

题目

一. 考察内容:

数论 素数 LCM

二. 题目分析:

[题目大意]

给出1-n, n个数字, 求出有多少种置换周期。

[写题思路]

首先, 对于n个数字的一种置换, 可以分解成几个几个一组的轮换, 那么该置换周期就是这些轮换的周期的最小公倍数。

考虑转化问题, 我们可以发现周期为1的轮换 (其实就是不换) 对结果不产生任何影响, 我们考虑把这些数, 分为几个几个的轮换, 其中这些轮换的周期是互质的, 由于唯一分解定理, 这些轮换周期的乘积是唯一的, 现在的问题变为, 求有多少种置换, 满足其中包含一些两两互质的轮换, 并且轮换周期之和不超过n (小于n的可以用周期为1的轮换补上)。

考虑用动态规划解决该问题, 设状态 $f[i][j]$ 表示考虑前i个素数的情况下, 轮换周期之和为j的时候的总方案数, 转移: 首先如果不使用第i个素数, 那么可以继承之前的方案数 $f[i][j]=f[i-1][j]$ 。如果使用第i个素数, 则要枚举使用多少个, 若使用k个, 则 $f[i][j]+=f[i-1][j-\text{prime}[i]*k]$, 其中需要注意 $\text{prime}[i]*k$ 不能小于j。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATED
/*****
*创建时间: 2018 08 12
*文件类型: 源代码文件
*题目来源: 洛谷
*当前状态: 已通过
*备忘录: 数论 LCM
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
using namespace std;
#define MAXN 1001
int n, prime[MAXN], _prime;
long long ans, f[MAXN][MAXN];
/* Variable explain:

*/
void read()
{
    // freopen(".in", "r", stdin);
    // freopen(".out", "w", stdout);
    scanf("%d", &n);
    return;
}
void pri()
{
```

```
bool vis[MAXN]={false};
for(int i=2;i<=n;++i)
{
    if(!vis[i])prime[++_prime]=i;
    for(int j=1;j<=_prime;++j)
    {
        int k=i*prime[j];
        if(k>n)break;
        vis[k]=1;
        if(i%prime[j]==0)break;
    }
}
}
void dp()
{
    //设状态f[i][j]为前i个素数，总和为j的方案数。
    f[0][0]=1;
    for(int i=1;i<=_prime;++i)//枚举素数
    {
        for(int j=0;j<=n;++j)//枚举容量
        {
            f[i][j]=f[i-1][j];
            for(int k=prime[i];k<=j;k+=prime[i])//枚举第i个素数要选多少个
            {
                f[i][j]+=f[i-1][j-k];
            }
        }
    }
}
int main()
{
    read();
    pri();
    dp();
    for(int i=0;i<=n;++i)ans+=f[_prime][i];
    printf("%lld\n",ans);
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)