INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Carrera: Licenciatura en Ciencia de Datos

Unidad de aprendizaje: Análisis y diseño de algoritmos

Práctica 4 Divide y vencerás
Multiplicación de enteros largos
Algoritmo de Karatsuba

Equipo:
CodeClimbers

Alumnos:
González Miranda Luis Antonio
Melgar García Genaro de Jesús
Merlos Hernández Aarón

Grupo:
3AM1

Profesora: Luz María Sánchez García

Práctica 4
Algoritmo de Karatsuba

Objetivo de la práctica
Desarrollar un programa con el algoritmo de Karatsuba para la multiplicación de enteros largos mediante Divide y vencerás.

Introducción

La característica clave del algoritmo Karatsuba es que divide los números a multiplicar en dígitos más pequeños, reduciendo así el tamaño de los operandos. Luego, realiza una serie de operaciones recursivas para calcular los productos parciales necesarios y, finalmente, combina estos productos para obtener el resultado final. El algoritmo aprovecha la propiedad algebraica de la multiplicación para reducir la cantidad de operaciones requeridas, lo que resulta en una mejora significativa en el tiempo de ejecución.

La estrategia "Divide y vencerás" se aplica al algoritmo Karatsuba dividiendo los números grandes a multiplicar en dígitos más pequeños, lo que reduce la complejidad del problema original. Luego, se aplican operaciones recursivas en los subproblemas generados para obtener los productos parciales. Estos productos parciales se combinan utilizando sumas y desplazamientos adecuados para obtener el resultado final de la multiplicación.

En resumen, el algoritmo Karatsuba aprovecha la estrategia de diseño "Divide y vencerás" para lograr una multiplicación eficiente de números grandes. Al dividir el problema en subproblemas más pequeños y aplicar operaciones recursivas, se reduce la cantidad de operaciones necesarias y se mejora el rendimiento del algoritmo. Esta combinación de técnicas hace del algoritmo Karatsuba una herramienta poderosa en el campo de la computación numérica.

Metodología

La metodología del algoritmo Karatsuba se basa en un enfoque recursivo y en la división de los números a multiplicar en dígitos más pequeños. A continuación, describiré paso a paso cómo funciona:

1) Dados dos números grandes que queremos multiplicar, los dividimos en dígitos más pequeños. Por ejemplo, si tenemos los números A y B de n dígitos cada uno, los dividimos en dos partes aproximadamente iguales:
   A = A1 * 10^(n/2) + A0
   B = B1 * 10^(n/2) + B0

   Donde A1 y B1 representan las mitades más significativas de A y B, respectivamente, y A0 y B0 representan las mitades menos significativas.

2) Calculamos tres productos parciales utilizando las siguientes fórmulas:

$P0 = A0 * B0$
$P1 = A1 * B1$
$P2 = (A0 + A1) * (B0 + B1)$

3) Utilizando la recursión, multiplicamos los productos parciales P0 y P1 para obtener dos productos intermedios:
   $R0 = P0$
   $R1 = P1$

   Nota: Si el tamaño de los números es lo suficientemente pequeño, se puede utilizar un algoritmo de multiplicación directa en lugar de la recursión.

4) Restamos los productos intermedios de P2 y agregamos los resultados a R1:
   $R1 = R1 + P2 - P0 - P1$

5) Finalmente, combinamos los productos intermedios multiplicados por potencias de 10 apropiadas y los sumamos para obtener el producto final:
   Resultado = R0 + R1 * 10^(n/2) + P1 * 10^n

Este proceso se repite de forma recursiva hasta llegar al caso base definido en el paso 1, momento en el cual se retorna directamente el producto de los dígitos individuales. Al combinar los productos parciales obtenidos en cada nivel de recursión, se logra calcular eficientemente el resultado final de la multiplicación de los números grandes.

Complejidad temporal

Ecuación de recurrencia, caso base y notación O para el algoritmo clásico de multiplicación de matrices:

Ecuación de recurrencia: T(n) = T(n-1) + O(n)
Caso base: Cuando n = 1, se realiza una única multiplicación de dos elementos de las matrices.
Notación O: O(n^3)

Explicación: En el algoritmo clásico, cada multiplicación de elementos requiere O(1) operaciones y hay un total de n^2 multiplicaciones necesarias para obtener el resultado final. Por lo tanto, la ecuación de recurrencia se define como T(n) = T(n-1) + O(n), donde T(n-1) representa el tiempo necesario para multiplicar las matrices de tamaño (n-1) x (n-1). El caso base se alcanza cuando se llega a matrices de tamaño 1x1, donde se realiza una única multiplicación de elementos. La notación O del algoritmo clásico de multiplicación de matrices es O(n^3), ya que se realizan n^2 multiplicaciones y cada multiplicación toma O(1) tiempo.

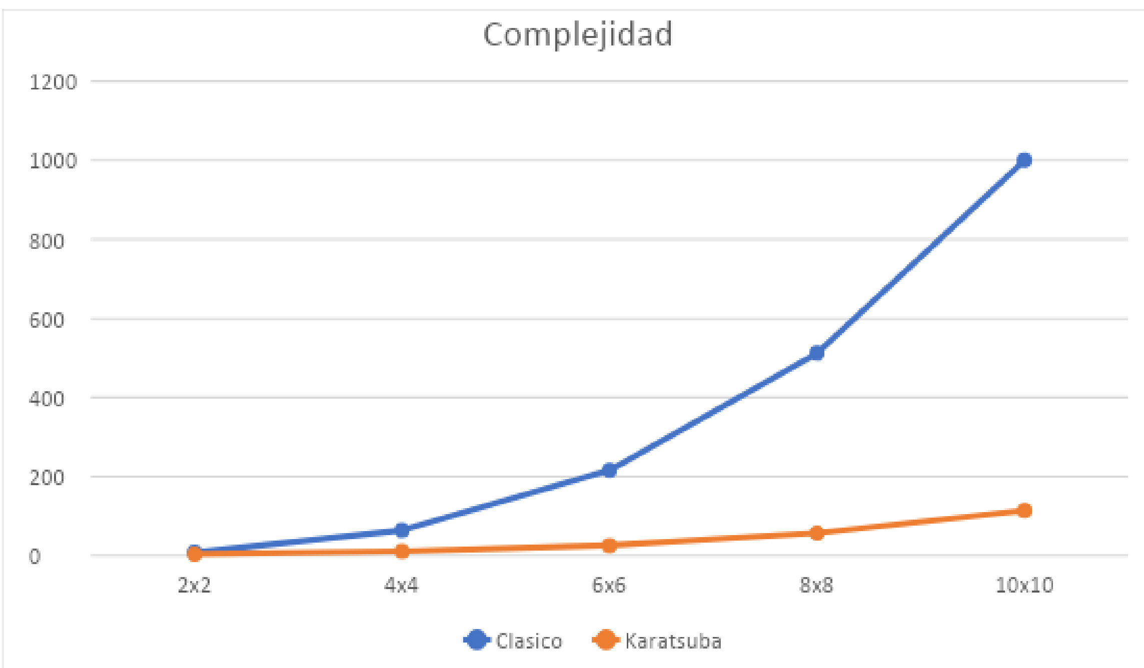
Si consideramos dos números de longitud n, el algoritmo de Karatsuba tiene una complejidad temporal aproximada de O(n^log2(3)), lo cual se puede simplificar a O(n^1.585) utilizando las propiedades de la exponenciación.

Resultados

La práctica se desarrolló a partir de los programas publicados en el sitio de GeeksforGeeks https://www.geeksforgeeks.org/divide-and-conquer-algorithm-introduction/

La siguiente tabla muestra los resultados de las complejidades temporales de los algoritmos para la multiplicación de enteros por el método tradicional y con el algoritmo recursivo con Divide y vencerás.

nxn	Algoritmo clásico	Algoritmo de Karatsuba
2x2	8	4
4x4	64	11
6x6	216	26
8x8	512	57
10x10	1000	114



En el algoritmo clásico, la multiplicación de dos números de n dígitos requiere n^2 operaciones de multiplicación y luego realizar sumas y acarreos para obtener el resultado final. Por lo tanto, la complejidad temporal del algoritmo clásico es O(n^2).

Al dividir los números en partes más pequeñas, el algoritmo de Karatsuba reduce la cantidad de multiplicaciones necesarias. En lugar de realizar n^2 multiplicaciones, el algoritmo realiza solo 3 multiplicaciones parciales de números con la mitad de dígitos y algunas operaciones adicionales de suma y resta.

Conclusiones

En esta práctica hemos explorado el algoritmo de Karatsuba, una técnica eficiente para multiplicar números grandes utilizando la técnica de divide y vencerás. Hemos aprendido que este algoritmo reduce la cantidad de multiplicaciones necesarias al descomponer los números en partes más pequeñas y realizar multiplicaciones parciales, para luego combinar los resultados de manera recursiva.

Comparado con el algoritmo clásico de multiplicación, el algoritmo de Karatsuba ofrece una complejidad temporal menor, lo que significa que es más eficiente para multiplicar números de gran tamaño. Al reducir la cantidad de multiplicaciones requeridas, el algoritmo de Karatsuba logra una mejora significativa en el tiempo de ejecución.

Además, hemos visto que la implementación del algoritmo de Karatsuba es relativamente sencilla y se basa en la recursión para dividir y vencer el problema. Sin embargo, es importante tener en cuenta que el algoritmo de Karatsuba tiene un umbral a partir del cual el algoritmo clásico puede ser más eficiente, debido a las operaciones adicionales requeridas por el algoritmo de Karatsuba.

Anexos

Los códigos fuente se anexan en un archivo comprimido.