棋子分组

猜想上界 $l=\frac{\binom{p^k}{2}}{\binom{p}{2}}$ 一定可以达到，考虑降维构造：

- 当 $k=1$，一组选完所有点；
- 当 $k>1$，分为 p 组每组 p^{k-1} 个点，处理完组与组之间共 $\frac{(p^{k-1})^2\times\binom{p}{2}}{\binom{p}{2}}=(p^{k-1})^2$ 组点（总边数除以每组边数）之后递归处理。一种构造方案是枚举第一组里选的编号 $l\in[0,p^{k-1})$ 以及模意义下的间距 d，选择第 $x=0\sim p-1$ 组里的第 $(l+d\times x) \bmod p^{k-1}$ 个点。考虑 $x\neq y$ 的第 x 组中第 i 个点与第 y 组中第 j 个点，这条边出现时对应组的参数 l,d 满足 $i-j\equiv d\times(x-y)\pmod{p^{k-1}}, i-d\times x\equiv l\pmod{p^{k-1}}$，由于 p 是质数，$0\le x,y<p$，所以 $(x-y)|p^{k-1}$，存在逆元，则 $d=(i-j)\times(x-y)^{-1}$，以及 $l=i-d\times x$，也就是说对于每条组间的边 l,d 是存在且唯一的，于是这样构造是正确的。

复杂度 $O(l\times p)$。

信息传输

树的做法：$f_{i,c}$ 表示 i 的颜色为 c 时 i 子树的方案数之和。进一步地，$f_{i,*}$ 的值均相同，容易优化到 $O(Tn)$。

n 小的做法：枚举同色点集合来划分，并枚举每条边计算贡献。需要枚举 n 个结点的划分方案数 $B(n)$，有 $B(11)=678570$，复杂度 $O(TB(n)m)$。

发现非树边条数 $m-n+1$ 很小，枚举这些返祖边的祖先（关键点）的划分，令 $N=\min(m-n+1,n)$，则复杂度 $O(T(m+N^2B(N)))$，跑不满。

具体做法假设枚举的关键点划分中有 A 个集合，设 $f_{i,c}$ 表示当 i 的颜色属于祖先划分中的第 c 个集合的方案数（$c=0$ 表示和所有祖先的颜色不同），则初始若 i 不是关键点，则 $f_{i,c}=1$，否则设 $f_{i,col_i}=1$。遍历出边，设出边为 $i\to j$，如果边 e 是回到父亲的边则跳过；如果 e 是返祖边，把每个 $f_{i,c}$ 乘上 $([c=col_j]same_e+[c\neq col_j]diff_e)$；如果 e 是树边，则先递归处理 j。

考虑用 $f_{j,b}$ 更新 $f_{i,a}$，设 S 为所有 $f_{j,b\neq0}$ 之和。

首先，对于 $f_{i,a=0}$，有两种情况：

- 若 $b\neq0$，则必然不相等，方案数 $S\times diff_e$；
- 若 $b=0$，有两种情况；
    - i 的真实颜色和 j 相同，j 的真实颜色方案数为 1，方案数 $f_{j,0}\times same_e$；
    - i 的真实颜色和 j 不同，j 的真实颜色方案数为 $k-A-1$，方案数 $f_{j,0}\times(k-A-1)\times diff_e$；

对于其它 a 的转移，有三种情况：

- $a=b$，方案数为 $f_{j,a}\times same_e$；
- $a\neq b,b\neq0$，所有 b 的总方案数为 $(S-f_{j,a})\times diff_e$；
- $a\neq b,b=0$，j 的颜色方案数为 $k-A$，方案数为 $f_{j,0}\times(k-A)$。

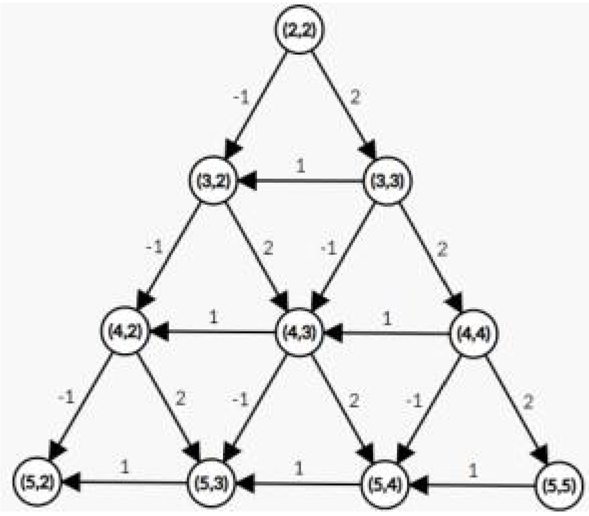
最终的答案为 $\sum_{c=0}^{A}f(rt,c)\times([c\neq0]+[c=0](k-A))$。

双栈排序

手玩/爆搜出 $n=4$ 时不合法的两个排列是 3142 和 4132，猜测不合法当且仅当存在这样相对大小关系的子序列，跑一下 $n=5$ 发现是对的。充分性显而易见，必要性可以将 n 缩小到 4 来证明。

考虑 dp 出合法的方案数，从大到小插入元素，一个元素插入后作为不合法的末端，当且仅当前面两个比它大的元素中间夹了一个比它小的。那么也就是插入到一个地方就保证这个位置前面已经插入的所有元素中间不能再插入元素了。设 $f_{i,j}$ 表示插了 i 个数，有 j 块（块之间可以插元素）的方案数，转移枚举插在哪个空隙，发现插在最开头和前两个块中间会使块数加一，其它的会使 j 块变为 $2 \le k \le j$ 块，转移就是 $f_{i,j} \times 2 \to f_{i+1,j+1}$，$f_{i,j} \to f_{i,k}(2 \le k \le j)$，直接写是 $O(n^3)$ 的。优化成 $O(n^2)$ 也是容易的，可以得到 $f_{i,j} = f_{i,j+1} + 2f_{i-1,j-1} - f_{i-1,j}$。

$n = 1$ 是特殊的，需要特判掉。否则这个 dp 是一个路径权值计数的模型，如下图。



要求的就是 $\sum_{2 \le i \le n} f_{n,i} = f_{n+1,2}$，把两维都减二，转为 $(0,0) \to (n-1,0)$ 并令 $n \leftarrow n-1$。

枚举向右下走了 i 步，贡献为 $(-1)^{n-i}2^i\binom{n+i}{n-i}C_i$，其中 C 是卡特兰数。原因是 C_i 保证任意时刻向左走的次数不超过向右下走的次数，组合数在所有的步数中选择 $n-i$ 个向左下走。

复杂度 $O(Tn)$，因为模数不固定所以时限较大，为了卡可能不存在的多项式算法。

模糊匹配

先提供一种 $O(n\sqrt{n})$ 的做法。枚举 S 中长为 $|T|$ 的连续子串 S'。有一个快速判断方法就是，T 和 S' 的 lcp 长度，加上 T, S' 反串的 lcp 长度大于等于 $|T| - k$。

发现对于一个询问串 T 只有 $|T|$ 种不同的 lcp 长度，于是暴力枚举 lcp 的长度 k_1，这样反串的 lcp 长度 k_2 只需要满足 $k_2 \in [|T| - k - k_1, |T|]$ 即可合法。

换言之，我们枚举 $lcp(T, S') = k_1$，满足条件的串 S'，左端点在 $height$ 数组里是 $\mathcal{O}(1)$ 个区间。同时需要满足 $lcp(rev(S'), rev(T)) \in [|T| - k - k_1, |T|]$ 。（其中 $rev(S)$ 表示 S 的反串。），容易发现这样的串在 $height$ 数组里也是 $\mathcal{O}(1)$ 个区间。对于某一个从 i 开始，长为 $|T|$ 的连续子串 S'，需要满足 i 在正串选取的区间里，$i + |T| - 1$ 在反串选取的区间里。

由于总长是 $u = 2 \times 10^5$，所以不同的 $|T|$ 只会有 $\sqrt{u}$ 种，对于每一种长度 $|T|$，我们都把反串 $height$ 数组重新编号。即使反串 $hight$ 数组内的 i 变成 $i - |T| + 1$。

于是问题就变成了简单的，两个数组，每个数组给出一段区间，求区间交。（区间内的数互不相同）

由于区间数互不相同，可以先差分，变成两个数组前缀求交。我们再把第一个数组离散化，变成 $1 \dots n$，这样就是每次在第二个数组里的区间查询小于等于 x 的个数。这部分时间复杂度离线扫描线 $\mathcal{O}(n \log n)$ 即可。

前面部分时间复杂度是 $\mathcal{O}(n\sqrt{n})$。

然后是一种 polylog 的做法。模糊匹配相当于失配位置可以用长度为 k 的条 cover。我们枚举一个长度为 k 的条，忽略里面的内容，计算匹配方案数。为了去掉多算的，我们还得枚举长度为 $k-1$ 的条，忽略里面的内容，计算匹配方案数。注意这里的条均可以超出边界。

对于每个长度为 k 的条计算忽略这个子串的匹配方案数。我们先假设条不超过边界。

假设被忽略的部分对应原串 a 的 $[x,x+k-1]$ 区间，在小串 b 中对应 $[y,y+k-1]$ 区间。要求的就是 $b[1:y-1]$ 是 $a[1:x-1]$ 的后缀，$b[y+k:m]$ 是 $a[x+k:]$ 的前缀。

对于 $b[y+k:m]$ 是 $a[x+k:]$ 的前缀这个条件，我们把原串和小串用 `$` 隔开一起建后缀数组，相当于要求 $lcp(y+k+o,x+k)\geq m-y-k+1$，相当于 $x+k$ 在 rank 上要求的位置是一段区间，在 height 上二分 rmq 定位出即可。反过来也一起建后缀数组，也对应着反过来 $x-1$ 的 rank 的一个区间。

对于超过边界的部分，对应着的是 $x\geq$ 某个负数和 $x\leq$ 某个超出边界的正数这些限制，对于这些超出边界的 x，我们可以直接令它的 rank 为本身，这样限制同样是 rank 区间。

现在问题就变成了，每个 x 有两个 rank，每次询问要求出满足第一个 rank 在一个区间，第二个 rank 在一个区间的 x 的个数，相当于有一些点，求矩形内的点数。离线扫描线树状数组或可持久化线段树。