Análisis de Algoritmos, Sem: 2018-1, 3CV2 Práctica 7, 2 de Noviembre del 2017

Práctica 7: Multiplicacion de una secuencia de matrices

Martínez Berumen Luis Daniel

Escuela Superior de Computo
Instituto Politecnico Nacional
dany.berumen@gmail.com

Abstract

En esta Practica analizaremos la complejidad del algoritmo de la multiplicacion secuencial de matrices, así como demostraremos su funcionamiento, daremos un ejemplo de como funciona con matrices aleatorias.

Palabras Clave

- Algoritmo
- Complejidad
- Multiplicación
- Secuencia
- Dinámica

1. Introducción

Divide y Vencerás es una frase quye hemos escuchado todos, al menos, una vez en nuestra vida, para nosotros es técnica de diseño de algoritmos, siendo de gran utilidad para nuestra carrera, ya que, los problemas a los que nos enfrentamos día con día son mas faciles de resolver si aplciamos una tecnica de este tipo. De hecho, suele ser considerada una filosofía general para resolver problemas, no solo del termino informatico, sino que también se utiliza en muchos otros ámbitos

2. Conceptos Básicos

Para la correcta comprension de este trabajo, es necesario definir algunos terminos tales como θ, O y Ω.
θ(n):
Sea g(n) una función. Se define θ (g(n)) como:

$\theta(\text{g(n)}) = \{f(n) \quad | \quad \exists c1, c2 > 0 \quad \& \quad n_0 > 0 \quad | \quad \forall n >= n_0 \quad 0 <= c1g(n) <= f(n) <= c2g(n)\}$

O(n):
Sea g(n) una función, O(n) (el pero de los casos) se define como:

$$\text{O(n)}=\{f(n) \quad | \quad \exists c > 0 \quad \& \quad n_0 > 0 \quad | \quad f(n) <= Cg(n) \quad \forall n >= n_0\}$$

Ω(n):
Sea g(n) una función. Se define Ω (g(n)) (el mejor de los casos) como:

$$\Omega(\text{g(n)}) =\{f(n) \quad | \quad \exists c > 0 \quad \& \quad n_0 > 0 \quad | \quad 0 <= cg(n) <= f(n) \quad \forall n >= n_0\}$$

Dentro del trabajo solo se observará la ejecución del algoritmo de multiplicación secuencial de matrices, corroborando la salida del programa, con el resultado esperado, posteriormente se planteará la demostraciñon de la función que rige el algoritmo.

3. Experimentación y Resultados

3.1. 1 Implementar el algoritmo de multiplicación secuencial de matrices

i) Como entrada el algoritmo tendrá n matrices A_i de tamaño p_{i-1} X p_i.

```
ArrayList<Integer> m = new ArrayList<Integer>();
m.add(3);
m.add(5);
m.add(6);
m.add(5);
m.add(3);
m.add(3);
```

Figura 1. Entrada de 5 matrices.

Entrada : 3,5,6,5,3,3 (Hace referencia implícitamente a cinco matrices $A_i = A_{3x5}$, $A_2 = A_{5x6}$, $A_3 = A_{6x5}$,$A_4 = A_{5x3}$, $A_5 = A_{3x3}$).

ii) Como salida, se mostrará la configuración de paréntesis que genera el algoritmo.

```
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
Solucion optima: ((A1(A2(A3A4)))A5)
```

Figura 2. «Parentizacion» optima para las 5 matrices.

iii)Además se mostrarán todas las configuraciones de paréntesis y se corroborará que la generada en el punto ii) en efecto, es la óptima .

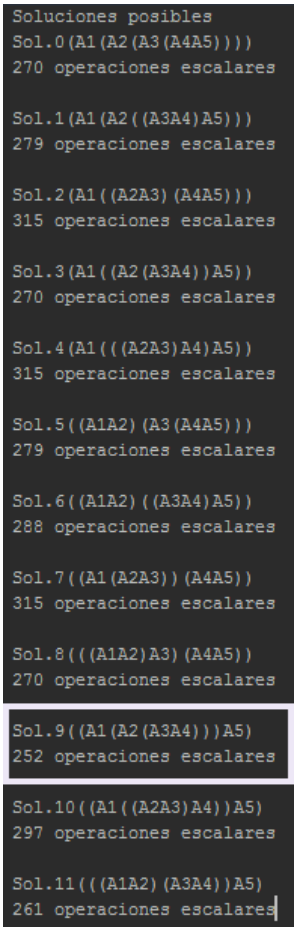

Figura 3. Todas las configuraciones.

Ejemplo. **Entrada** : 3,5,2,2 (Hace referencia implícitamente a tres matrices $A_i = A_{3x5}$, $A_2 = A_{5x2}, A_3 = A_{2x2}$).
Salida :
La configuración que genera el algoritmo de multiplicación de una secuencia de matrices es: $((A_1A_2)A3)$.
Muestra todas las configuraciones y las operaciones escalares realizadas.
$(A_1(A_2A_3))$ realiza 20 operaciones escalares
$((A_1A_2)A_3)$ realiza 42 operaciones escalares
Por lo tanto una configuración óptima es $((A_1A_2)A_3)$.

4. Conclusión

La ideología de divide y vencerás hace el problema más mas sencillo de analizar, pero no siempre se está seguro que la solución presentada sea la más óptima.
Es importante dejar claro que el analisis de algoritmos es una parte muy importante de la carrera, ya que los vamos a utilizar SIEMPRE, dicho esto es importante recalcar que no todos los algoritmos pueden ser la mejor opcion.
Por eso es fundamental saber obtener los mejores resultados para simplificar tareas grandes, obteniendo así un ahorro tanto en tiempo como en recursos.

5. Anexos

Problema : Utilizando método por sustitución muestre que la ecuación de recurrencia P(n)= $\sum_{i=0}^{n-1} P(K)P(n-k)$ si n¿=2 y P(n=1)=1 cumple con la definición 1.5 Ω (2^n

6. Bibliografía

- Brassard, G. (1997). Fundamentos de Algoritmia. España: Ed. Prentice Hall. ISBN 848966000X
- Harel, D. (2004). Algorithmics: The spirit of Computing (3rd. Ed). Estados Unidos de América: Addison Wesley. ISBN-13: 978-0321117847