爽快计数随手切的一眼（例）题

[agc005f]Many Easy Problems

- 给你一棵n个点的树，对于每个k求出在树上任选k个点形成虚树大小的和。

- n<=200000。

爽一爽

假设这是有根树。

考虑一个点i对于一个k的贡献，考虑容斥。

$C_n^k - \sum_{j\text{是}i\text{儿子}} C_{size[j]}^k - C_{n-size[i]}^k$

假设a[i]表示最终答案中C_i^k的系数，对于任何k来说a都是不变的。

然后答案$ans[k] = \sum_{i=k}^{n} a[i] * C_i^k$

$ans[k] = \frac{1}{k!} * \sum_{i=k}^{n} a[i] * i! * \frac{1}{(i-k)!}$

这显然可以卷积。

NTT即可。

连续段

对于一个下标和值都为1 到k 的k 阶排列p，我们称一个区间$[l,r]$ 是连续的当且仅当不存在三个数x,y,z 使得$p_x < p_y < p_z, x \in [l,r], z \in [l,r], y \notin [l,r]$。

我们称两个排列p_1 和p_2 等价当且仅当它们的阶数（长度）相等且连续区间集合相同。

输入N，求对于每个$n = 1, 2, \ldots, N-1$，所有n 阶排列形成的等价类个数对质数P 取模的结果。

N=1e5,P=998244353

这题就是计算析合树的个数。两个排列的析合树相同则连续段的区间集合相同。
由于析合树的子树仍然是析合树，我们可以根据这个列方程

$$F(x) = \frac{F^2(x) + F^4(x)}{1 - F(x)} + x$$

显然我们不会解四次方程。
但是我们知道常数项是0，用牛顿迭代就能直接解出最后的多项式。

逆序对

给定 n, k，请求出长度为 n 的逆序对数恰好为 k 的排列的个数。答案对 $10^9 + 7$ 取模。

对于一个长度为 n 的排列 p，其逆序对数即满足 $i < j$ 且 $p[i] > p[j]$ 的二元组 (i, j) 的数量。

- n,k<=10^5。

模型转换

- 假设有一个i的排列，插入i+1逆序对个数会增加多少?
- 发现会增加0~i。
- 因此模型转化为，n个变量，0<=xi<i
- 问有多少x序列，满足和为k。

推生成函数

把第i个的生成函数写出来。

$\sum_{j=0}^{i-1} x^j = \frac{1-x^i}{1-x}$

定义$F(x) = \prod_{i=1}^{n} \frac{1-x^i}{1-x}$

那么$[x^k]F(x)$就是答案。

爽快多项（理论复杂度优越实际效率差）

$$lnF(x) = \sum_{i=1}^{n} ln(1-x^i) - ln(1-x)$$

我们知道$ln(1-x^i) = -\sum_{j>=1} \frac{x^{ij}}{j}$

因此可以用调和级数即n log n复杂度内求出ln F(x)。

接着用FFT做多项式的exp。

本题模数比较诡异，要用毛爷爷FFT。

DP

$$F(x) = \frac{\Pi_{i=1}^{n}(1-x^i)}{(1-x)^n}$$

好好学过生成函数都知道

$$\frac{1}{1-x^m} = \sum_{i>=0} C_{i+m-1}^{i-1} * x^i$$

因此

$$F(x) = \Pi_{i=1}^{n}(1-x^i) * \sum_{i>=0} C_{i+n-1}^{i-1} * x^i$$

- 后面就是个组合数，因此想办法求出前面的部分。

DP

- 观察它的组合意义： 有n个物品，第i个物品大小为i。 假如选了j个物品，大小和为n，会贡献(-1)^j。
- 就是个带符号的连续背包问题!
- 我们考虑代入经典DP，设f[i,j]表示i个数和为j的贡献和。
- 想象有i个数递增的排列着。
- 第一种情况，全部+1。
- 那么f[i][j]+=f[i-1][j-i]
- 第二种情况，全部+1，再在最前面添加一个1。
- 那么f[i][j]-=f[i-1][j-i]（带符号所以这里是减号）
- 上面这两种情况可以造出任意互不相同的一堆递增数。 但是注意可能最大的那个会超出n，因此要除掉，有
- f[i][j]+=f[i-1][j-(n+1)]（带符号所以这里是加号）
- 于是就行了。

题

- 求出长度为n，不包含长度为m连续子序列是公差为±1的等差数列的排列个数，模998244353
- Level1:n,m<=200
- Level2:n,m<=5000

Level1:
考虑如下容斥
一个排列可以唯一地拆成若干个等差数列拼接，但是如果我们直接dp若干个等差数列拼接的话会算重。因为一个极长等差数列也可以看做若干个等差数列拼接。所以我们要容斥一下。
记容斥系数为g[i]，其生成函数为G(x)，一个拼接的权值为∏g[a[i]]，其中a[i]为各个等差数列的长度。
最终要求的权值记为f[i]，其生成函数为F(x)，由定义得f[i]=[1≤i ≤m]，F(x)=x((1-x^m)/(1-x))
由于这是排列的有序拼接，所以我们可以得到如下的方程F(x)=G(x)/(1-G(x))，用这个式子可以由F解出G
解出G之后可以设计一个简单dp解决这个问题，设f[i][j]表示插了j段，总长为i，转移枚举下一个等差数列的长度，乘上容斥系数，再乘j+1表示插入在某一个空位间。复杂度O(n^3)

Level2:
考虑优化dp，我们发现这个dp的本质是一个多项式复合
ans[i]=∑(m!∏g[a[i]]) （∑a[i]=n,a的长度为m)
即A(x)=∑i!G(x)^i
多项式复合的复杂度目前最优是O((nlogn)^1.5)的。
当然还有一个O(n^2)的简单做法：
预处理p以内和p的倍数次幂的G的点值(p=sqrt(n))，然后就可以O(n^2)求G的0~n次幂的点值，最后dft回去就行了。