Instituto Politécnico Nacional

Escuela Superior de Cómputo

Sesión 6: Multiplicación de matrices

Integrantes:

Escutia López Arturo

López Santiago Daniel

14/OCT/2016

Ejemplo con Matriz de 4x4

Arturo Ezutia López 5/Oct/2016
Daniel López Santiago

¡Díaz! ¡PCV!

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 5 & 6 & 7 & 8 \\ 8 & 7 & 6 & 5 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 & 8 & 9 & 6 \\ 3 & 2 & 0 & 1 \\ 3 & 4 & 2 & 1 \\ 8 & 0 & 1 & 0 \end{bmatrix}$$

A, B, C, D (blocks of X); E, F, G, H (blocks of Y)

$$P_1 = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 7 & 5 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 7 \\ 25 & 23 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 4 & 6 \\ 6 & 4 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 14 & 4 \\ 16 & 6 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} 12 & 14 \\ 14 & 12 \end{bmatrix} \begin{bmatrix} 1 & 8 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} 54 & 124 \\ 50 & 136 \end{bmatrix}$$

$$P_4 = \begin{bmatrix} 7 & 8 \\ 6 & 5 \end{bmatrix} \begin{bmatrix} 2 & -4 \\ 5 & -2 \end{bmatrix} = \begin{bmatrix} 54 & -44 \\ 37 & -34 \end{bmatrix}$$

$$P_5 = \begin{bmatrix} 8 & 10 \\ 10 & 8 \end{bmatrix} \begin{bmatrix} 3 & 9 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} 64 & 92 \\ 62 & 106 \end{bmatrix}$$

$$P_6 = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix} \begin{bmatrix} 5 & 5 \\ 9 & 0 \end{bmatrix} = \begin{bmatrix} -56 & -20 \\ -56 & -20 \end{bmatrix}$$

$$P_7 = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix} \begin{bmatrix} 10 & 14 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} -52 & -68 \\ -52 & -68 \end{bmatrix}$$

$$XY = \begin{bmatrix} 48 & 24 & 19 & 11 \\ 27 & 46 & 41 & 29 \\ 108 & 80 & 67 & 43 \\ 87 & 102 & 89 & 61 \end{bmatrix}$$

$P_1 = 1(4) = 4$

$P_2 = 3(1) = 3$

$P_3 = 7(7) = 49$

$P_4 = 3(-8) = -24$

$P_5 = 4 \cdot 8 = 32$

$P_6 = -1 \cdot 0 = 0$

$P_7 = -3 \cdot 12 = -36$

$$\begin{bmatrix} 5 & 7 \\ 25 & 23 \end{bmatrix}$$

```java
    private static int[][] Strassen(int X[][], int Y[][]){

            if(X.length==1) {
                int R[][]=new int [1][1];
                R[0][0]=X[0][0]*Y[0][0];
                return R;
            }
            int A[][]=Arrays.copyOfRange(X,0,X.length/2);
            int B[][]=RSubmatrix(X,0);
            int C[][]=Arrays.copyOfRange(X,X.length/2,X.length);
            int D[][]=RSubmatrix(X,X.length/2);
            int E[][]=Arrays.copyOfRange(Y,0,Y.length/2);
            int F[][]=RSubmatrix(Y,0);
            int G[][]=Arrays.copyOfRange(Y,Y.length/2,Y.length);
            int H[][]=RSubmatrix(Y,Y.length/2);

            int P1[][]=Strassen(A,OpM(F,H,"-"));
            int P2[][]=Strassen(OpM(A,B,"+"),H);
            int P3[][]=Strassen(OpM(C,D,"+"),E);
            int P4[][]=Strassen(D,OpM(G,E,"-"));
            int P5[][]=Strassen(OpM(A,D,"+"),OpM(E,H,"+"));
            int P6[][]=Strassen(OpM(B,D,"-"),OpM(G,H,"+"));
            int P7[][]=Strassen(OpM(A,C,"-"),OpM(E,F,"+"));

            int R1[][]=OpM(OpM(P5,P4,"+"),OpM(P2,P6,"-"),"-");
            int R2[][]=OpM(P1,P2,"+");
            int R3[][]=OpM(P3,P4,"+");
            int R4[][]=OpM(OpM(P5,P1,"+"),OpM(P3,P7,"+"),"-");;

            return JoinM(R1,R2,R3,R4);

    }
}
```

Esta es la función STRASSEN la que recibe como parámetros las dos matrices a multiplicar X ,Y primeramente debemos partir a la matriz por la mitad como nos lo indica el algoritmo en 4 submatrices por cada X,Y, esto se visualiza en el código al declarar los arrays de dos dimensiones y mandar a llamar las funciones correspondientes para dividir las secciones de las submatrices, una vez que obtenemos esas submatrices llamamos recursivamente la función STRASSEN para obtener 7 submatrices, resultado de algunas sumas,restas y multiplicaciones según la fórmula de cada caso como se ve a continuación

$$P_1 = A(F - H) \qquad P_5 = (A + D)(E + H)$$

$$P_2 = (A + B)H \qquad P_6 = (B - D)(G + H)$$

$$P_3 = (C + D)E \qquad P_7 = (A - C)(E + F)$$

$$P_4 = D(G - E)$$

La función se detiene cuando la dimensión de la matriz es de 1x1 por lo que la multiplicación original se reduce a la multiplicación de matrices de 1x1 junto con sumas y restas. Ya solamente queda realizar la unión de los R's y retornar el valor ya "reconstruido" con las submatrices que será el resultado total de la multiplicación entre X*Y, en el código :

- R1=P5+P4-P2+P6
- R2=P1+P2
- R3=P3+P4
- R4=P1+P5-P3-P7 y por ultimo R seria X*Y

$$XY = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

Implementación

Matriz X:

```
1,4,5,6,1,5,6,9
1,5,6,9,2,4,5,7
2,4,5,7,1,4,5,6
2,4,5,7,2,4,5,7
1,5,6,9,2,4,5,7
2,4,5,7,1,4,5,6
1,4,5,6,1,4,5,6
2,4,5,7,1,5,6,9
```

Matriz Y:

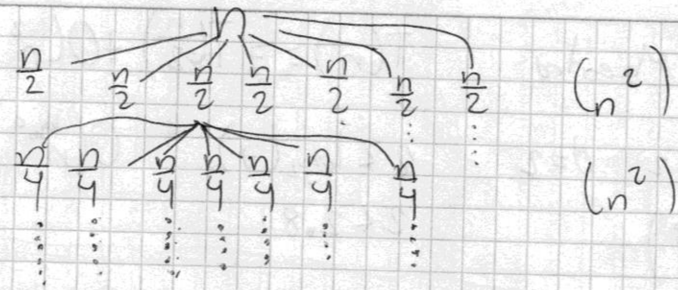
```
2,4,5,7,1,5,6,9
2,4,5,7,1,4,5,6
1,4,5,6,1,4,5,6
1,5,6,9,2,4,5,7
1,4,5,6,1,4,5,6
2,4,5,7,1,5,6,9
2,4,5,7,2,4,5,7
1,5,6,9,2,4,5,7
```

Resultado

```
lopesc@lopesc-Lenovo-G40-45 ~/Desktop/algoritmos/p6
File  Edit  View  Search  Terminal  Help
lopesc@lopesc-Lenovo-G40-45 ~/Desktop/algoritmos/p6 $ java Strassen X Y
53 163 200 283 58 154 191 261
54 172 211 297 60 161 200 270
49 149 183 258 52 142 176 240
51 158 194 273 55 150 186 253
54 172 211 297 60 161 200 270
49 149 183 258 52 142 176 240
46 140 172 242 49 133 165 224
56 172 211 299 61 163 202 277

lopesc@lopesc-Lenovo-G40-45 ~/Desktop/algoritmos/p6 $
```

Analisis de complejidad



$N_1 = 7$ subproblemas $\frac{n}{2}$ (n^2) $\qquad 7^j\, O\left(\frac{n}{2^j}\right) = \frac{7^j Cn}{2^j}$

$N_2 = 49$ subproblemas $\frac{n}{4}$ (n^2)

$N_3 = 343$ subproblemas $\frac{n}{8}$ (n^2)

$N_j = 7^j$ subproblemas $n/2^j$ (n^2)

$$Cn \sum_{j=0}^{\log n} \left(\frac{7}{2}\right)^j = Cn\left[\frac{\left(\frac{7}{2}\right)^{\log n}\left(\frac{7}{2}\right) - 1}{\frac{7}{2} - 1}\right] = Cn\left[\frac{\frac{7^{\log n}}{2^{\log n}}\left(\frac{7}{2}\right) - 1}{\frac{5}{2}}\right] =$$

$$Cn\left[\frac{n^{\log 7}}{n^{\log(2)}}\left(\frac{7}{\cancel{2}}\right)\left(\frac{\cancel{2}}{5}\right) - \left(\frac{2}{5}\right)\right] = Cn\left[n^{\log(7)-1}\left(\frac{7}{5}\right) - \frac{2}{5}\right] =$$

$$Cn\left[\frac{7}{5} n^{1.8} - \frac{2}{5}\right] = \frac{7}{5} Cn^{2.8} - \frac{2}{5} Cn \quad \therefore \quad O(n^{2.8})$$

Teorema
~~Método~~ Maestro $T(n) = 7\,T(\frac{n}{2}) + O(n^2)$

$a=7$ $b=2$ $d=2$ $2 < \log_2(7)$ $O(n^{\log_b a})$ $d < \log_b a$

$\therefore O(n^{2.8})$ $2 < 2.8$

Supoción $T(n) = O(n^{2.8})$

$T(\frac{n}{2}) = c\left(\frac{n}{2}\right)^{2.8}$ $T(n) = \frac{7cn^{2.8}}{2^{2.8}} + O(n^2)$

$= \frac{7}{2^{2.8}} cn^{2.8} + n^2$

$\frac{7}{2^{2.8}} n^{2.8} + n^2 \leq cn^{2.8}$ ✓

para alguna c si $c = 2$ $n \geq n_0$