necklace 解题报告

福建省福州第一中学 董克凡

1 试题大意

给出一个长度为n的环，一共有k种颜料，你要给环中的每一个位置上色，要求环中相邻两个位置的颜色不同，若两个环旋转之后能够完全重合那么这两个环被认为是相同的。要求所有的颜色都要使用，求能染成的不同的环的数量。结果对10^9+7取模

对于20%的数据$n \le 10, k \le 5$

对于40%的数据$n \le 20, k \le 10$

对于60%的数据$n \le 10000, k \le 10^4$

对于80%的数据$n \le 10^7, k \le 10^4$

对于100%的数据$n \le 10^9, k \le 5 * 10^4$

2 考察内容

初等数论，二项式反演，Burnside引理。

3 算法介绍

3.1 算法一

对于20%的数据，k^n很小，所以可以考虑搜索算法，枚举每一个位置的颜色。由于要求每种颜色都要用上，所以可以强制让第一个位置填颜色0，这是因为总可以找到一种旋转方法将颜色0旋转到第一个位置。如果按照顺序搜索的话，其余的每一个位置能够产生的方案数只有$k-1$种，所以搜索复杂度为$(k-1)^{n-1}$。在搜出每一种合法方案之后，可以将每一个方案映射到一个不超

过k^{n-1}的非负整数上。然后在搜出一个新的方案之后，枚举它的旋转情况，判断之前是否出现了与其等价的方案即可。

时间复杂度：$O\left((k-1)^{n-1}\right)$

期望得分：20分。

3.2 算法二

记长度为n的环，k种颜色全部用上的本质不同的合法方案数为$F(n,k)$，k种颜色不一定全部用上的本质不同的合法方案数为$G(n,k)$，那么将$G(n,k)$中所有的方案按照一共使用了多少颜色分类，就可以得到如下等式：

$$G(n,k)=\sum_{i=0}^{k}\binom{k}{i}F(n,i)$$

其中枚举的i表示实际使用了多少种颜色。对这个式子进行二项式反演，得到：

$$F(n,k)=\sum_{i=0}^{k}\binom{k}{i}G(n,i)(-1)^{k-i}$$

于是接下来我们只需关注$G(n,k)$的求法即可。

由于对项链的所有旋转操作构成了一个置换群，所以由Burnside引理，可以得到如下等式：

$$G(n,k)=\frac{1}{n}\sum_{i=0}^{n-1}T(g,k)$$

其中$g=(n,i)$，为n与i的最大公约数，$T(g,k)$表示一个长度为g的环，要求相邻的两个位置颜色不同，至多使用k种颜色的方案数。通过Burnside引理，去掉了要求旋转不同构的限制。

考虑$T(g,k)$的求法。根据算法一的分析，可以令第一个位置的颜色为1，接下来用$dp[i][j]$，表示已经做完了前i个位置，第i个位置所填颜色为j的方案数。这个动态规划可以$O\left(nk^2\right)$求解。这样可以用过40%的数据。更进一步，我们只关心第i个位置的颜色是否为1即可。所以j只需要记一个01状态。那么这个动态规划的复杂度就优化为了$O(n)$，结合二项式反演以及Burnside引理，总时间复杂度为$O(knlogn)$

期望得分：40~60分。

4 算法三

我们对Burnside引理得出的式子进一步简化：

$$
\begin{aligned}
G(n,k) &= \frac{1}{n}\sum_{i=0}^{n-1} T(g,k) \\
&= \frac{1}{n}\sum_{d|n}\left(T(d,k)\sum_{i=0}^{n-1}[(i,n)=d]\right) \\
&= \frac{1}{n}\sum_{d|n}\left(T(d,k)\sum_{i=0}^{n-1}[(\frac{i}{d},\frac{n}{d})=1]\right) \\
&= \frac{1}{n}\sum_{d|n}\left(T(d,k)\varphi\left(\frac{n}{d}\right)\right)
\end{aligned}
$$

接下来考虑$T(d,k)$的计算。当$n=2$时，$T(d,k)=k(k-1)$。当$n>2$时，如果不考虑第一个元素与最后一个元素不相等的限制，那么方案数就是$k(k-1)^{n-1}$，但是这有可能导致第一个元素与最后一个元素相等。考虑将这种情况扣除，若第一个元素与最后一个元素相等，那么直接删除最后一个元素之后就得到了一个合法的方案，所以

$$T(d,k)=k(k-1)^{n-1}-T(d-1,k)$$

所以可以归纳证明，得到：

$$T(d,k)=\sum_{i=1}^{n-1}k(k-1)^{n-i}(-1)^{i+1}$$

这是一个等比数列求和，公比为$1-k$，所以当$k\neq 1$时

$$T(d,k)=(k-1)^n+(-1)^n(k-1)$$

所以$T(d,k)$可以用$O(logn)$的时间求出。

经过上面两个优化，总时间复杂度为$O\left(k\sqrt{n}logn\right)$

期望得分：60~80分。

5 算法四

注意到，算法三的每一步计算中要求出i^n，这一步的时间复杂度为$klogn$。考虑只对$i\in\mathbb{P}$使用快速幂求i^n，对于其它的i，一定可以将其表示为$i=ab(a,b<i)$，

那么$i^n = a^n b^n$，这样就只需要做$k/logk$次快速幂，所以时间复杂度为$O\left(k\sqrt{n}logn/logk\right)$，期望得分：100 分。

6 得分估计

这应该是一道中规中矩的NOI第一天第三题难度的题目，相信大家应该都能拿到不错的分数。

应该绝大多数选手应该能拿到至少40分，70%的选手应该能拿到至少60分，50%的选手可以拿到满分。

另外，预计会有30%的选手在拿到满分之前经过不少代码实现上面的优化，我也不太确定能否通过卡常数的方法而不是加入最后一点优化的方法通过此题。