-Cách tính a\*b mod m (|a\*b| > 2^63-1)

code:

long long mul(long long a, long long b, long long m)

{

long long ans = 0;

while (a)

{

if (a % 2 == 1) ans = (ans+b) % base;

a/= 2;

b = b\*2 % base;

}

return ans;

}

////////////////////

- Cách để phân tích thừa số nguyên tố nhanh (n <= 10^6)

Sàng nguyên tố, code:

Void sang()

{

For(i,2,n)

if (dd[i] == 0)

{

dd[i] = i;

long long j = 1LL\*i\*i;

while (j <= n)

{

dd[j] = i;

j+= i;

}

}

}

(dd[i] là 1 số nguyên tố bất kì mà i chia hết)

phân tích một số ra thừa số nguyên tố, code:

void ptich(int a)

{

while (a > 1)

{

int p = dd[a];

s[p]++;

a/= p;

}

} //dpt: O(log2(a)), s[i] là số mũ của thừa số nguyên tố i

/////////////////////////

-Tính tổ hợp chập k của n (C(k,n) % m) (n, k <= 10^6; m<= 10^9)

+ C(k,n) = n!/(k!\*(n-k)!)

+ Cách làm:

Để tính n!: For i = 1->n phân tích i ra thừa số nguyên tố (i = a1^p1\*a2^p2...\*ax^px)

Rồi Tu[a[i]]+= p[i] (Tu[i] là số mũ của thừa số nguyên tố i trong n!);

Để tính k!: For i = 1->k phân tích i ra thừa số nguyên tố (i = a1^p1\*a2^p2...\*ax^px)

Rồi Mau[a[i]]+= p[i] (Mau[i] là số mũ của thừa số nguyên tố i trong k!), tương tự với (n-k)!;

Cuối cùng để tính C(k, n):

{

kq = 1;

for(int i = 1; i<=n; i++) kq = kq\*pow(i, tu[i]-mau[i]) % m;

}

+ dpt: O(nlog(n)) nếu sử dụng thuật toán pitch đã nêu ở trên

////////////////////

-Có 1 số dãy số có quy luật, ví dụ:

+ f[i] = (r\*f[i-1] + f[i-2]) % base;

Với base nhỏ, ta có thể tìm chu kì của dãy số để tính f[i] (i<=1e18) (như ví dụ trên). Hoặc nhân ma trận với dãy có chu kì lớn hoặc base lớn