2}/(2x))$$

答案就是$[x^N]F_K$，直接广义二项式定理展开即可得到答案，复杂度为
$O(Nmax+Kmax+T)$ 。

$ans=(-4)^i*(C((K+1)/(-2),N)+2C(k/(-2),N+1)-2C((k-1)/(-2),N+1))$

(答案里的C表示组合数)