Taller multiplicación de matrices

Jose Fernando Zuluaga, Nicolas Daniel Vargas

[1]Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana
Bogotá, Colombia
{zuluaga_jose@javeriana.edu.co

1 de septiembre de 2022

Resumen

En este documento se presenta la documentación correspondiente al taller 3 de análisis de algoritmos, donde se muestra el desarrollo y descripción del algoritmo planteado como solución al problema de multiplicación de matrices haciendo uso del método de *backtracking* **Palabras clave:** iterativo, algoritmo, formalización, experimentación, complejidad, dividir y vencer.

Índice

1. Introducción

En matemática, una matriz es un conjunto bidimensional de números. Dado que puede definirse tanto la suma como el producto de matrices, en mayor generalidad se dice que son elementos de un anillo. Se compone por un numero de filas y columnas, no necesariamente iguales. Esta diferencia de tamaños entre filas y columnas, es importante de tener en cuenta al realizar la operación de multiplicación, producto, entre matrices, de tal forma, que para llevarse acabo la operación de A*B=C, se debe cumplir con la condición de igualdad de tamaño entre las columnasA y filasB, como resultado se obtendrá C que tendrá como tamaño filasA x columnasB.

2. Formalización del problema

Cuando se quiere multiplicar una secuencia de matrices $A1, A2 \cdots An$ es importante saber cuál es la mejor forma de agrupa(asociar), de dos en dos, las multiplicaciones matriciales para reducir la cantidad de multiplicaciones escalares. Por ejemplo, si se tiene $A1 : R10\text{x}100$, $A2 : R100\times5$ y $A3 : R5\times50$:

- La agrupación ((A1A2) A3) implica 7500 multiplicaciones escalares.
- La agrupación (A1 (A2A3)) implica 75000 multiplicaciones escalares.

2.1. Definición del problema de "multiplicación de matrices"

Así, el problema de la multiplicación de matrices se define a partir de:

- $A = (A1, A2, A3, \cdots, An_i)$, donde $A_i \in R^{r_i c_i}$ $r_i = c_i 1$.
- $A1,_k\, A_{k+1,n}$ $(A1 \cdots Ak)\ (Ak+1 \cdots An)$, donde $1 \leq k \leq n$.

Entonces, $M_{i,j}$ es el número óptimo de multiplicaciones escalares al agrupar las matrices $A_{i,j}$:

- $M_{i,j} \rightarrow 0$;i = j
- $M_{i,j} \rightarrow \min_{i \leq k < j} M_{i,k} + M_{k+1,j} + m_{ikj}$; $i \neq j$

donde m_{ikj} es número de multiplicaciones escalares para calcular$A1,_k\, A_{k+1,j}$.

Dado que:

$$A_i A_{i+1} \in R^{r_i c_{i+1}}, A_i A_{i+1} A_{i+2} \in R^{r_i c_{i+2}}, ..., A_{i,k} \in R_{r_i c_k}, A_{k+1,j} \in R^{r_{k+1} c_j} y r_{k+1} = c_k,$$

- Entradas:
  - S secuencia de números naturales que representa el tamaño de filas y columnas de la secuencia de matrices
- Salidas:
  - n numero óptimo de multiplicaciones a realizar en la correspondiente operación entre matrices, con el correcto uso de paréntesis

3. Algoritmos de solución

Para el correcto uso del ejercicio, se utiliza la memoria dinamica, lo cual significa utilizar los pasos correspondientes para llegar a una solución de tal manera, que se puedan emplear tabulación, tablas, y de tal forma que se llenen de forma correcta de acuerdo de la información analizada y recibida del problema

Algoritmo 1 Backtracking

Require: B : Matriz donde se guardara el resultado del backtracking, s donde representa el principio de la secuencia con los tamaños de las matrices, e final de la secuencia donde se tiene la información de las matrices.

Ensure: $n \in N$ donde representa el numero óptimo de multiplicaciones en el producto de matrices

```
1:  procedure BACKTRACKING(B, s, e)
2:      if s=e then
3:          return A*string[s]
4:      end if
5:      if s+1 = e then
6:          return A*string(s)*A*string(e)
7:      end if
8:      pos ← B_{s,e}
10:     cadenaA ← BACKTRACKING(B, s, pos)        cadenaB ← BACKTRACKING(B, s, pos)
12:         return (cadenaA)*(cadenaB)
13:     end procedure
14:
```

3.0.1. Análisis de complejidad

Por inspección de código: no hay ningún ciclo, sin embargo contiene dos llamados recursivos a la misma función, se puede ver $logN$ la complejidad de este algoritmo, similar a una búsqueda binaria. En este algoritmo, se valida que al menos haya mas de dos matrices, en caso de que solo exista una matriz, solo es necesario retornarla, pues no se realiza ninguna operación en caso de solo un elemento, y si existen dos matrices, no tiene importancia de agrupamiento de las matrices, debido a que no es relevante el orden mas allá del tamaño de sus dimensiones.

4. Análisis Experimental

Para la parte experimental, se cuenta con una conclusión acerca del uso de la programación dinámica, donde se realiza el backtracking para construir la solución de vuelta. Se obtiene que el tiempo de ejecución aumenta directamente proporcionar a la cantidad de matrices en la secuencia, sin embargo, no tiene relación al numero de multiplicaciones y operaciones realizadas.

```
7×8 Matrix{Float64}:
 0.0  10080.0  14280.0  57080.0  15930.0  36930.0
 0.0      0.0   1200.0  17000.0   5850.0  11850.0
 0.0      0.0      0.0   5000.0   5250.0   6500.0
 0.0      0.0      0.0      0.0   5000.0   7500.0
 0.0      0.0      0.0      0.0      0.0  25000.0
 0.0      0.0      0.0      0.0      0.0      0.0
 0.0      0.0      0.0      0.0      0.0      0.0
5))))(A6)
El numero minimo de multiplicaciones es  36930.0
```

Figura 1: Captura de pantalla durante una sesión.