Tarea # 2

Análisis y Diseño de Algoritmos / Ingeniería Civil Informática
Departamento Ciencias de la Computación y
Tecnologías de la Información

Universidad del Bío-Bío

Prof.: Gilberto Gutiérrez Ayud. Israel Gajardo

Primavera 2018

Multiplicación encadenada de matrices. Dadas dos matrices $A_{p\times q}$ y $B_{q\times r}$ el producto de AB se define como sigue

$$C_{p\times r} = AB = \sum_{k=1}^{n} a_{ik}b_{jk}, 1 \leq i \leq p, 1 \leq j \leq r$$

Algorítmicamente se puede expresar de la siguiente forma:

```
for (i = 1; i <= p; i++)
  for (j = 1; j <= r; j++) {
    C[i,j]= 0
    for (k = 1; k <= q; k++)
      C[i,j]= C[i,j] + A[i,k] * B[k,j]
  }
```

de donde se deduce que se necesitan pqr multiplicaciones de escalares (me). Supongamos ahora que se desea calcular el producto de más de dos matrices. El producto matricial es asociativo, por lo tanto es posible calcular el producto matricial $M = M_1M_2 \dots M_n$ de muchas maneras diferentes, todas las cuales darán la misma respuesta:

$$M = (\dots((M_1M_2))M_3)\dots M_n)$$

$$M = (M_1(M_2(M_3 \dots (M_{n-1}M_n)\dots)))$$

$$M = (\dots((M_1M_2)(M_3M_4))\dots)$$

y así sucesivamente. Recordar que la multiplicación de matrices no es conmutativa por lo que no es posible modificar el orden las multiplicaciones matriciales.

La selección del método de cálculo puede influir de manera importante en el tiempo requerido para realizar las multiplicaciones de las matrices. A modo de ejemplo suponga que necesita calcular el producto de las matrices $ABCD$ en donde $A_{13\times 5}, B_{5\times 89}, C_{89\times 3}$ y $D_{3\times 34}$. Para medir la eficiencia de cada posible forma de multiplicar $ABCD$ usaremos como medida la cantidad de multiplicación de escalares que se necesitan hacer en cada caso. Un caso es $M = ((AB)C)D$ lo que nos da:

AB	5.785 multiplicaciones
$(AB)C$	3.471 multiplicaciones
$((AB)C)D$	1.326 multiplicaciones
Total	10.582 multiplicaciones

Hay cinco forma más de calcular de manera diferente el producto de $ABCD$. El problema es encontrar la forma óptima de multiplicar las matrices. Es decir, cual es la forma de multiplicar las matrices de tal manera que la cantidad de multiplicaciones de escalares sea la mínima.

Una forma sencilla de poder hallar la mejor forma de multiplicar las matrices, sería encontrar todas las maneras posibles de poner los paréntesis en la expresión y contar el número de multiplicaciones de escalares que es necesario realizar.

Sea $T(n)$ el número de formas diferentes de poner paréntesis en un producto de n matrices. Supongamos que decidimos hacer el primer corte entre las matrices i-ésima e $(i+1)$-ésima del producto en la forma siguiente:

$$(M_1 M_2 \dots M_i)(M_{i+1} M_{i+2} \dots M_n)$$

Ahora hay $T(i)$ formas de poner los paréntesis en el término del lado izquierdo y $T(n-i)$ en el término del lado derecho. Cualquier forma del primer grupo se puede combinar con cualquier forma del segundo grupo. De esta forma y para este valor concreto de i hay $T(i)T(n-i)$ formas de poner los paréntesis. Pero dado que i puede tomar cualquier valor entre 1 y $n-1$ obtenemos finalmente la recurrencia siguiente:

$$T(n) = \sum_{i=1}^{n-1} T(i)T(n-i)$$

con $T(1) = 1$. La tabla siguiente muestra algunos valores de $T(n)$[1]

n	1	2	3	4	5	10	15
$T(n)$	1	1	2	5	14	4.862	2.274.440

$T(n) = \Omega(\frac{4^n}{n^2})$ y se necesita tiempo $\Omega(n)$ para calcular la cantidad de multiplicaciones escalares por lo tanto se requiere tiempo (en el mejor de los casos) $\Omega(\frac{4^n}{n})$ lo que evidentemente no es práctico.

En clases se explicó el principio de optimalidad para este problema y que consiste en lo siguiente. Asumamos que la mejor forma de multiplicar las matrices nos exige hacer el primer corte entre la i-ésima e $(i+1)$-ésima matriz, entonces los dos subproductos $M_1 M_2 \dots M_i$ y $M_{i+1} M_{i+2} M_n$ también deben calcularse de manera óptima. Sea $me(i,j)$ la cantidad de multiplicaciones de escalares para multiplicar las matrices $M_i \dots M_j$ y sea $s = i-j$. Existen dos casos bases. El primero cuando tenemos $s = 0$, en tal caso $me(i,i) = 0$. El segundo corresponde cuando se tienen que multiplicar dos matrices, es decir, cuando $s = 1$. En este caso $me(i,j)$ se calcula directamente usando las dimensiones de las matrices $M_i M_j$. El caso general corresponde cuando $1 < s < n$ y en tal caso $me(i,j) = min_{i \le k < i+s}(me(i,k) + me(k+1, i+s) + d_{i-1} d_k d_{i+s})$, es decir se necesitan $me(i,k)$ para el término de la izquierda (de dimensiones $d_{i-1} \times d_k$), $me(k+1, i+s)$ (de dimensiones $d_k d_{i+s}$) y $d_{i-1} \times d_k d_{i+s}$ para multiplicar las dos matrices resultantes.

En esta tarea usted debe:

a) Implementar un algoritmo recursivo (dividir para reinar) para calcular $me(1,n)$

b) Implementar un algoritmo de programación dinámica (visto en clases) para calcular $me(1,n)$.

[1] Los valores de $T(n)$ se llaman números de Catalan

c) Obtener la complejidad del algoritmo de programación dinámica.

d) Generar secuencias de tamaño 5, 10, 15, 20, 25 y 30 válidas[2] de multiplicaciones de matrices y mida el tiempo en cada caso para cada algoritmo (el de dividir para reinar y el de programación dinámica). Si el algoritmo de dividir para reinar toma demasiado tiempo (más de una hora), anote como respuesta ∞.

e) Extienda el algoritmo de programación dinámica, de tal manera que este entregue la forma de multiplicar las matrices. Por ejemplo, para la multiplicación de las matrices $ABCD$ entregar $((ABC)D)$ si esta forma es la que minimiza la multiplicación de escalares.

Forma de entrega: La entrega de la tarea consiste en:

a) Archivos fuentes de los programas (JAVA o C)

b) Archivos ejecutables

c) Un archivo llamado IR.doc conteniendo:

- Las instrucciones para ejecutar los programas
- Los resultados de ejecutar los algoritmos con secuencias de multiplicación de matrices de tamaño 5, 10, 15, 20, 25 y 30.
- Cálculo de la complejidad del algoritmo de programación dinámica. Complejidad y explicación.

Todos los archivos indicados arriba en un único archivo .rar, .zip u otro.

Fecha de Entrega: 08 de Noviembre de 2018. **Grupos:** 3 estudiantes máximo

[2]que se puedan multiplicar de acuerdo a las restricciones de la multiplicación de matrices. Por ejemplo, la secuencia $A_{10\times5}B_{8\times20}$ no se pueden multiplicar pues las dimensiones de A y B no lo permiten.