

多项式练习 1

习题一

洛谷p3702

题意

询问有多少序列满足下列条件：

1. 序列长度等于 n
2. 序列由不超过 m 的正整数构成且至少含有一个素数
3. 序列所有数之和是 p 的倍数

题解

考虑容斥，于是答案为由不超过 m 的正整数构成的序列数减去不超过 m 的合数构成的序列数。

不超过 m 且模 p 为 i 的正整数个数，而利用线性筛可以计算出不超过 m 且模 p 为 i 的合数个数。

设 $\text{dp}(i,j)$ 表示长度为 i 且和模 p 为 j 的序列数，发现可以利用倍增 NTT 转移。时间复杂度 $O(m+p\log n\log p)$

本题 $p \leq 100$ 且模数为合数，所以考虑直接暴力转移，时间复杂度 $O(m+p^2\log n)$

```
const int MAXM=2e7+5,MAXP=105,Mod=20170408;
int prime[MAXM],p_cnt,p;
bool vis[MAXM];
void get_prime(int n){
    vis[1]=true;
    _rep(i,2,n){
        if(!vis[i])prime[p_cnt++]=i;
        for(int j=0;i*prime[j]<=n&& j<p_cnt;j++){
            vis[i*prime[j]]=true;
            if(i%prime[j]==0)break;
        }
    }
}
void mul(int *a,int *b){
    static int temp[MAXP];
    mem(temp,0);
    _for(i,0,p)_for(j,0,p)
        temp[(i+j)%p]=(temp[(i+j)%p]+1LL*a[i]*b[j])%Mod;
    _for(i,0,p)a[i]=temp[i];
}
```

```
int cnt1[MAXP],cnt2[MAXP],dp1[MAXP],dp2[MAXP];
int main()
{
    int n=read_int(),m=read_int();
    p=read_int();
    get_prime(m);
    _rep(i,1,m){
        cnt1[i%p]++;
        cnt2[i%p]+=vis[i];
    }
    dp1[0]=dp2[0]=1;
    for(int i=30;i>=0;i--){
        if(n&(1<<i)){
            mul(dp1,cnt1);
            mul(dp2,cnt2);
        }
        if(i){
            mul(dp1,dp1);
            mul(dp2,dp2);
        }
    }
    enter((dp1[0]-dp2[0]+Mod)%Mod);
    return 0;
}
```

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

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E5%A4%9A%E9%A1%B9%E5%BC%8F%E5%BA%94%E7%94%A8&rev=1602398997

Last update: 2020/10/11 14:49