

Tarea Programada Karatsuba

Estudiantes:

Giancarlo Alvarado Sánchez

Andrés Álvarez Durán

Hilary Vanessa Granados Álvarez

Grupo: 03

Horario: 08:00 am

**Índice**

1. Portada
2. Índice
3. Introducción e Investigación
4. Metodología
5. Ejemplo simple
6. Nuestro algoritmo en Python
7. Conclusión
8. Bibliografía

**Introducción**

En el presente trabajo los estudiantes Giancarlo Alvarado Sánchez, Andrés Álvarez Durán y Hilary Vanessa Granados Álvarez, mostraremos información pertinente sobre el algoritmo creado en los años sesenta por el matemático ruso Anatoli Alekséyevich Karatsuba, algoritmo conocido bajo nombre del apellido del autor y que muestra una gran eficiencia para realizar multiplicaciones de grandes números en poco tiempo. En este trabajo profundizamos en diversos temas que son de interés para entender y utilizar Karatsuba, su método de implementación, su tiempo de corrida, verificaciones propias y distintos ejemplos.

**Investigación**

Para la realización de este reporte escrito utilizamos distintas fuentes para comprender datos referentes del inicio de la creación de este algoritmo. Entre los datos que nos resultaron dignos de mención se encuentran datos como que dicho algoritmo es posee la singularidad de ser un algoritmo con derivación inmediata al de Toom-Cook o que es el primero creado de división y conquista que se llegó a usar para multiplicar.m Todo esto nos sirvió para darnos una idea de donde se encontraban las bases para la creación del algoritmo y hacia dónde se dirigía; ya que la necesidad matemática o informática para mejorar e intentar dar mucha más eficiencia a los procesos en relación a las multiplicaciones era indispensable para el avance tecnológico de la época y que este algoritmo dio un nuevo aire a los métodos utilizados en ese entonces e incentivó muchísimo el uso de la multiplicación en nuevos algoritmo.

El algoritmo de Karatsuba es una ejemplificación de cómo usar el método de “divide y conquista” o “divide y vencerás”, método algorítmico con el que se intenta solucionar un problema complicado realizandole disminuciones al mismo cuantas veces sean necesarias para así ir simplificando el mismo hasta hacerlo muy sencillo. Y es justamente la manera en la que Anatoli Karatsuba consigue refutar una afirmación de años anteriores en la que se afirmaba que para realizar multiplicaciones de manera cibernética siempre iba a tener (sin importar los números) un tiempo de corrida O(n^2), ya que con este método se logra un tiempo de corrida de O(log \_2(3)).

**Metodología**

El método consiste en realizar los siguientes pasos:

1. Primero se divide el número en dos mitades

• x = x\_H r^n/2 + x\_L

• y = y\_H r^n/2 + y\_L

1. Luego

• xy = (x\_H r^n/2 + x\_L) y\_H r^n/2 + y\_L = x\_H y\_H r^n+ (x\_H y\_L + x\_Ly\_H) r^n/2 + x\_L y\_L

1. Así pasamos a tener tres subproblemas

• a = x\_H y\_H

• b = x\_L y\_L

• c = (x\_H + x\_L) (y\_H y\_L) - a - d

1. Y para finalizar

• xy = a r^n + e r^n/2 + d

**Ejemplo simple 1234 \* 4321:**

• a\_1 = 12 \* 43

• b\_1 = 34 \* 21

• c\_1 = (12 + 34) \* (43 + 21) - a\_1 - d\_1 = 46 \* 64 - a\_1 - d\_1

**Se usa recursividad para el primer subproblema:**

• a\_1 = 12 \* 43

Subproblemas:

• a\_2 = 1 \* 4 = 4

• b\_2 = 2 \* 3 = 6

• e\_2 = (1 + 2)(4 + 3) - a\_2 - d\_2 = 11

Resultado = 4 \* 10^2 + 11 \* 10 + 6 = 516

**Se usa recursividad para el segundo subproblema:**

• d\_1 = 34 \* 21

Subproblemas:

• a\_2 = 3 \* 2 = 6

• b\_2 = 4 \* 1 = 4

• e\_2 = (3 +4)(2 + 1) - a\_2 - d\_2 = 11

Resultado = 6 \* 10^2 + 11 \* 10 + 4 = 714

**Se usa recursividad para el tercer subproblema:**

• e\_1 = 46 \* 64 - a\_1 - d\_1

Subproblemas:

• a\_2 = 4 \* 6 = 24

• d\_2 = 6 \* 4 = 24

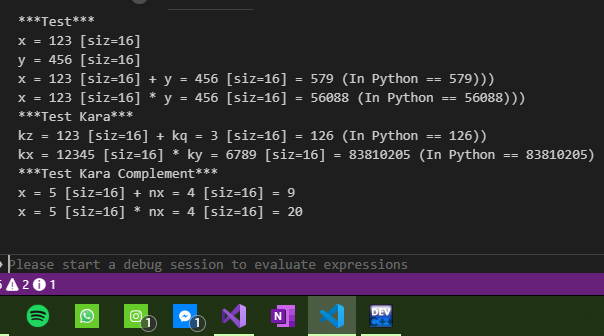
• e\_2 = (4 + 6)(6 + 4) - a\_2 - d\_2 = 52

Resultado = 24 \* 10^2 + 52 \* 10 + 24 - 714 - 516 = 1714

**Resultado final = 1234 \* 4321 = 516 \* 10^4 + 1714 \* 10^2 + 714 = 5,332,114**

**Nuestro algoritmo en Python:**

Bajo nuestra adaptación del código en Python, para el cual utilizamos el programa de de Visual Studio Code obtuvimos los siguientes resultados:



**Conclusión**

Con este trabajo, nosotros como compañeros conseguimos entender la importancia de conocer el tiempo de corrida de nuestros algoritmos para futuros trabajos, entendimos la importancia del algoritmo de Karatsuba en la informática moderna y que tanto significó en los años sesenta este mismo para la implementación de la multiplicación de maneras mucho más eficientes en los algoritmos siguientes, y de manera más abstracta, de cómo un trabajo universitario puede llegar a ser de gran relevancia en las distintas áreas de interés científico, social y humano , de cómo no debemos minimizar nuestro trabajo y tratar de siempres expandir nuestros límites en cuanto al conocimiento y apoyar siempre la curiosidad y emprendimiento en trabajos académicos en nuestros compañeros, profesores y hasta en nuestra vida privada.

**Bibliografía**

1. Algoritmo de Karatsuba. (s.f.). Recuperado de https://es.wikipedia.org/wiki/Algoritmo\_de\_Karatsuba
2. Karatsuba Multiplication. (s.f.). Recuperado de http://mathworld.wolfram.com/KaratsubaMultiplication.html
3. Karleigh Moore, K. M. (s.f.). Karatsuba Algorithm. Recuperado de https://brilliant.org/wiki/karatsuba-algorithm/
4. Kiko Correoso, K. C. (s.f.). PFS: Multiplicación de Karatsuba. Recuperado de https://www.pybonacci.org/2013/09/04/pfs-multiplicacion-de-karatsuba/
5. Wikimedia foundation. (2010). Anatoli Karatsuba. Recuperado de https://esacademic.com/dic.nsf/eswiki/79785