Algoritmo de Karatsuba

Wilfredo Gallegos Paz

Universidad del Valle de Guatemala
Teoría de números

26 de noviembre de 2023

Tabla de contenido

Anatoli Karatsuba

Anatoli Alekséyevich Karatsuba nació el 31 de enero de 1937 en Rusia, y murió el 20 de septiembre de 2008 a los 71 años de edad. Estudió en la Universidad Estatal de Moscú el grado de Matemáticas y obtuvo su doctorado en matemáticas con su tesis "Suma de razones trigonométricas de una forma especial y sus aplicaciones". Su más grande aporte, y por la razón que es conocido, es por su método para multiplicar grandes números bastante rápido. Sus investigaciones se centraron principalmente en sumas trigonometricas, integrales trigonométricas, la función Zeta de Riemann, caracteres de Dirichlet, autómatas finitos y algoritmos eficientes.

Figura: Anatoli karatsuba

Algoritmo de Karatsuba

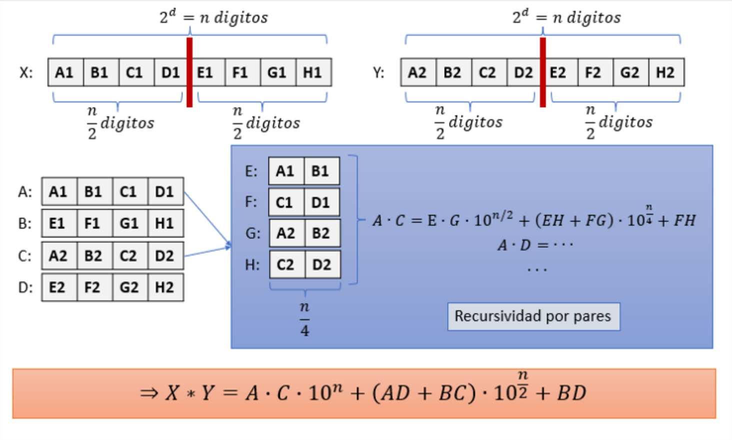
Qué es este algoritmo?
El algoritmo de Karatsuba, también conocido como, 'Algoritmo divide y vencerás', es un procedimiento que sirve para multiplicar números grandes eficientemente. La idea simplificada del algoritmo es descomponer los números a multiplicar en componentes de la mitad del tamaño, y operar cada una de estos componentes por separado para luego volver a juntarlos.
Pros y contras del algoritmo
Como se explicó, el algoritmo divide los números a pares, por lo que se necesita dos números de la misma longitud y con 2^n dígitos para que mantenga su eficiencia. La eficiencia de la multiplicación normal es de $\mathcal{O}(n^2)$ mientras que la complejidad de este algoritmo es de $\mathcal{O}(n^{log_2(3)}) \approx \mathcal{O}(n^{1,58`})$. (Esto se explicara en las siguientes diapositivas)

¿Cómo funciona el algoritmo de Karatsuba?

Algoritmo de Karatsuba

Representación visual del algoritmo

Algoritmo Visual



Algoritmo de Karatsuba

Pseudocódigo del algoritmo

```
def Karatsuba(a,b,n):
    if len(a)<=2:
        return a*b
    mitad=floor(n/2)
    A=a[n . . . n/2]
    B=a[n/2-1 . . . 1]
    C=b[n . . . n/2]
    D=b[n/2-1 . . . 1]
    E=Karatsuba(A,B,mitad)
    F=Karatsuba(A,D,mitad)
    G=Karatsuba(B,C,mitad)
    H=Karatsuba(B,D,mitad)
    XxY=E*10^n+(F+G)^mitad+H
    return XxY
```

Complejidad

Previo a argumentar la complejidad del algoritmo de Karatsuba es necesario tener el conocimiento del siguiente teorema:

Teorema maestro(Master Theorem)

Dada la recurrencia de la forma:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n),$$

para constantes $a \geq 1$ y $b > 1$ con f asintóticamente positiva, las siguientes afirmaciones son verdaderas:

Caso 1. Si $f(n) = O(n^{\log_b a - \epsilon})$ para algún $\epsilon > 0$, entonces $T(n) = \Theta(n^{\log_b a})$.

Caso 2. Si $f(n) = \Theta(n^{\log_b a})$, entonces $T(n) = \Theta(n^{\log_b a} \log n)$.

Caso 3. Si $f(n) = \Omega(n^{\log_b a + \epsilon})$ para algún $\epsilon > 0$ (y $a \cdot f\left(\frac{n}{b}\right) \leq cf(n)$ para algún $c < 1$ y todo n suficientemente largo), entonces $T(n) = \Theta(f(n))$.

Complejidad

La recurrencia del algoritmo de Karatsuba es:

$$T(n) = 3T\left(\frac{n}{2}\right) + c \cdot O(n)$$

Donde:
- $T(n)$ es el tiempo de ejecución del algoritmo para multiplicar dos números de n dígitos.
- $3T\left(\frac{n}{2}\right)$ representa las tres multiplicaciones recursivas de números más pequeños.
- $c \cdot O(n)$ es el tiempo necesario para combinar los resultados intermedios(c es la cantidad de resultados lineales).

Usando el teorema maestro, que establece $T(n) = aT\left(\frac{n}{b}\right) + f(n)$, donde $a = 3$, $b = 2$, y $f(n) = O(n)$, y si $f(n) = O(n^c)$ para $c < \log_b a$, entonces la solución es $T(n) = \Theta(n^{\log_b a})$.

En este caso, $a = 3$, $b = 2$, y $c = 1$ (ya que $O(n) = O(n^1)$). Como $c < \log_2 3$, según el teorema maestro, la complejidad del algoritmo de Karatsuba es $\Theta(n^{\log_2 3})$.

Dígitos de distinta longitud

Pseudocódigo modificado

```
def Potenciar(a,b):
    maximo=max(len(a),len(b))#longitud_maxima(a,b)
    diferencia=0
    while (#digitos not 2^n):
        n+=1#aumentar potencia de 2
        if(maximo == 2^n):
            diferencia=2**n-maximo
            a,b,n = volver_par(a,b,diferencia)
            #exit while
        elif(2**n>maximo):
            diferencia=2**n-maximo
            a,b,n = volver_par(a,b,diferencia)
            #exit while
        else:
            #continue
    return a,b
```

Ejemplos

Archivo .py

Referencias

Rudini Sampaio
Construção e Análise de Algoritmos - UFC - Algoritmo de Karatsuba (aula 1).
Video de YouTube
https://www.youtube.com/watch?app=desktopv=FRtigtlTILM

László Babai
Divide and Conquer: The Karatsuba algorithm.
Pdf
https://people.cs.uchicago.edu/ laci/HANDOUTS/karatsuba.pdf

Algoritmo de Karatsuba
Algoritmo de Karatsuba
https://es.wikipedia.org/wiki/Algoritmo$_de_Karatsuba$
Wikipedia

Anatoli Karatsuba
Anatoli Karatsuba
https://es.wikipedia.org/wiki/Anatoli$_Karatsuba$
Wikipedia

¿Preguntas?
Muchas gracias
gal20399@uvg.edu.gt