Lab & Ex 8　FFT

_Written by_11710823 __ 冰鱼__

Lab 8

题意：给定长度为N的序列A1,A2,...,AN。 给定质数P。求满足

Ai×Aj≡S(mod P) 的有序对 (i, j) 个数, i < j

> **思路：**题目中提示我们回忆三元组的例题
>
> 相互比较可以得到本题与三元组的题目不同之处在于题目所给条件为乘法同余式，并非系数加法求和式，因此难以套用FFT解决
>
> 考虑通过变换将题目所给乘法同余式，转化为系数加法
>
> 由于题目所给P为质数，同时由同余的性质可以想到 利用 **原根**
>
> ### Question 1
>
> 1）为什么想到使用原根？什么是原根？
>
> > 一个数m如果有原根，则其原根个数为phi(phi(m))。特别地，对素数有phi(p)=p-1。
> >
> > 假设g是奇素数p的一个原根，则g^1,g^2,...,g^(p-1)在模p意义下两两不同，且结果恰好为1~p-1
>
> 可以见到，如果我们有了原根g，就可以建立**Ai %p的余数和原根指数之间的一一对应关系**
>
> 而有了这个关系，我们就可以拿着 Ai ， Aj， S mod P 的余数去寻找他们对应的原根指数p, q, r
>
> 进而可以将 Ai % P 改写为 g^p % P
>
> Aj % P 改写为 g^q % P
>
> S % P 改写为 g^r % P
>
> 原条件可以化为 g^p * g^q ≡ g^r (mod P) , 即 g ^ (p + q) ≡ g ^ r (mod P)
>
> 再由原根的指数周期性得 (p + q) % (P - 1) = r

如此，题目就完全归约到之前的二元组问题了。接下来我们只需要对于整个序列，统计其中不同原根指数出现的次数，将其作为对应指数的系数，即可形成一个**P-1次多项式。**

而问题也转化为求 **该多项式卷积自己的结果中，指数为S的项的系数**

Question 2

2) 完成了题目的模型化简，那么如何求原根呢?

> **模 m 有原根的充要条件：** $m=2,4,p^{a},2p^{a}$ ，其中 p 是奇素数。
>
> **求模素数 p 原根的方法：** 对 $p-1$ 素因子分解，即 $p-1=p_1^{a_1}p_2^{a_2}\cdots p_k^{a_k}$ 是 $p-1$ 的标准分解式，若恒有
>
> $$g^{\frac{p-1}{p_i}}\neq 1(\mathrm{mod}\ p)$$
>
> 成立，则 g 就是 p 的原根。（对于合数求原根，只需把 $p-1$ 换成 $\varphi(p)$ 即可）

模板

```
'8. 原根
//前置模板：素数筛，快速幂，分解质因数
//要求 p 为质数
//快速幂
ll bin(ll x, ll n, ll mod) {
    ll ret = mod != 1;
    for (x %= mod; n; n >>= 1, x = x * x % mod)
        if (n & 1) ret = ret * x % mod;
    return ret;
}

//分解质因数
ll factor[30], f_sz;
void get_factor(ll x) {
    f_sz = 0;
    ll t = sqrt(x + 0.5);
    for (ll i = 0; prime[i] <= t; ++i)
        if (x % prime[i] == 0) {
            factor[f_sz++] = prime[i];
            while (x % prime[i] == 0) x /= prime[i];
        }
    if (x > 1) factor[f_sz++] = x;
}

//寻找原根
ll find_smallest_primitive_root(ll p) {
    get_factor(p - 1);
    for(int i=2; i<p; i++) {
        bool flag = true;

        for (int j=0; j<f_sz; j++)
```

```
            if (bin(i, (p - 1) / factor[j], p) == 1) {
                flag = false;
                break;
            }
        if (flag) return i;
    }
    assert(0); return -1;
}
```

EX8

首先，由于字符集合{A, B, C, D, E}**大小只有5**，因此我们可以**对每个字符c分别考虑。**

Step 1.

根据题意，对于每个字符，我们预处理a[i]数组，代表i位置是否能匹配当前字符（左右k个能匹配也算）

Step 2.

创建b[i]数组，保存文本串T[i]位置的字符是不是当前所枚举的字符c。

Step 3.

将b数组倒转，使用FFT卷积，将对字符处理得到的贡献加到母串S的相应位置

Step 4.

在对所有的A,B,C,D,E五个字符完成以上操作后，遍历母串所有位置i，检查该位置上之前所统计的贡献总和是否为文本串T长度，若相等则代表当前位置成配。统计成配位置数量即可