.UBAfiuba
FACULTAD DE INGENIERÍA

División y conquista: Multiplicación - Karatsuba

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

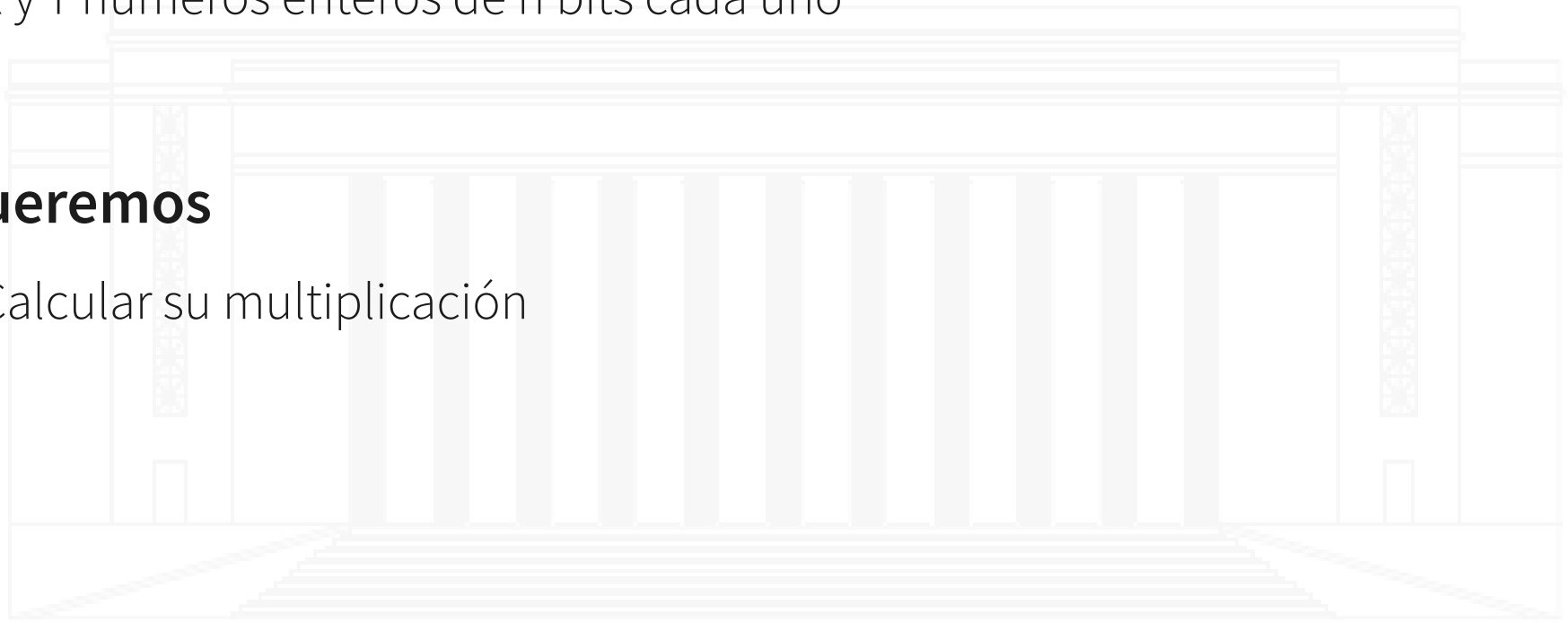
Multiplicación de números enteros

Sean

X y Y números enteros de n bits cada uno

Queremos

Calcular su multiplicación



Un problema histórico

Hace milenios la humanidad construyó algoritmos para multiplicar

- Método egipcio
- Método babilonio
- Método japones (o chino)
- Método hindú (o de celdillas)

Método “tradicional”

Sean X y Y de n bits (dígitos)

Por cada bit (dígito) lo multiplicamos por cada uno de los bits (dígitos) del otro numero

Encolumnamos los resultados y sumamos

Realizamos

Orden de n^2 multiplicaciones básicas

Mas un conjunto de sumas elementales

```
    4572
  x 3169
 -------
   41148
+  27432
    4572
  13716
 -------
 14488668
```

Complejidad $O(n^2)$

Conjetura n^2

Andrey Nikolaevich Kolmogorov

Sostuvo que no era posible una complejidad mejor

(No era el único)

Conjetura: Afirmación que se supone cierta

pero que no ha podido ser ni probada ni refutada.

“Si existiese mejor método, ya tendría que haber sido encontrado”

El piso se rompe

Anatoly Karatsuba

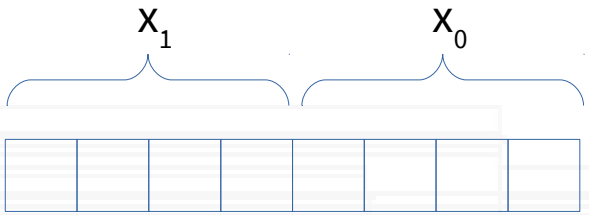
- estudiante de 23 años en 1960
- seminario de problemas matemáticos en cibernética,
- Facultad de mecánica y matemáticas de la universidad de Moscú
- Le bastó una semana para refutar la conjetura n^2
- Paper "Multiplication of Many-Digital Numbers by Automatic Computers"

Idea

Representamos

$$X = x_1 * 2^{n/2} + x_0$$

$$Y = y_1 * 2^{n/2} + y_0$$



Operamos

$$X*Y = (x_1 * 2^{n/2} + x_0) * (y_1 * 2^{n/2} + y_0)$$

$$X*Y = x_1 y_1 * 2^n + (x_0 y_1 + x_1 y_0) * 2^{n/2} + x_0 y_0$$

En vez de 1 multiplicación de n bits tenemos 4 multiplicaciones de n/2 bits

Idea (cont)

Realizamos recursivamente la multiplicación

Hasta la multiplicación de 1 a 1 bits (o dígito)

Relación de recurrencia:

$T(n) = 4\,T(n/2) + cn$

Por teorema maestro $O(n^2)$

… por el momento no mejore nada

El aporte de Karatsuba

Partimos de

$$X*Y = x_1 y_1 * 2^n + (x_0 y_1 + x_1 y_0) * 2^{n/2} + x_0 y_0$$

Podemos ver que

$$(x_0 + x_1) * (y_0 + y_1) = x_1 y_1 + (x_0 y_1 + x_1 y_0) + x_0 y_0$$

Ya calculado

Lo que busco

$$(x_0 y_1 + x_1 y_0) = (x_0 + x_1) * (y_0 + y_1) - x_1 y_1 - x_0 y_0$$

Multiplicación de n/2 bits

Pseudocodigo

Calculamos:

$X*Y =$

$x_1 y_1 * 2^{n/2} +$

$[(x_0 + x_1) * (y_0 + y_1) - x_1 y_1 - x_0 y_0] * 2^{n/2} +$

$x_0 y_0$

Son 3 multiplicaciones de n/2 bits

Mas sumas elementales

Cada recursión trabaja con n/2 bits

```
Karatsuba(x,y)
     x = x1 * 2^(n/2) + x0
     y = y1 * 2^(n/2) + y0

     Calcular x1 + x0
     Calcular y1 + y0

     P = Karatsuba (x1 + x0 , y1 + y0)

     x1y1 = Karatsuba (x1 , y1 )
     x0y0 = Karatsuba (x0 , y0 )

     Retornar x1y1 2^n + (P - x1y1 - x0y0)2^(n/2) +
               x0y0
```

Complejidad

Relación de recurrencia

$T(n) = 3T(n/2) + cn$

$T(1) = a$

Por teorema maestro

$O(n^{\log_2 3}) = O(n^{1,59})$

Que es más “eficiente” que el algoritmo tradicional

Este método tiene mas sumas básicas, por lo que la ventaja se ve para n grandes

Desenrrollando la recursión

Relación de recurrencia

$T(n) = 3T(n/2) + cn$

$T(1) = a$

$T(n/2) = 3T(n/4) + n/2$

$T(n/4) = 3T(n/8) + n/4$

...

$T(n) = 3 [3 (3 \{ T(n/16) + n/8 \} + n/4) + n/2] + n$

$$T(n) = 3^k * 1 + \sum_{i=0}^{k} \frac{3^i * n}{2^i}$$

Continuemos

vemos que k= logn

$$T(n) = 3^k * 1 + \sum_{i=0}^{k} \frac{3^i * n}{2^i}$$

$$T(n) = 3^{\log(n)} + n * \sum_{i=0}^{\log(n)} \left(\frac{3}{2}\right)^i$$

$$T(n) = 3^{\log(n)} + \frac{n * \left(1 - (3/2)^{\log(n)+1}\right)}{(1 - 3/2)}$$

$$T(n) = 3^{\log(n)} + 2n * \left((3/2)^{\log(n)+1} - 1\right)$$

$$T(n) = 3^{\log(n)} + 3n * (3/2)^{\log(n)} - 2n$$

$$\sum_{i=0}^{x} r^i = \frac{1 - r^{(x+1)}}{1 - r}$$

Y continuamos…

$$a^{\log_2 n} = n^{\log_2 a}$$

$$T(n) = 3^{\log(n)} + 3n\frac{3^{\log(n)}}{2^{\log(n)}} - 2n$$

$$T(n) = 3^{\log(n)} + 3 * 3^{\log(n)} - 2n$$

$$T(n) = 4 * 3^{\log(n)} - 2n$$

$$T(n) = 4 * n^{\log 3} - 2n$$

Si n→∞

$O(n^{\log_2 3}) = O(n^{1,59})$

Evolución de la multiplicación

Se puede hacer mejor?

Nueva conjetura de Shönhage-Strassen (1971) → O(nlogn)

Avances en los métodos de multiplicación.

Año	$M(n)$	autor
-2000?	$O(n^2)$	desconocido
1960	$O(n^{\log_2 3})$	Karatsuba
1966	$O(n \log n \, e^{\sqrt{2 \log_2 n}})$	Toom-Cook
1971	$O(n \log n \log \log n)$	Schönhage-Strassen
2007	$O(n \log n \, K^{\log^* n})$	Fürer
2018	$O(n \log n \, 4^{\log^* n})$	Harvey-van der Hoeven

Se ha llegado al máximo?

David Harvey y Joris van der Hoeven presentan en 2019

Paper: “Integer multiplication in time O(n log n)”

https://hal.archives-ouvertes.fr/hal-02070778/document

Se apoyan en la FFT pero la llevan a 1729 dimensiones

FFT: Fast Fourier transform

Técnica utilizada por primera vez por Schönhage y Strassen

.UBAfiuba

FACULTAD DE INGENIERÍA

Presentación realizada en Abril de 2020