

JUGANDO CON DIVISORES

AUTOR: LAUTARO LASORSA

Solución Esperada (Nivel 3)

En la solución esperada lo primero que hay que hacer es obtener los diferentes divisores primos del número **N**, dado que cada operación consiste básicamente en dividir al actual número **x** por uno de sus factores primos.

Esto se puede hacer considerando todos los números desde 2 hasta la **raíz cuadrada de N**, en orden creciente, y cada vez que pasamos por un número, si este divide al N actual nos lo guardamos (ya que es uno de sus factores primos) y dividimos al **N** por este número hasta que ya no sea posible. Notar que como modificamos el N en el proceso, es necesario hacer una copia para no perder el valor original. Al final de este proceso, si N vale 1 encontramos todos los factores primos, y sino el valor de N nos indicará cuál es el primo que falta. Notar que la cantidad de factores primos distintos de **N** es menor a $\log_2(N)$.

Luego, lo que hay que hacer es una dp minmax sobre los valores, ya que durante el juego solo vamos a pasar por los divisores de **N** (cuya cantidad para **N** hasta 10^{14} está acotado por 17280). En cada número nos guardaremos el valor que habría si Gastón o Agustín empiezan jugando en ese número, como los números pueden ir hasta **N**, utilizaremos un `unordered_map` para cada jugador. Para calcular cada valor (si no fue calculado anteriormente) lo que debe hacer es ver a todos los posibles números que podría ir desde ese, y considerar lo que resultaría de que su adversario empiece jugando con dicho número. Luego, de todas las opciones, toma la que más le convenga según su objetivo (la mínima/máxima) y le agrega el puntaje asociado al número en el que está actualmente.

El caso base es cuando vale 1, que solo se toma el valor correspondiente a 1. En la solución oficial se carga este valor en los `unordered_map` antes de llamar a las funciones recursivas.

Notar que para los niveles 1 y 2, esta misma solución anda correctamente (**solo que en esos casos $p_x=x$**).

La complejidad de esta solución es $N^{1/2}$ por buscar los divisores primos de **N**, más $\log(N)$ por cada divisor de **N**, que son a lo mucho 17280. (Y, en general, la máxima cantidad de divisores para un número de X dígitos está acotada por **x^{exp} para cualquier $\text{exp}>0$**). Es decir, siendo D la cantidad de divisores de N, la complejidad de la solución es $O(N^{1/2}+D*\log(N)) = O(N^{1/2})$, dado que el término de la raíz cuadrada de N es ampliamente mayor al resto.

Notar que el definir $p_x=((x^2)\%M)^2$ tiene por única función evitar que haya un goloso sencillo como en las versiones de nivel 1 y 2 del problema, y no tiene

JUGANDO CON DIVISORES

AUTOR: LAUTARO LASORSA

ninguna propiedad particular que se aproveche en la solución. Lo único a tener en cuenta es que los participantes deben tener cuidado con el orden de las operaciones para evitar un overflow.

Soluciones Parciales (Nivel 3)

En la primer subtarea, $N \leq 10$, se espera que un fuerza bruta que realice el minmax pero sin memorizar (es decir, sin hacer dp) entre en tiempo, a su vez tampoco se espera que el participante pre calcule los divisores de N (o siquiera note que solo se va a pasar por esos números durante el juego).

En la subtarea $N \leq 1.000$, se espera por ejemplo que tampoco observe que solo va a pasar por los divisores de N , pero si utilice memorización en la programación dinámica. Además, puede o no pre calcular qué números son primos. En función de esto la complejidad oscilara entre $O(N^2)$ y $O(N^2 \cdot \log(N))$

En la subtarea $N \leq 1.000.000$ se espera que sepa que solo debe considerar los divisores de N , pero que no los obtenga en tiempo $O(N^{1/2})$ sino en tiempo $O(N)$, por lo demás se espera una dp similar a la solución completa o a la de la siguiente subtarea.

Para la subtarea $N \leq 10^{10}$ se espera una programación dinámica similar a la de la solución completa, solo que no note que en cada caso solo debe considerar por distintos divisores primos de N para saber los posibles números a los que puede ir, y en lugar de eso evalúe todos los divisores de N para saber si dividen al número en el que está actualmente y además tenga precalculados cuales son primos (en un `unordered_set` por ejemplo). Así, la complejidad de la solución es $O(N^{1/2} + D^2)$ o $O(N^{1/2} + D^2 \cdot \log(D))$, en ambos casos con una constante importante. Como para $N \leq 10^{10}$ D está acotado por 2304 estas soluciones deberían entrar perfectamente en tiempo.

Solución Esperada (Nivel 2)

En la solución general se espera que el participante obtenga todos los factores primos (incluyendo repetidos) del N en $O(N^{1/2})$, y se de cuenta de una observación clave. A Agustín siempre le convendrá dividir por el mayor factor primo posible, y a Gastón por el menor factor primo posible. Por tanto, la solución pasa por ordenar los factores primos de N , y en cada paso si juega Gastón saca el menor

JUGANDO CON DIVISORES

AUTOR: LAUTARO LASORSA

factor primo (es decir, divido al **x** actual por ese factor primo y ese es mi nuevo **x**) y si juega Agustín saco el mayor factor primo. Esto se puede hacer con un set o con un array y 2 punteros (uno al inicio y otro al fin), una deque, etc.

Soluciones Parciales (Nivel 1 y 2) y Solución Esperada (Nivel 1)

Para la subtarea donde **N ≤ 10** y **N ≤ 1.000** la solución esperada es la misma que en el caso de Nivel 3.

Para la subtarea **N ≤ 1.000.000**, que es a su vez el problema completo en Nivel 1, lo que se espera es la misma solución que en el problema completo pero obteniendo los factores primos en **O(N)** en vez de **O(N^{1/2})**.

Para la subtarea donde **N** es el producto de 2 factores primos, se pretende premiar al participante por obtener los divisores de N en **O(N^{1/2})**, dado que en este caso hay solo 3 o 4 divisores y es un caso que el participante puede plantear y resolver a mano.