Teoría de Números

I wanna be the very best

Ivo Pajor

Univesidad de Buenos Aires

Training Camp 2024

Gracias Sponsors!

Organizador

Diamond

Gold

Resumen

El Cosechero

Lema conocido

$$\sum_{i=1}^{n} \frac{n}{i} \approx n \times \log n$$

El Cosechero

Lema conocido

$$\sum_{i=1}^{n} \frac{n}{i} \approx n \times \log n$$

¿De qué nos sirve esto?

- Para justificar la complejidad de algunos algoritmos.
- Una forma de pensar algoritmos.

Problema

Arpa and a list of numbers

Arpa has found a list containing n numbers. He calls a list bad if and only if it is not empty and the gcd of the numbers in the list is 1.
Arpa can perform two types of operations:

- Choose a number and delete it with cost x.
- Choose a number and increase it by 1 with cost y.

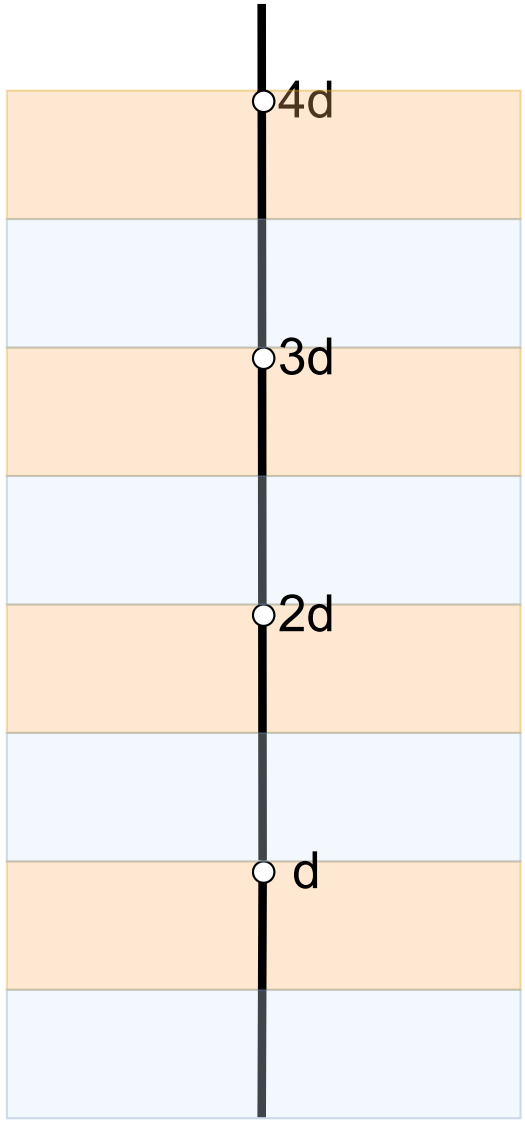
Arpa can apply these operations to as many numbers as he wishes, and he is allowed to apply the second operation arbitrarily many times on the same number.
Help Arpa to find the minimum possible cost to make the list good.

- $n \leq 5 \times 10^5$.
- $a_i \leq 10^6$.

Ideas

- Como el **gcd** tiene que ser distinto de 1, algún numero $d > 1$ debe dividir a todos los números.
- Se puede ver que solo nos interesan los $d <= 10^6$.
- La idea central es poder chequear cuál es el mínimo costo para hacer que todos los números sean divisibles por un d arbitrario.
- Si podemos iterar por todos los d tenemos la respuesta.

Idea



Observación clave

Sea una sucesión a_i de enteros. Entonces la sucesión:

$$g_i = \begin{cases} a_0 & \text{si } i = 0 \\ gcd(g_{i-1}, a_i) & \text{caso contrario} \end{cases}$$

tiene a lo sumo $\log a_0$ números distintos y es decreciente. No solo eso, los g_i son divisores de a_0.

Amor Salvaje

Observación clave

Sea una sucesión a_i de enteros. Entonces la sucesión:

$$g_i = \begin{cases} a_0 & \text{si } i = 0 \\ gcd(g_{i-1}, a_i) & \text{caso contrario} \end{cases}$$

tiene a lo sumo $\log a_0$ números distintos y es decreciente. No solo eso, los g_i son divisores de a_0.

¿De qué nos sirve esto?

- En muchos de estos casos podemos utilizar programación dinámica.

Problema

Array GCD

Tenemos un array a_i de longitud n. Se pueden aplicar dos operaciones a este array:

1. Eliminar algún subsegmento (subsecuencia continua) de longitud $m < n$ y pagar por ello $m \cdot a$ monedas.
2. Cambiar algunos elementos del array en no más de 1, y pagar b monedas por cada cambio.

Cada una de las operaciones puede aplicarse como máximo una vez (y puede que no se aplique en absoluto), por lo que solo se puede eliminar un segmento y cada número puede cambiarse (aumentarse o disminuirse) en no más de 1. Además, no se permite eliminar todo el array.
Nuestro objetivo es calcular el número mínimo de monedas que necesitamos gastar para hacer que el máximo común divisor de los elementos del array resultante sea mayor que 1.

Ideas

- Intentemos enchufar la observación en este problema.
- Antes de eso, notemos que al menos el primer o el último elemento están presenten en el arreglo final.
- Concentrémonos en el primer elemento, por la observación anterior (un poco tirado de los pelos) sabemos que si no lo borramos, el **gcd** final debe ser un divisor de este número.
- Pero además, todo los **gcd** parciales son divisores.
- Entonces podemos ir "construyendo" el arreglo de forma parcial, guardando el **gcd** de este arreglo parcial (que es un divisor de a_0!). Esto nos da naturalmente una dp.
- Esto, sin embargo, tiene una complejidad de $O(n \times d(a_0))$, que es medio malo.

Otra observación clave

Muchas veces, los primos los únicos que importan son.

Continuando

- Podemos hacer una dp para ver si conseguimos que todos los números de nuestro arreglo final sean divisibles por algún primo que divide a a_0.
- De esta forma reducimos en mucho la complejidad. Nos queda $O(n \times loga_0)$.

Funciones Multiplicativas

Decimos que una función aritmética f es multiplicativa si:

$$f(a \times b) = f(a) \times f(b) \;\; \forall a, b \in \mathbb{N} \text{ tal que } gcd(a, b) = 1$$

Convolución de Dirichlet

Sean f, g dos funciones aritméticas definimos $f * g$, la convolución de f y g, como:

$$f * g = \sum_{d|n} f(d) g(\frac{n}{d})$$

Funciones Multiplicativas

- Función constante 1.
- $Id_a(p^k) = p^{ak}$
- $\epsilon(p^k) = [p^k == 1]$
- Función de Möbius.
- ϕ de Euler.

La propiedad

Si f y g son funciones multiplicativas entonces $f * g$ es multiplicativa.

La propiedad

Si f y g son funciones multiplicativas entonces $f * g$ es multiplicativa.

Pa qué sirve todo esto?:

- Una de las razones es que para calcular los primeros valores de una función multiplicativa, se puede utilizar una idea similar a criba de Eratóstenes.
- Muchas veces se puede probar que la función que queremos calcular es multiplicativa, y ahí es todo lindo.

Problema

Bash Plays with Functions

Definimos una función $f_0(n)$, que denota el número de formas de factorizar n en dos factores p y q tales que $\gcd(p, q) = 1$. En otras palabras, $f_0(n)$ es el número de pares ordenados de enteros positivos (p, q) tales que $p \cdot q = n$ y $\gcd(p, q) = 1$. Ahora definamos una serie de funciones, donde f_{r+1} se define como:

$$f_{r+1}(n) = \sum_{u \cdot v = n} \frac{f_r(u) + f_r(v)}{2}$$

Queremos saber el valor de $f_r(n)$ para diferentes r y n. Dado que el valor podría ser enorme, hacerlo módulo nuestro primo favorito.

Ideas

- ¿Cuánto vale $f_0(p^k)$? ¿ Es multiplicativa?

Ideas

- ¿Cuánto vale $f_0(p^k)$? ¿ Es multiplicativa?
- Veamos con cariño la definición de f_r.

Ideas

- ¿Cuánto vale $f_0(p^k)$? ¿ Es multiplicativa?
- Veamos con cariño la definición de f_r.
- Como f_r es multiplicativa, podes usar la idea de la criba, pero cómo calculamos $f_r(p^k)$?

Ideas

- ¿Cuánto vale $f_0(p^k)$? ¿ Es multiplicativa?
- Veamos con cariño la definición de f_r.
- Como f_r es multiplicativa, podes usar la idea de la criba, pero cómo calculamos $f_r(p^k)$?
- Notemos que para cada primo es igual.

Ideas

- ¿Cuánto vale $f_0(p^k)$? ¿ Es multiplicativa?
- Veamos con cariño la definición de f_r.
- Como f_r es multiplicativa, podes usar la idea de la criba, pero cómo calculamos $f_r(p^k)$?
- Notemos que para cada primo es igual.
- Luego, podemos hacer una dp ya que el exponente de cada primo está acotado por 20.

(Esta definición no es la correcta, pero nos va a servir)

Raíces Primitivas

Sea p un primo, llamamos a g raíz primitiva de p si para todo entero a con $gcd(a, p) = 1$ ocurre que existe k tal que:

$$g^k = a \mod p$$

Dato importante

Para todo p existe una raíz primitiva, además se puede probar que el orden de alguna raíz primitiva es: $O(\log(p)^6)$. (Asumiendo hipótesis de Riemann)

¿Cómo calculamos una raíz primitiva?

Ejemplo

Sea $p = 5$, una raíz primitiva es 2 ya que:

1. $1 = 2^4 \mod 5$
2. $2 = 2^1 \mod 5$
3. $3 = 2^3 \mod 5$
4. $4 = 2^2 \mod 5$

Ejemplo

Sea $p = 5$, una raíz primitiva es 2 ya que:

1. $1 = 2^4 \mod 5$
2. $2 = 2^1 \mod 5$
3. $3 = 2^3 \mod 5$
4. $4 = 2^2 \mod 5$

Usos:

- Nos permite trabajar productos de números modulo p como suma modulo $p - 1$. ¡Cuidado con el 0!

Problema

Product Modulo (Atcoder)

Tomemos un número primo $P = 200003$. Se te dan N enteros $A_1, A_2, \ldots, A_N$. Encuentre la suma de $((A_i \cdot A_j) \mod P)$ sobre todos los $\frac{N \cdot (N-1)}{2}$ pares no ordenados de elementos $(i < j)$.
La suma no se calcula módulo P.

Ideas

Si sobra tiempo lo cuento en el pizarrón.