Định lý Lucas

Định lý Lucas

Định lý ¶

Nếu M là số nguyên tố thì $C_N^K \equiv C_{n_0}^{k_0}.C_{n_1}^{k_1}...C_{n_p}^{k_p} \ (mod\ M)$

Trong đó:

$\overline{n_p n_{p-1} ... n_0}$ là dạng biểu diễn của N trên hệ cơ số M

$\overline{k_p k_{p-1} ... k_0}$ là dạng biểu diễn của K trên hệ cơ số M

Nói cách khác:

$N = n_0.M^0 + n_1.M^1 + ... + n_p.M^p$

$K = k_0.M^0 + k_1.M^1 + ... + k_p.M^p$

Chứng minh

Với M là số nguyên tố và i là số nguyên với $1 \leq i < M$

$\Rightarrow C_M^i = \frac{M.(M-1)...(M-i+1)}{i.(i-1)...1} \equiv 0 \ (mod\ M)$ do $gcd(M, i!) = 1$

$\Rightarrow (1+X)^M = \sum_{i=0}^{M} C_M^i.X^i \equiv 1 + X^M \ (mod\ M)$ với mọi $X \in \mathbb{Z}$

Lại có:

$(1+X^M)^M \equiv ((1+X)^M)^M \equiv (1+X)^{M^2} \ (mod\ M)$

và

$(1+X^M)^M \equiv 1 + (X^M)^M \equiv 1 + X^{M^2} \ (mod\ M)$

$\Rightarrow (1+X)^{M^2} \equiv 1 + X^{M^2} \ (mod\ M)$

Cứ tiếp tục như vậy, với mọi $i \in N$ ta được:

$(1+X)^{M^i} \equiv 1 + X^{M^i} \ (mod\ M)$

Ta có:

$\sum_{K=0}^{N} C_N^K . X^K$

$= (1 + X)^N$ (nhị thức Newton) (1)

Tách N về dạng cơ số M ta được:

$(1) = (1 + X)^{n_0.M^0 + n_1.M^1 + \ldots + n_p.M^p}$

$= \prod_{i=0}^{p} ((1 + X)^{M^i})^{n_i}$

$\equiv \prod_{i=0}^{p} (1 + X^{M^i})^{n_i} \ (mod\ M)$

$= \prod_{i=0}^{p} \sum_{k_i=0}^{n_i} C_{n_i}^{k_i} X^{k_i.M^i}$ (nhị thức Newton)

$= \prod_{i=0}^{p} \sum_{k_i=0}^{M-1} C_{n_i}^{k_i} X^{k_i.M^i}$ ($n_i \le M - 1$ với mọi i và $C_j^i = 0$ với $i > j$) (2)

Nhóm các $C_{n_i}^{k_i} X^{k_i.M^i}$ lại ta có

$C_{n_0}^{k_0} . C_{n_1}^{k_1} \ldots C_{n_p}^{k_p} . X^{k_0.M^0 + k_1.M^1 + \ldots k_p.M^p}$

Do đó với một bộ $(k_0, k_1, \ldots k_p)$ bất kì ta được một hạng tử

$C_{n_0}^{k_0} . C_{n_1}^{k_1} \ldots C_{n_p}^{k_p} . X^K$

($C_{n_0}^{k_0} . C_{n_1}^{k_1} \ldots C_{n_p}^{k_p}$ là hệ số của X^K)

Vậy $(2) = \sum_{K=0}^{N} \prod_{i=0}^{p} C_{n_i}^{k_i} X^K$

Từ đó suy ra: $\sum_{K=0}^{N} C_N^K . X^K \equiv \sum_{K=0}^{N} \prod_{i=0}^{p} C_{n_i}^{k_i} X^K \ (mod\ M)$ với mọi $X \in \mathbb{Z}$

$\Leftrightarrow C_N^K \equiv \prod_{i=0}^{p} C_{n_i}^{k_i} \ (mod\ M)$

Cài đặt

Biểu diễn một số N ở dạng cơ số M

```
vector<int> getRepresentation(int N) {
    vector<int> res;
    while (N > 0) {
        res.push_back(N % M);
        N /= M;
    }
    return res;
}
```

Đoạn code trên chạy trong thời gian $O(log_M N)$

Tính $C_{n_i}^{k_i}$

Thuật toán $< O(n^2), O(1) >$

Với N nhỏ ta có thể sử dụng công thức tam giác Pascal để tính trước trong $O(n^2)$ và truy vấn trong $O(1)$:

```
int C[M][M];
for (int i = 0; i < M; ++i) {
    for (int j = 0; j <= i; ++j) {
        if (i == 0 || j == 0) {
            C[i][j] = 1;
        } else {
            C[i][j] = (C[i - 1][j - 1] + C[i - 1][j]) % M;
        }
    }
}
```

Thuật toán $< O(M), O(logM) >$

Với M nhỏ các bạn có thể tiền xử lý trong $O(M)$ và truy vấn trong $O(logM)$ bằng trick #3 ở đây .

Tiền xử lý

```
long long fact[M];
fact[0] = 1;
for (int i = 1; i < M; ++i) {
    fact[i] = (fact[i - 1] * i) % M;
}
```

Truy vấn

```
int C(int N, int K) {
    if (K > N) {
        return 0;
    }
    return (((fact[N] * binpow(fact[N - K], M - 2)) % M) * binp
}
```

Trong đó hàm `binpow(a,n)` dùng để tính nhanh a^n trong $O(logn)$ bằng chia để trị:

$a^n = (a^{n/2})^2$ nếu n chẵn

$a^n = (a^{n/2})^2 * a$ nếu n lẻ

Có thể cài đặt bằng đệ quy theo công thức trên hoặc cài khử đệ quy như sau:

```
int binpow(int a, int n) {
    long long res = 1;
    while (n > 0) {
        if (n % 2 != 0) {
            res = (res * a) % M;
        }
        a = ((long long)a * a) % M;
        n /= 2;
    }
    return (int)res;
}
```

Tính C_N^K

```
vector<int> n = getRepresentation(N);
vector<int> k = getRepresentation(K);
long long res = 1;
for (int i = 0; i < k.size(); ++i) {
    res = (res * C(n[i], k[i])) % M;
}
```

Trường hợp M không là số nguyên tố

Chúng ta thực hiện các bước như sau:

- Phân tích thừa số nguyên tố $M = {m_1}^{a_1}.{m_2}^{a_2}...{m_r}^{a_r}$
- Tính $C_N^K \ mod \ m_1, C_N^K \ mod \ m_2, ... C_N^K \ mod \ m_r$
- Sử dụng Định lý Thặng dư Trung Hoa để khôi phục $C_N^K \ mod \ M$

Luyện tập

- Xông đất ngày tết - SPOJ

Được cung cấp bởi Wiki.js