

Clase Decimal

Análisis Temporal:

Función/Operador	Complejidad Temporal	Explicación
decimal(long valor)	$O(\log n)$	n es el valor numérico del argumento valor. La operación de división utilizada para extraer los dígitos tiene una complejidad logarítmica.
operator+(const decimal &op2) const	$O(1)$	Realiza una iteración fija de 30 iteraciones, independientemente del tamaño de los operandos.
operator<<(ostream &salida, const decimal &num)	$O(1)$ en el mejor caso $O(30)$ en el peor caso	En el mejor caso, cuando el valor de num es cero. En el peor caso, realiza una iteración fija de hasta 30 iteraciones para imprimir los dígitos no nulos.

Análisis Espacial:

Componente	Consumo de Memoria
Clase decimal	60 bytes + tamaño de ostream (suponiendo short de 2 bytes y ostream como un objeto externo)
operator+(const decimal &op2) const	60 bytes adicionales (para almacenar el resultado temporal)
operator<<(ostream &salida, const decimal &num)	No consume memoria adicional significativa

Clase listadecimal

Análisis Temporal:

Función/Operador	Complejidad Temporal	Explicación
ListaDecimal() (constructor predeterminado)	$O(1)$	Inicializa los punteros head y tail.
ListaDecimal(const ListaDecimal& copia) (constructor de copia)	$O(n)$	n es el número de nodos en la lista original. Recorre la lista original una vez.
~ListaDecimal() (destructor)	$O(n)$	n es el número de nodos en la lista. Recorre la lista una vez para liberar la memoria.
pedirDatos()	$O(n * m)$	n es el número de datos ingresados, m es el número de dígitos de cada dato. Recorre cada dato una vez.
entrarDatos(long v)	$O(1)$	Inserción en una lista doblemente enlazada es en tiempo constante.
quitarDatos()	$O(n * m)$	n es el número de datos eliminados, m es el número de dígitos de cada dato. Recorre la lista para encontrar y eliminar cada dato.
validar(long v)	$O(n)$	n es el número de nodos en la lista. Recorre la lista una vez para verificar si el dato existe.
borrarDatos(long v)	$O(n)$	n es el número de nodos en la lista. En el peor caso, recorre toda la lista para encontrar el dato a eliminar.

imprimir1() e imprimir2()	$O(n)$	n es el número de nodos en la lista. Recorre la lista una vez para imprimir los datos.
operator+(const ListaDecimal& op2) const	$O(n)$	n es el número de dígitos del número más largo entre los operandos. Recorre ambas listas una vez para realizar la suma.
operator<<(ostream& salida, const ListaDecimal& num)	$O(n)$	n es el número de dígitos del número. Recorre la lista una vez para imprimir los dígitos.

Análisis Espacial:

Componente	Consumo de Memoria
Clase ListaDecimal	$O(n)$, donde n es el número de dígitos del número. Cada nodo en la lista doblemente enlazada consume memoria adicional.
operator+(const ListaDecimal& op2) const	$O(n)$, donde n es el número de dígitos del número más largo entre los operandos. Se crea una nueva lista doblemente enlazada para almacenar el resultado, cuya longitud no puede ser mayor que la del operando más largo.