INFORME PRÁCTICA MULTIPLICACIÓN DE MATRICES: ALGORITMO DE STRASSEN

Autor : Alien Embarec Riadi

Curso : Diseño y Análisis de Algoritmos, Tercero del grado de Ingeniería Informática, Universidad de la Laguna, Canarias, España**.**

Datos de contacto : alu0101035406@ull.edu.es

ÍNDICE

1. Introducción **3**

2. Análisis del algoritmo de Strassen y comparativa con otros algoritmos **3**

3. Conclusiones **4**

4. Bibliografía **4**

1. Introducción

La multiplicación de Strassen es uno de los algoritmos más empleados para resolver problemas en ciencias de la computación. Este algoritmo se diferencia del resto en que realiza siete multiplicaciones en vez de las 8 que emplean los métodos de multiplicación estándar.

Es asintóticamente más rápido que el algoritmo de multiplicación de matrices estándar, pero más lento que el algoritmo más rápido conocido, y es útil en la práctica para matrices grandes.

En este informe se analizarán los resultados de la implementación de este algoritmo, el tiempo de ejecución medio, aplicación del teorema maestro y árbol de recurrencia, así como una comparativa con los resultados de otros algoritmos de multiplicación de matrices, el recursivo y el método estándar de los tres bucles.

2. Análisis del algoritmo de Strassen y comparativa con otros algoritmos

Aplicando el método maestro, obtenemos la siguiente ecuación:

$$T(n) = \begin{cases} \Theta(1) & if\ n = 1 \\ 7T(n/2) + \Theta(n^2) & if\ n > 1 \end{cases}$$

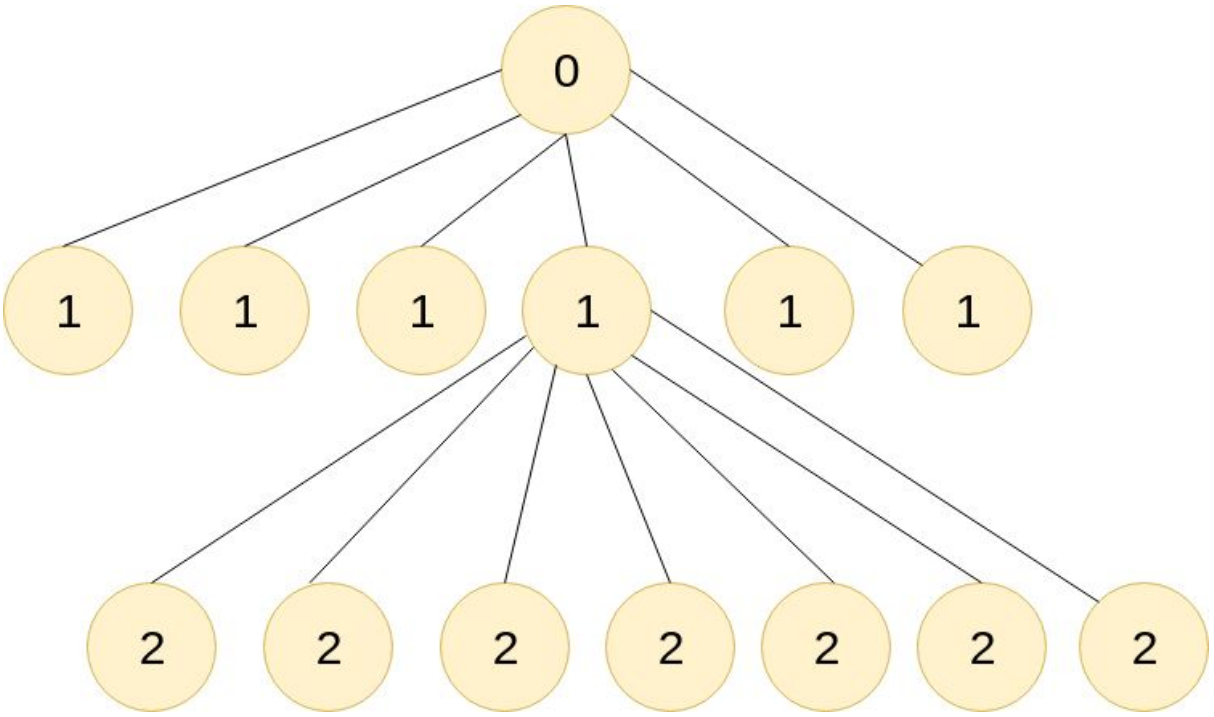
Donde:

- Para el caso base, cuando el tamaño de matriz se reduce hasta 1x1 con las llamadas recursivas, la operación de multiplicación se realiza en 1 paso.

- Cuando el tamaño de la entrada es mayor que 1:
    - **a = 7**, el número de subproblemas por cada subproblema, se realizan siete multiplicaciones en cada llamada, supone una mejora en la eficiencia frente a las 8 operaciones de los métodos de multiplicación tradicional.
    - **n/2** , tamaño de cada uno de los subproblemas, cada matriz se divide en 4 submatrices más pequeñas, el tamaño de estas matrices es, como se puede observar en el código, es de n/2.
    - **D(n) = Θ(1).** Tiempo invertido en dividir las matrices.
    - **C (n) = Θ(n^2).** Tiempo invertido en combinar.

El árbol de recursividad sería el que se muestra a continuación.

INFORME MULTIPLICACIÓN DE MATRICES



Cada problema en el nivel i se subdivide en 7 subproblemas en el nivel i+1, y así sucesivamente.

	Strassen			Recursivo			Tres bucles
Dimensión	Longitud camino	Nodos generados	CPU en ms	Longitud camino	Nodos generados	CPU en ms	
20x20	26	219	3,66	26	294	6,19	
20x20	16	63	*0,63	16	36	0,27	
20x20	14	43	0,051	14	28	0,0026	
20x20	18	56	0,45	18	128	1,339	
20x20	10	16	0,079	10	39	0,233	
30x30	34	240	5,26	34	140	1,94	
30x30	22	57	0,01	22	103	0,1	
30x30							
50x50	31	132	2,32	31	199	3,49	
50x50	78	507	0,10	78	600	0,10	

50x50	31	94	0,93	31	100	1,00	
70x70	29	197	6,12	29	314	19,86	

3. Conclusiones

En primer lugar, se ha podido comprobar que para entradas pequeñas (inferiores a 128) el algoritmo de Strassen no mejora sustancialmente los tiempos de ejecución de los algoritmos estándar y recursivo. No obstante, la diferencia en favor del primer algoritmo se va haciendo más notoria a medida que aumenta el tamaño de la instancia (a partir de), que es donde reside la utilidad de la eficiencia computacional en la multiplicación de matrices, con tiempos de ejecución de frente a .

Por otro lado, se demuestra la utilidad del enfoque Divide y Vencerás, permitiendo subdividir el problema en subproblemas más pequeños hasta llegar a un trivial, y a partir de ahí recomponer las soluciones, de otro modo no hubiera sido posible implementar la multiplicación de Strassen preservando la eficiencia.

4. Bibliografía

- “Algoritmo De Strassen.” *Wikipedia*, Wikimedia Foundation, 13 May 2018, es.wikipedia.org/wiki/Algoritmo_de_Strassen.
- Ambarmodi. “Ambarmodi/Strassen-Multiplication.” *GitHub*, 26 May 2017, github.com/ambarmodi/Strassen-Multiplication.
- JohannaJohanna. “Exception in Thread ‘Main’ Java.lang.OutOfMemoryError: Java Heap Space.” *Stack Overflow*,stackoverflow.com/questions/2381849/exception-in-thread-main-java-lang-outofmemoryerror-java-heap-space.