Algoritmo de Multiplicación Entera de Harvey y van der Hoeven

O(n $\log n$) : $Reducci\acute{o}ndeComplejidadComputacional$

Universidad La Salle
Análisis de Algoritmos
Estudiante: Esther Mariana Chunga Paheco
Docente: Edsona Mamani

1. Introducción

La multiplicación de enteros grandes constituye un pilar fundamental en disciplinas como la criptografía, el cálculo científico y la teoría de números computacional. El algoritmo de Harvey y van der Hoeven (2019) representa un hito revolucionario al alcanzar la complejidad óptica teórica de $O(n \log n)$, resolviendo un problema abierto durante más de cinco décadas.

Este algoritmo innova mediante tres componentes clave:

- **Descomposición Multivariante**: Transforma el polinomio univariado tradicional en una estructura multivariada $A(x_1, \ldots, x_k)$, permitiendo FFTs más pequeñas y paralelizables.
- **Aritmética en Anillos Especiales**: Utiliza anillos cociente Z/pZ con:
  - Primos especiales de la forma $p = 2^{2^k} + 1$ (Primos de Fermat generalizados)
  - Raíces primitivas de alta orden ω para transformadas eficientes ditem Estructura modular que evita errores de redondeo
- **Mecanismo de Recomposición**: Un método novedoso para combinar resultados parciales mediante:

$$C = \sum_{i=0}^{m} c_i 2^{64i} \quad \text{con} \quad c_i = \sum_{j+k=i} a_j b_k \tag{1}$$

El núcleo del algoritmo implementa una Transformada Teórico-Numérica (NTT) multivariante mediante cuatro etapas precisas: 1. *Segmentación*: Divide los enteros en dígitos base-2^{64} 2. *Mapeo Dimensional*: Convierte a polinomio en $(Z/pZ)[x_1, \ldots, x_k]$ 3. *Transformada Multidireccional*: Ejecuta NTTs univariadas en cada dimensión 4. *Reconstrucción Jerárquica*: Combina resultados usando el Teorema Chino del Resto

La Figura 1 ilustra este flujo, donde la complejidad se reduce al evitar las recursiones anidadas características de Schönhage-Strassen, eliminando así el factor $\log \log n$. Actualmente, aunque su implementación práctica requiere precauciones de memoria caché y optimización de acceso a datos, establece un nuevo paradigma en el diseño de algoritmos numéricos.

2. Contexto Histórico

- **Método escolar**: $O(n^2)$
- **Karatsuba (1963)**: $O(n^{\log_2 3}) \approx O(n^{1,585})$
- **Métodos basados en FFT (Schönhage-Strassen, 1971)**: $O(n \log n \log \log n)$
- **Harvey y van der Hoeven (2019)**: $O(n \log n)$

3. Base Teórica

3.1. Convolución y DFT

La multiplicación de enteros puede reducirse a convolución polinomial mediante la Transformada Discreta de Fourier (DFT):

$$(a * b)_k = \sum_{j=0}^{n-1} a_j b_{k-j} \tag{2}$$

El Teorema de Convolución establece:

$$\mathcal{F}(a * b) = \mathcal{F}(a) \cdot \mathcal{F}(b) \tag{3}$$

3.2. Transformada Rápida de Fourier (FFT)

La FFT calcula la DFT en $O(n \log n)$. Para multiplicación:

1. Convertir enteros en polinomios $A(x)$ y $B(x)$
2. Calcular $\mathcal{F}(A)$ y $\mathcal{F}(B)$
3. Multiplicar resultados en dominio frecuencial
4. Calcular FFT inversa para obtener $A(x) \cdot B(x)$

4. Algoritmo de Harvey y van der Hoeven

4.1. Innovaciones Clave

- **Polinomios multivariados**: Descomposición del problema en múltiples variables para reducir el tamaño de la FFT
- **Selección óptima de anillos**: Uso de anillos con raíces de la unidad para evitar problemas de precisión numérica
- **Manejo de coeficientes**: Evita pasos recursivos de FFT que introducen términos $\log \log n$

4.2. Análisis de Complejidad

Al optimizar la aritmética de anillos y parámetros FFT, el algoritmo elimina el factor $\log \log n$:

$$T(n) = O(n \log n)$$

5. Esquema de Código en Python

```python
import numpy as np

def multiplicar(a, b):
    n = len(a)
    # Rellenar para evitar convoluci n circular
    a_rellenado = np.fft.fft(a, 2*n)
    b_rellenado = np.fft.fft(b, 2*n)
    # Multiplicaci n componente a componente
    c_rellenado = a_rellenado * b_rellenado
    # FFT inversa
    c = np.fft.ifft(c_rellenado)
    # Manejar acarreos y convertir a entero
    resultado = np.zeros(2*n, dtype=int)
    acarreo = 0
    for i in range(2*n):
        acarreo += int(round(c[i].real))
        resultado[i] = acarreo % 10
        acarreo = acarreo // 10
    return resultado
```

Listing 1: Multiplicación basada en FFT (simplificada)

Nota: La implementación real usa transformadas teoricanuméricas (NTT) en campos finitos para aritmética exacta.

6. Comparación de Complejidad

7. Conclusión

El algoritmo de Harvey y van der Hoeven alcanza el límite teórico inferior para multiplicación de enteros mediante la optimización de FFTs multivariantes y aritmética de anillos. Aunque su implementación es altamente no trivial, representa un resultado histórico en la teoría de complejidad computacional.

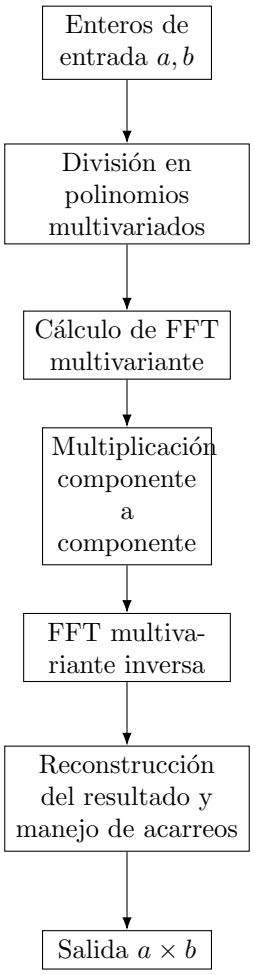

Figura 1: Flujo del algoritmo

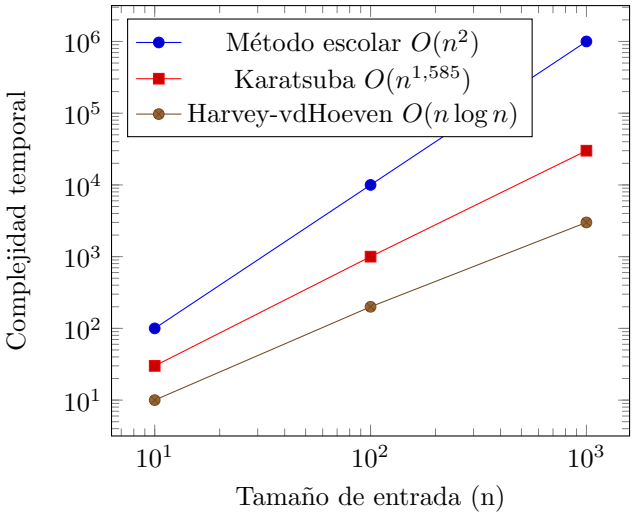

Figura 2: Comparación de complejidad asintótica (escala log-log)