

理论

理论部分太长惹。。 晚点填

题目

1、模板:

poj2154 Color

给正 n 边形染 m 种颜色，问有多少种染色方案。

对任意正 n 边形有如下 $2n$ 阶二面体群 $G = \{\rho^0, \dots, \rho^{n-1}, \tau^1, \dots, \tau^n\}$ 然后通过 Burnside定理 : $N(G, \mathcal{C}) = \frac{1}{|G|} \sum_{f \in G} |\mathcal{C}(f)|$ 求解

对于旋转产生的置换 ρ^i 产生的贡献 $|\mathcal{C}(\rho^i)|$ 奇偶都一样，要通过 \gcd 来找循环节个数: $|\mathcal{C}(\rho^i)| = m^{\gcd(n,i)}$

对反射产生的置换要分奇偶讨论
$$\begin{cases} \sum |\mathcal{C}(\tau^i)| = n \times m^{\lfloor n/2 \rfloor + 1} & (n \% 2 == 1) \\ \sum |\mathcal{C}(\tau^i)| = \frac{n}{2} \times m^{\lfloor n/2 \rfloor + 1} + \frac{n}{2} \times m^{\lfloor n/2 \rfloor} & (n \% 2 == 0) \end{cases}$$
 最后的答案即 $\frac{1}{2n} \sum |\mathcal{C}(f)|$

```
while(cin>>m>>n && n+m){
    ans = 0;
    rep(i,0,n-1){
        ans += (ll)pow(m,gcd(n,i)); // 旋转
    }
    if(n&1) ans += n*(ll)pow(m,n/2+1); // 反射
    else ans += (ll)pow(m,n/2)*n/2 + (ll)pow(m,n/2+1)*n/2;
    ans /= 2*n;
    cout<<ans<<endl;
}
```

poj2409 Let it Bead

n 种颜色染正 n 边形，忽略关于旋转带来的重复（即只算旋转的置换）

注意 n 很大($1e9$)

由于去掉反射类型的置换那么答案就是

$$ans = \frac{1}{n} \sum_{i=1}^n n^{\gcd(n,i)}$$
 如果直接用上面的公式显然会 t

考虑如何化简上式，因为 $\gcd(n,i)=k$ 的 i 肯定有很多，所以可以枚举 n 的因数，这样复杂的就降到了 $O(\sqrt{n})$

$$\sum_{i=1}^n n^{\gcd(n,i)} = \sum_{d|n} n^d \sum_{i=1}^n [\gcd(n,i)=d] = \sum_{d|n} n^d \sum_{i=1}^{\lfloor n/d \rfloor} [\gcd(\frac{n}{d}, i)=1] = \sum_{d|n} n^d \phi(\frac{n}{d})$$

其中 $\phi(k)$ 表示 k 的欧拉函数

那么我们可以预处理出前 \sqrt{n} 个质数，并用远小于 $O(\sqrt{n})$ 的复杂度求出每个数的欧拉函数 $\phi(k)$

```
for(i=1;i*i<n;i++){
    if(n%i==0) ans =
    (ans+1LL*qpow(n,i-1)*getPhi(n/i)%M+1LL*qpow(n,n/i-1)*getPhi(i)%M)%M;
}
if(i*i==n) ans = (ans+1LL*qpow(n,i-1)*getPhi(i)%M)%M;
```

2020-2021:teams:wangzai_milk:wzx27:combinatorial_mathematics Magic Bracelet

m 种颜色染正 n 边形，但有限制条件颜色 u 和 v 不能相邻出现

注意 m 很小

对于颜色是否能同时出现的问题，考虑用一个关于颜色的邻接矩阵 G 来表示两个颜色之间的关系： $G[u][v]=G[v][u]=0$ ； \leftarrow ； u 和 v 不能同时出现。接着更泛化地表示每个旋转置换 ρ^i 的对非等价着色数的贡献 $\mathcal{C}(\rho^i)$

首先 ρ^i 会产生 $\gcd(n,i)$ 个长度为 $\frac{n}{\gcd(n,i)}$ 循环节，然后枚举每个循环节的颜色就得到了 $\mathcal{C}(\rho^i)$ 。例如没有任何限制时每个循环节都可以取 m 种颜色，所以 $\mathcal{C}(\rho^i)=m^{\gcd(n,i)}$

对于有上述限制的，可以这么做，构造出邻接矩阵后，有

$$\mathcal{C}(\rho^i)=\sum_i \sum_j G^{\gcd(n,i)-1}[i][j] \times [G[i][j]=1]$$

在离散数学的图论中曾提到 $G^k[i][j]$ ； $:=$ ； i 出发，经过 k 条边到达 j 的路径数

在这里就是 $G^k[i][j]$ ； $:=$ ；第一个循环节的颜色是 i 第 $k+1$ 个循环节的颜色是 j 的染色数

因为最后一个循环节会和第一个循环节再次相邻，所以只对 $G[i][j]=1$ 的 $G^{\gcd(n,i)-1}[i][j]$ 算贡献

$$\mathcal{C}(\rho^i)=\sum_i \sum_j G^{\gcd(n,i)-1}[i][j] \times [G[i][j]=1]=\sum_i G^{\gcd(n,i)}[i][i]$$

即第一个循环节颜色是 i 经过 $\gcd(n,i)+1$ 条合法路径又回到自身的数量

所以只要用矩阵快速幂优化一下就可以了

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:wzx27:combinatorial_mathematics&rev=1590053350

Last update: 2020/05/21 17:29