9/7/19

Teorema fundamental de la aritmetica:

Todo numero natural admite una unica descomposicion como producto de primos

Determinar si un numero es primo- algoritmo basico:

m no es primo sii m = a.b (a,b != 1)

Funciona bien hasta 10^14

Esto tiene complejidad O(raiz(m).logm)

es mejor cuando consultan por un unico primo

Nota: funcion puts para imprimir

En linux existe la funcion factor en la terminal para factorizar por primos

criba→ O(1)

anda mejor para muchos primos

fill\_criba() → O(MAXN)\*O(log (log MAXN))

MAXN<= 10⁷

\_\_gcd maximo comun divisor

b >= a

gdc(a,b) == gcd(a, b-a) == gcd(b%a, a)

TEOREMA:

a\*b = gcd(a,b) \* mcm(a,b)

Cantidad de divisores de un numero:

multiplicacion de los exponentes de los factores primos del numero

Si a,b,m son enteros con m>=1

a congruente b (mod m)

significa que a-b es multiplo de m

Obs.

a congruente b (m) y c conguente d(m) => a+c congruente b+d (m)

a congruente b (m) y c congruente d(m) => a\*c=b\*d (m)

congruente establece una relacion de equivalencia

Obs.

6\*4 congruente 1 (mod 23)

Digo que el inverso del 6 mod 23 es 4

Teorema: Pequeno teorema de fermat (fermatito)

si m es primo

a^m-1 congruente 1 (mod m) para todo 1<=a<=m-1

Obs

a\*a^m-2 congruente 1 (mod m)

Si quiero por ejemplo el inverso de 7 mod 23 basta calcular

7^21

int ans=1

forn(i,27)

ans=(ans\*7)%23

Teorema chino del resto

Si conozco x modulo m y conozco x modulo n entonces tengo x modulo m\*n

n= 23^2019 + 88^42072

n congruente (mod 2)

23 congruente 3 mod 5

23^2019 congruente 3^2019 mod(5) =

3³ mod(5) = 27 mod(5) = 2 mod(5)

n congruente 3 modulo(5)

Encontrar ultima cifra del periodo de 1/19

realizar la suma de 1+18+18^2+18^3+...++18^10 (mod 10^6+3)