Codeforces 919E (2100)

Mmukul Khedekar

27 January 2026

https://codeforces.com/problemset/problem/919/E
Accepted: https://codeforces.com/contest/919/submission/360194638

§1 Solution

§1.1 Explanation

Lemma 1.1
For $k \not\equiv 0 \pmod p$, the map

$$x \mapsto kx \pmod p$$

is a bijection.

If we iterate on remainders of n modulo $(p-1)$, let's say

$$n \equiv i \pmod{p-1}$$

then we have,

$$na^n \equiv b \pmod p \implies n \equiv ba^{-i} \equiv j \pmod p$$

So for each i, we have the system of congruence equations

$$\begin{aligned} n &\equiv j \pmod p \\ n &\equiv i \pmod{p-1} \end{aligned}$$

where $j \equiv ba^{-i} \pmod p$. Using the chinese remainder theorem, we get

$$\begin{aligned} n &\equiv ip\Big(p^{-1} \bmod (p-1)\Big) + j(p-1)\Big((p-1)^{-1} \bmod p\Big) \pmod{p(p-1)} \\ &\equiv i\,p - j\,(p-1) \pmod{p(p-1)} \equiv y \pmod{p(p-1)} \end{aligned}$$

To count the number of solutions in the range $[1, x]$, we just need to compute

$$\left\lfloor \frac{x}{p(p-1)} \right\rfloor + [y \le x \bmod (p(p-1))]$$

Thus the algorithm terminates in $\mathcal{O}(p)$ time.

§1.2 Code

```
ll binpow(ll x, ll y, ll m) {
    x %= m;
    ll result = 1;

    while (y > 0) {
        if (y & 1) result = result * x % m;
        x = x * x % m;
        y >>= 1;
    }

    result %= m;
    return result;
}

ll modinv(ll x, ll p) {
    return binpow(x, p - 2, p);
}

void solve() {
    ll a, b, p, x;
    std::cin >> a >> b >> p >> x;

    ll ans = 0;
    ll val = 1;
    for (ll i = 0; i < p - 1; i++, val = (val * a) % p) {
        ll j = (b * modinv(val, p)) % p;
        ll n = -j * (p - 1) + i * p;
        if (n < 0) {
            n += p * (p - 1);
        }

        ans += (x / (p * (p - 1)));
        if (n <= (x % (p * (p - 1)))) {
            ans++;
        }
    }

    std::cout << ans << '\n';
}
```