MO417 – Análise de Algoritmos

Ata do exercício 15.2-1

Aula do dia 14/04/2009

Redator: Renato Hirata

Livro adotado: Cormen, T.H.;Leiserson, C.E.; Rivest, R.L.; Stein, C.; Algoritmos. Tradução da 2ª edição americana Teoria e Prática. 2002. Página 271

(15.2-1) Encontre uma colocação ótima dos parênteses de um produto de cadeias de matrizes cuja seqüência de dimensões é <5, 10, 3, 12, 5, 50, 6>.

Usando o algoritmo MATRIX-CHAIN-ORDER, obtemos as tabelas:

Tabela m

j: 1	2	3	4	5	6	i
0	150	330	405	1655	2010	1
	0	360	330	2430	1950	2
		0	180	930	1770	3
			0	3000	1860	4
				0	1500	5
					0	6

Tabela s

j: 2	3	4	5	6	i
1	2	2	4	2	1
	2	2	2	2	2
		3	4	4	3
			4	4	4
				5	5

Na tabela m, cada entrada m[i, j] representa o número mínimo de multiplicações escalares necessárias para calcular a matriz $A_{i..j}$. Assim, m fornece os custos de soluções ótimas para subproblemas.

Fórmula Geral: $m_{ij} = \min\limits_{i \le k < j}\{m_{ik} + m_{k+1,j} + p_{i-1}p_k p_j\}$

Onde

$p_0 = 5 \quad p_1 = 10 \quad p_2 = 3 \quad p_3 = 12 \quad p_4 = 5 \quad p_5 = 50 \quad p_6 = 6$

$m_{11} = 0 \quad m_{22} = 0 \quad m_{33} = 0 \quad m_{44} = 0 \quad m_{55} = 0 \quad m_{66} = 0$

Alguns exemplos de cálculos:

$$m_{12} = \min_{k=1}\{\underbrace{m_{11} + m_{22} + p_0 p_1 p_2}_{k=1}\} = \min\{\underbrace{0 + 0 + 5.10.3}_{k=1}\} = 150 \quad (k = 1)$$

$$m_{23} = \min_{k=2}\{\underbrace{m_{22} + m_{33} + p_1 p_2 p_3}_{k=2}\} = \min\{\underbrace{0 + 0 + 10 \cdot 3 \cdot 12}_{k=2}\} = 360 \quad (k = 2)$$

(...)

$$m_{13} = \min_{k=1,2}\{\underbrace{m_{11} + m_{23} + p_0 p_1 p_3}_{k=1}, \underbrace{m_{12} + m_{33} + p_0 p_2 p_3}_{k=2}\}$$
$$= \min\{\underbrace{0 + 360 + 5 \cdot 10 \cdot 12}_{k=1}, \underbrace{150 + 0 + 5 \cdot 3 \cdot 12}_{k=2}\} = \min\{960, 330\} = 330 \qquad (k = 2)$$

No caso da disposição da tabela m apresentada, as posições são preenchidas por diagonais começando na diagonal acima da principal, seguindo em direção à posição $m_{1,6.}$

A tabela s nos ajuda a controlar o modo de construir uma solução ótima. Como vimos, a cada cálculo de m[i,j] obtemos um k, e esse valor é posicionado em s[i,j]. Assim s[i,j] representa o valor de k no qual podemos dividir o produto $A_{i..j}$ para obter uma colocação ótima dos parênteses (um número mínimo de multiplicações escalares).

O algoritmo recursivo PRINT-OPTIMAL-PARENS utiliza a tabela s para construir a solução ótima. A idéia do algoritmo é encontrar o ponto de quebra (onde imprimir os parênteses) e recursivamente fazer o mesmo para as duas subcadeias resultantes.

No caso, a entrada s[1,6] = 2 indica que a primeira quebra (de $A_{1..6}$) deve ser executada entre A_2 e A_3. A entrada s[3,6] = 4, indica que a quebra da subcadeia ($A_{3..6}$) restante deve ser feita entre A_4 e $A_{5.}$

Assim resultado obtido é $((A_1A_2)((A_3A_4)(A_5A_6)))$, com custo de 2010 multiplicações escalares.