乘法逆元

Modular Multiplicative Inverse

概念

称使得

$$ax \equiv 1 \pmod p$$

成立的 x 为 a 在模 p 意义下的逆元。

快速幂求解

根据费马小定理，**当 p 是质数且 a 不是 p 的倍数时**，有：

$$a^{p-1} \equiv 1 \pmod p$$

于是乎：

$$a \cdot a^{p-2} \equiv 1 \pmod p$$

即 a^{p-2} 是 a 在模 p 意义下的逆元。用快速幂求解。

注意：只适用于 p 是质数且 a 不是 p 的倍数的情形。

扩展欧几里得算法求解

$$ax \equiv 1 \pmod p \iff ax + py = 1$$

由贝祖定理，当 $(a,p)=1$ 时，上式有解。用扩展欧几里得算法解出 x 即可。

注意：p 可以不是质数，但是 a 和 p 必须互质。

线性递推

Theorem：$i^{-1} \equiv -\left\lfloor\frac{p}{i}\right\rfloor \times (p \bmod i)^{-1} \pmod p$.

Proof：设 $p = k \times i + r$，即 $k = \left\lfloor\frac{p}{i}\right\rfloor, r = p \bmod i$，在模 p 意义下该式为 $k \times i + r \equiv 0 \pmod p$，两边同时乘以 $i^{-1}r^{-1}$ 得：$k \times r^{-1} + i^{-1} \equiv 0 \pmod p$，故 $i^{-1} \equiv -k \times r^{-1} \equiv -\left\lfloor\frac{p}{i}\right\rfloor \times (p \bmod i)^{-1} \pmod p$. 证毕。

Code：

```
int main(){
    scanf("%lld%lld", &n, &p);
    inv[1] = 1;
    printf("%lld\n", inv[1]);
    for(int i = 2; i <= n; i++){
        inv[i] = -(p / i) * inv[p % i];
        ((inv[i] %= p) += p) %= p;
        printf("%lld\n", inv[i]);
    }
    return 0;
}
```