PROGRAMACIÓN DINÁMICA:

MULTIPLICACIÓN ENCADENADA DE MATRICES

ÍNDICE

- Programación dinámica
- Multiplicación encadenada de matrices
- Método de fuerza bruta
- Método con programación dinámica

PROGRAMACIÓN DINÁMICA

1. Dividir el problema en subproblemas más pequeños.
2. Resolver estos problemas de manera óptima usando este proceso de tres pasos recursivamente.
3. Usar estas soluciones óptimas para construir una solución óptima al problema original.

MULTIPLICACIÓN ENCADENADA DE MATRICES

- Suponemos que tenemos las matrices A(13x5), B(5x89), C(89x3) y D(3x34). Y queremos multiplicarlas todas, por lo tanto tenemos 5 posibilidades:
- ((AB)C)D número de productos = 10.582
- (AB)(CD) número de productos = 54.201
- (A(BC))D número de productos = 2.856
- A((BC)D) número de productos = 4.055
- A(B(CD)) número de productos = 26.418

ALGORITMO DE FUERZA BRUTA

- Sea T(n) el número de formas distintas 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.

- Ahora hay T(i) formas de poner paréntesis en el término del lado izquierdo, y T(n-i) formas de poner paréntesis en el término del lado derecho.

- Cualquier forma de las primeras se puede combinar con cualquiera de las segundas, así que para este valor concreto de i hay T(i)T(n-i) formas de poner paréntesis en toda la expresión. Dado que i puede tomar cualquier valor entre 1 y n-1, obtenemos la recurrencia:

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

- Para cada manera de poner paréntesis en M, se necesita un tiempo que está en O(n). Dado que T(n) esta en $O\left(\frac{4^n}{n^2}\right)$ la búsqueda de la mejor manera de calcular M requiere un tiempo que está en $O\left(\frac{4^n}{n}\right)$.
- Este método no resulta práctico para valores grandes de n: hay demasiadas maneras de insertar paréntesis para examinarlas todas.

ALGORITMO DE PROGRAMACIÓN DINÁMICA

- Construimos una tabla m_{ij}, 1 ≤ i ≤ j ≤ n, en donde m_{ij} da la solución óptima, es decir, el número de multiplicaciones necesarias para la parte $M_i M_{i+1}$…M_{ij} del producto solicitado. La solución del problema original viene dada por m_{1n}.
- Supongamos que las dimensiones de la matriz están dadas por un vector d[0..n] tal que la matriz M_i con 1 ≤ i ≤ n, es de dimensión d_{i-1}xd_i .
- Construimos la tabla m_{ij} diagonal por diagonal: la diagonal s contiene los elementos m_{ij} tales que j - i = s. Por tanto la diagonal s=0 contiene los elementos m_{ii} con 1 ≤ i ≤ n, correspondientes a los productos M_i . Aquí m_{ii} = 0 para todo i.

- La diagonal s = 1 contiene los elementos m_{ii+1}correspondientes a productos de la forma $M_i M_{i+1}$. Calculamos directamente el producto efectuando $d_{i-1}d_i d_{i+1}$ multiplicaciones escalares.
- Finalmente cuando s>1, la diagonal s contendrá m_{ii+s} correspondientes a productos de la forma $M_i M_{i+1} \ldots M_{i+s}$.
- Si hacemos el corte después de M_k, con i ≤ k < i + s, entonces se necesitan m_{ik} multiplicaciones escalares para calcular el término de la izquierda, m_{k+1i+s} para calcular el término de la derecha, y por último $d_{i-1}d_k d_{i+s}$ para multiplicar las dos matrices resultantes con objeto de obtener el resultado final. Para hallar el resultado óptimo, tenemos que hallar el corte que devuelva menos operaciones.

$$m_{ij} = \begin{cases} 0, & \text{si} \quad i = j \\ \min\limits_{i \leq k < j} \{m_{ik} + m_{k+1,j} + d_{i-1} d_k d_j\}, & \text{si} \quad i < j \end{cases}$$

- Usando el ejemplo anterior pero aplicando el algoritmo dinámico:
- Rellenamos el vector d[5] = (13, 5, 89, 3, 34).
- El primer paso es rellenar la diagonal principal de la matriz, i = j y s = 0, con 0.
- Para s = 1, m_{12} = 5.785, m_{23} = 1.335 y m_{34} = 9.078.
- Para s = 2, m_{13} = min ($m_{11} + m_{23}$ + 13 x 5 x 3, $m_{12} + m_{33}$ + 13 x 89 x 3)

  = min (1.530, 9.256) = 1.530

  m_{24} = min ($m_{22} + m_{34}$ + 5 x 89 x 34, $m_{23} + m_{44}$ + 5 x 3 x 34)

  = min (24.208, 1.845) = 1.845

- Para el último paso s = 3, o lo que es lo mismo n – 1:
- m_{14} = min ($m_{11} + m_{24}$ + 13 x 5 x 34,

  $m_{12} + m_{34}$ + 13 x 89 x 34,

  $m_{13} + m_{44}$ + 13 x 3 x 34) = min (4.055, 54.201, 2.856) = 2.856

	j =1	2	3	4	
i =1	0	5.785	1.530	2.856	s = 3
2		0	1.335	1.845	s = 2
3			0	9.078	s = 1
4				0	s = 0

- Para s > 0 hay que calcular n-s elementos en la diagonal s; para cada uno de ellos, necesitamos decidirnos entre s posibilidades dadas por los distintos valores de k. Por tanto el tiempo de ejecución del algoritmo está en el orden exacto de:

$$\sum_{s=1}^{n}(n-s)s = n\sum_{s=1}^{n-1} s - \sum_{s=1}^{n-1} s^2$$
$$= n^2(n-1)/2 - n(n-1)(2n-1)/6$$
$$= (n^3 - n)/6$$

- Por lo tanto, T(n) = $O(n^3)$

```
fun minProd(dimMatriz[1..n]: ent,matrizSol[1..n,1..n]: ent,i,j: ent) dev minP:ent
  var
    aux: ent;
  fvar
  minP=+infinito;
  para k=i hasta j-1 hacer
    aux=matrizSol[i,k]+matrizSol[k+1,j]+(dimMatriz[i-1]*dimMatriz[k]*dimMatriz[j]);
    si (aux<minP) entonces
      minP=aux;              //Actualiza si el número de multiplicaciones es menor
    fsi
  fpara
  dev minP;
ffun
```

```
fun Mult_Mat_PD(dimMatriz[0..n]: ent) dev sol:ent  //Donde n es la dimension de la matriz solución
  var                                                es decir, el número de matrices a solucionar
    matrizSol[1..n,1..n]: ent;                     //Matriz solución
  fvar
  para i=1 hasta n hacer
    matrizSol[i][i]=0;                             //Inicializa la diagonal principal a 0
  fpara
  para diagonal=1 hasta n-1 hacer
    para i=0 hasta n-diagonal hacer
        matrizSol[i,i+diagonal]=minProd(dimMatriz,matrizSol,i,i+diagonal);
            //En cada casilla de cada diagonal escribe el su
              correspondiente mínimo de multiplicaciones necesaria
    fpara
  fpara
  dev matrizSol[0,n]);
ffun
```

Implementación:

```
// Suponiendo que hemos calculado previamente s[i,j]…

MultiplicaCadenaMatrices (A, i, j)
{
  if (j>i) {
     x = MultplicaCadenaMatrices (A, i, s[i,j]);
     y = MultplicaCadenaMatrices (A, s[i,j]+1, j);
     return MultiplicaMatrices(x, y);
  } else {
     return A[i];
  }
}
```

GRACIAS POR VUESTRA ATENCIÓN

- Imar Abreu Díaz
- Carlos García González
- Richard Morales Luis