

Complejidad de `BigInteger.h`:

- La inicialización de un objeto `BigInteger` tiene una complejidad de tiempo de $O(n)$, ya que se requiere asignar memoria para almacenar los dígitos del número entero.
- Las operaciones de asignación, copia y destrucción de `BigInteger` también tienen una complejidad $O(n)$, ya que implican la manipulación de los dígitos almacenados.

Complejidad de `BigInteger.cpp`:

- Las operaciones aritméticas básicas, como la suma, resta, multiplicación y división, tienen una complejidad de $O(n^2)$, donde n es el número de dígitos en el número entero más grande involucrado en la operación.
- La multiplicación y división por un solo dígito tienen una complejidad de tiempo de $O(n)$, ya que implican una multiplicación o división de cada dígito del número por el dígito dado.
- Las operaciones de comparación y igualdad tienen una complejidad de $O(n)$, ya que requieren comparar cada dígito del número entero.
- La conversión de `BigInteger` a una representación de cadena (string) tiene una complejidad $O(n)$, ya que implica la concatenación de los dígitos en orden.

Complejidad de `main.cpp`:

- El bucle principal que procesa los casos de prueba tiene una complejidad de $O(t * n^2)$, donde t es el número de casos de prueba y n es el número de dígitos en el número entero más grande en cada caso.
- Cada iteración del bucle realiza una comparación de dos `BigInteger`, lo cual tiene una complejidad de $O(n)$.
- La lectura de los casos de prueba desde la entrada estándar y la escritura de los resultados en la salida estándar tienen una complejidad de $O(1)$ por caso de prueba.

.