MULTIPLICACIÓN ENCADENADA DE MATRICES

ALBERTO VERDEJO

UNIVERSIDAD COMPLUTENSE MADRID

Multiplicación de matrices

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 & 9 & 1 \\ 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 29 & 35 & 41 & 38 \\ 74 & 89 & 104 & 83 \end{bmatrix}$$

- El producto de una matriz $A_{p \times q}$ y una matriz $B_{q \times r}$ es una matriz $C_{p \times r}$ cuyos elementos son

$$c_{ij} = \sum_{k=1}^{q} a_{ik} b_{kj}$$

- Se necesitan pqr multiplicaciones entre números para calcular C.

Multiplicación encadenada de matrices

- Queremos multiplicar una secuencia de matrices $M_1 M_2 \cdots M_n$ donde M_i tiene dimensión $d_{i-1} \times d_i$.
- El producto de matrices no es conmutativo, pero sí asociativo.
- ¿Cuál es la mejor forma de colocar paréntesis en la secuencia de multiplicaciones de forma que el número de multiplicaciones entre números sea *mínimo*?

Ejemplo

$$A_{13\times 5} \cdot B_{5\times 89} \cdot C_{89\times 3} \cdot D_{3\times 34}$$

$$\underbrace{\underbrace{\underbrace{((A \cdot B)}_{5785} \cdot C)}_{3471} \cdot D}_{1326} \quad \rightsquigarrow \quad 10582$$

$$\underbrace{\underbrace{(A \cdot \underbrace{(B \cdot C)}_{1335})}_{195} \cdot D}_{1326} \quad \rightsquigarrow \quad 2856$$

Multiplicación encadenada de matrices

$$\underbrace{(M_1 \cdot \ldots \cdot M_k)}_{d_0 \times d_k} \cdot \underbrace{(M_{k+1} \cdot \ldots \cdot M_n)}_{d_k \times d_n}$$

$matrices(i,j)$ = número *mínimo* de multiplicaciones básicas para realizar el producto matricial $M_i \cdots M_j$

Definición recursiva

$$\underbrace{(M_i \cdot \ldots \cdot M_k)}_{d_{i-1} \times d_k} \cdot \underbrace{(M_{k+1} \cdot \ldots \cdot M_j)}_{d_k \times d_j}$$

- Todas las posibilidades de decidir cuál es el último producto:

$$M_i \cdot (M_{i+1} \cdot \ldots \cdot M_j)$$

$$(M_i \cdot M_{i+1}) \cdot (M_{i+2} \cdot \ldots \cdot M_j)$$

$$\ldots$$

$$(M_i \cdot \ldots \cdot M_{j-1}) \cdot M_j$$

Definición recursiva

- Casos recursivos, $i < j$.

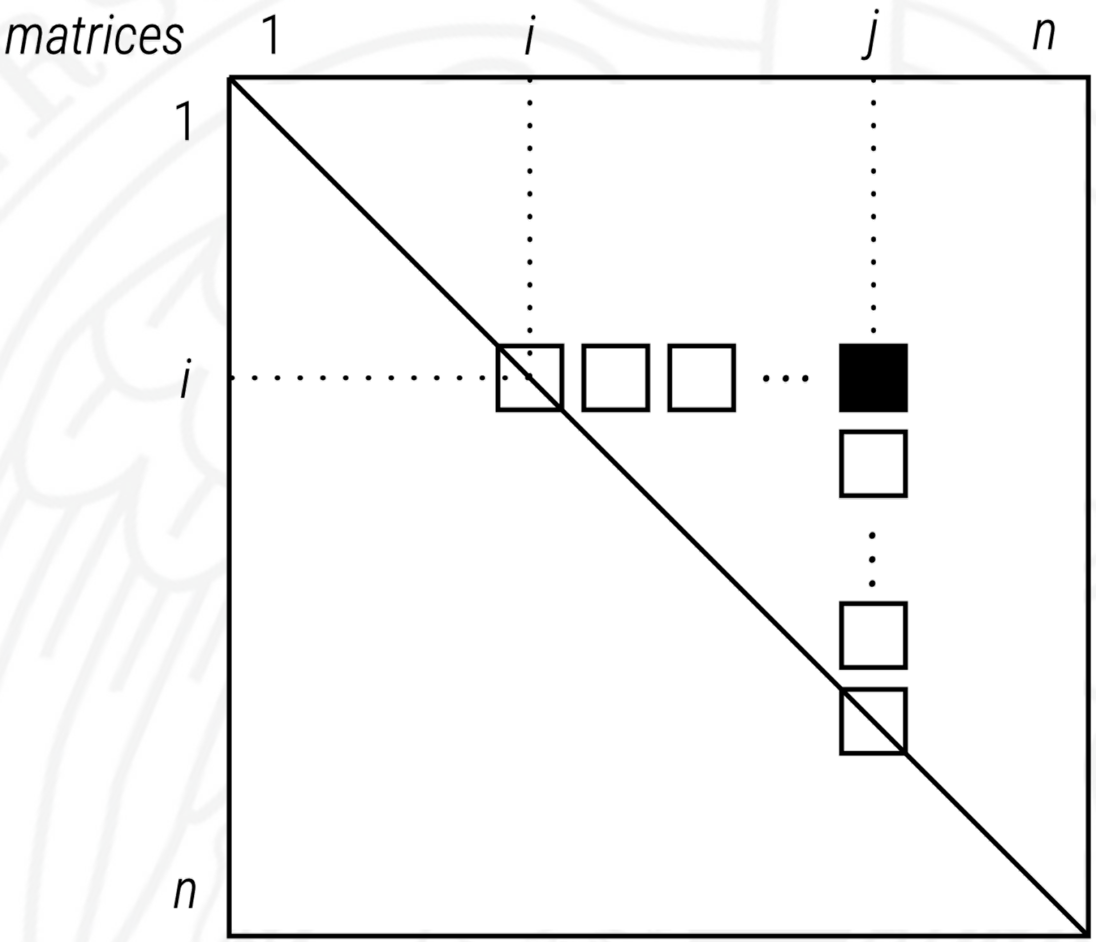
$$matrices(i,j) = \text{mín}_{i \leq k \leq j-1} \{matrices(i,k) + matrices(k+1,j) + d_{i-1} d_k d_j\}$$

- Casos básicos:

$$matrices(i,i) = 0$$

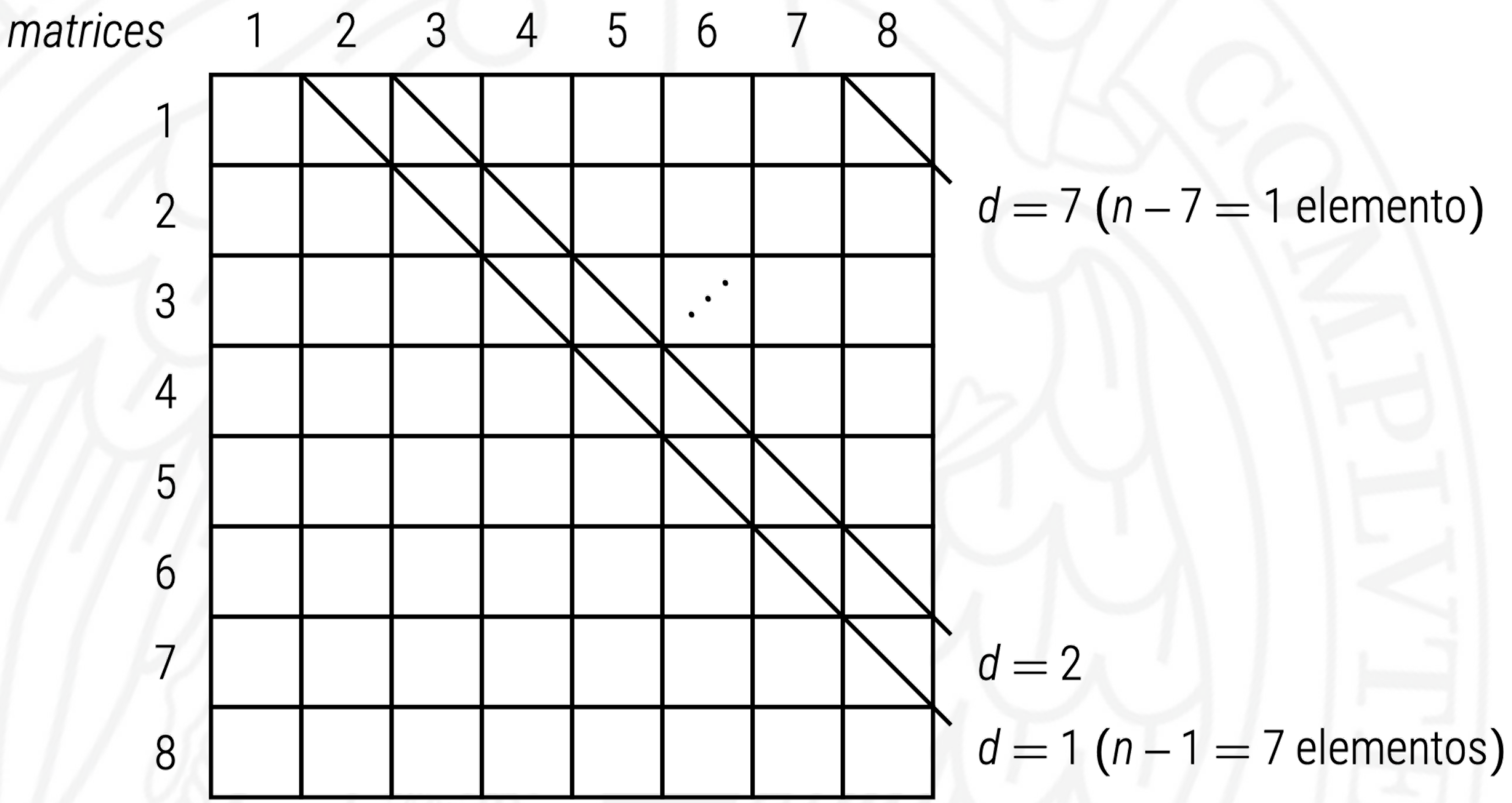
- Llamada inicial: $matrices(1, n)$

Tabla

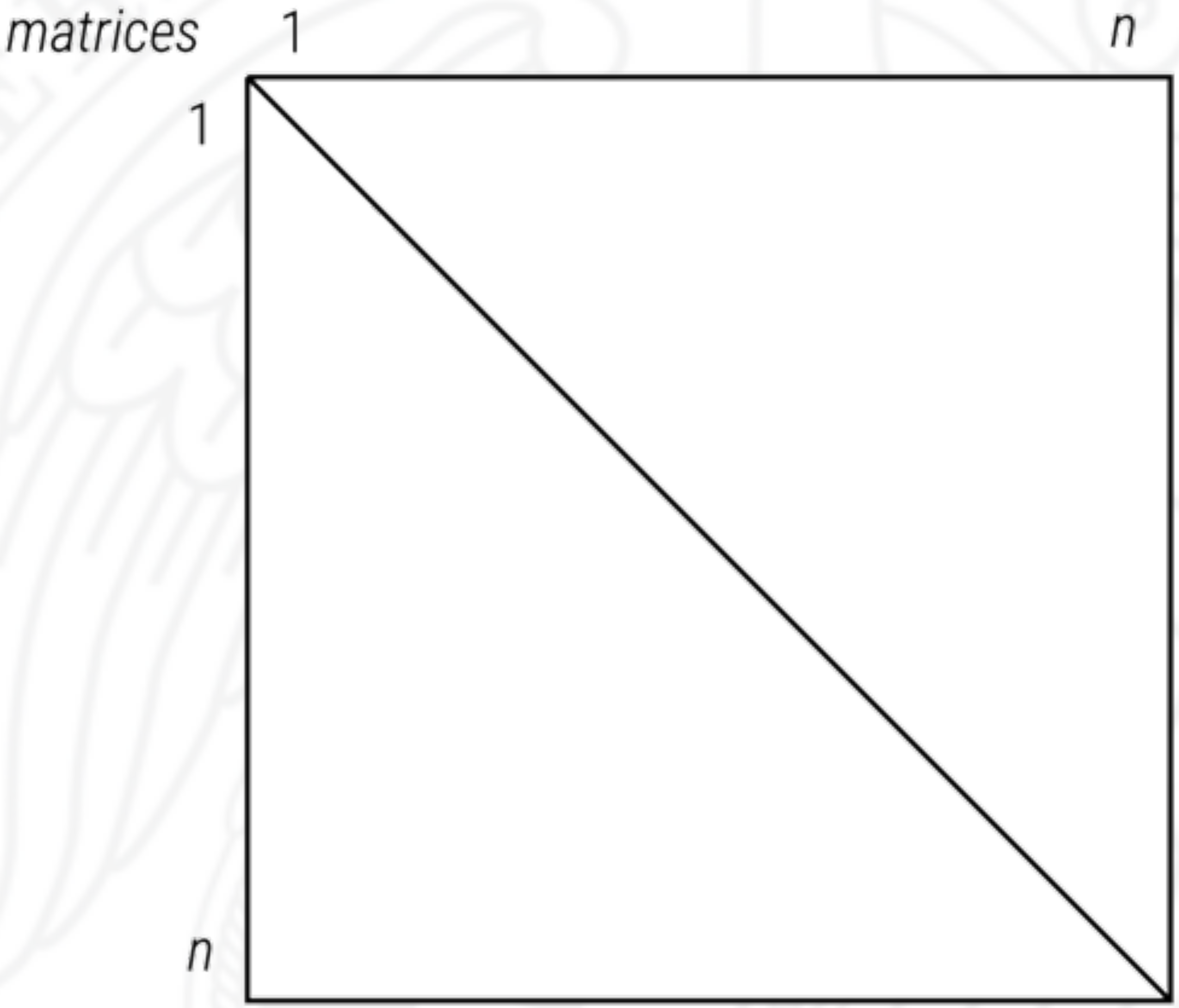


$$matrices(i,j) = \min_{i \leq k \leq j-1} \{matrices(i,k) + matrices(k+1,j) + d_{i-1} d_k d_j\}$$

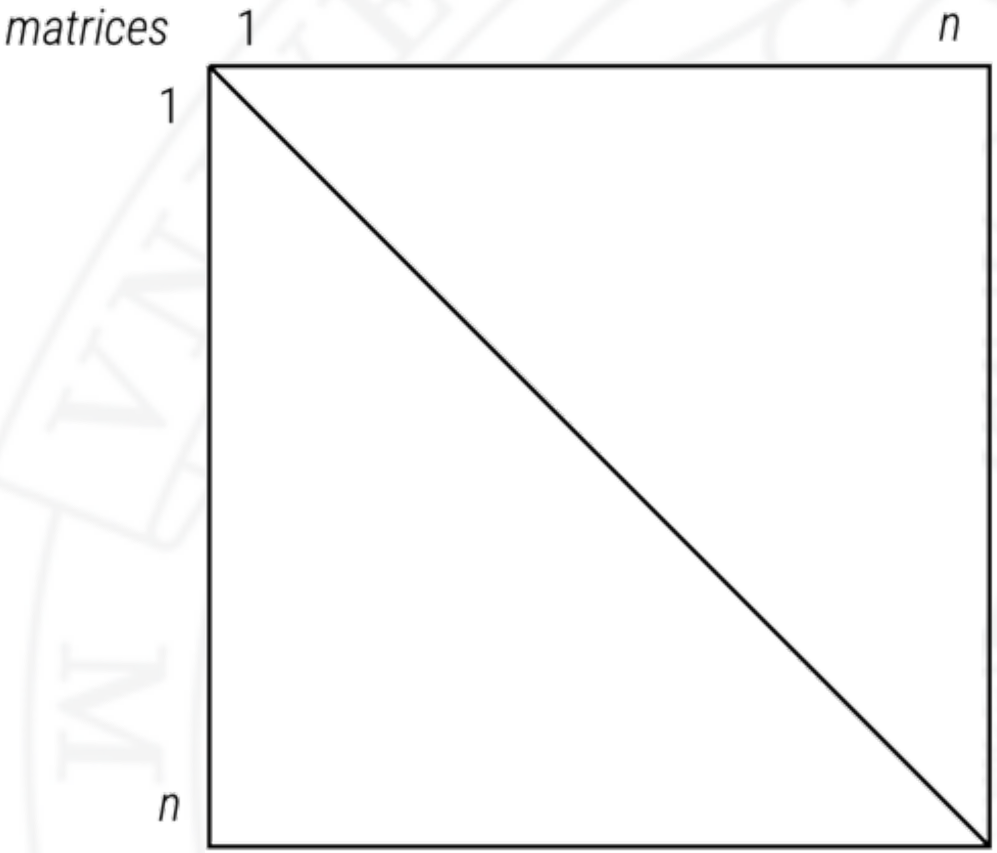
Recorrido por diagonales

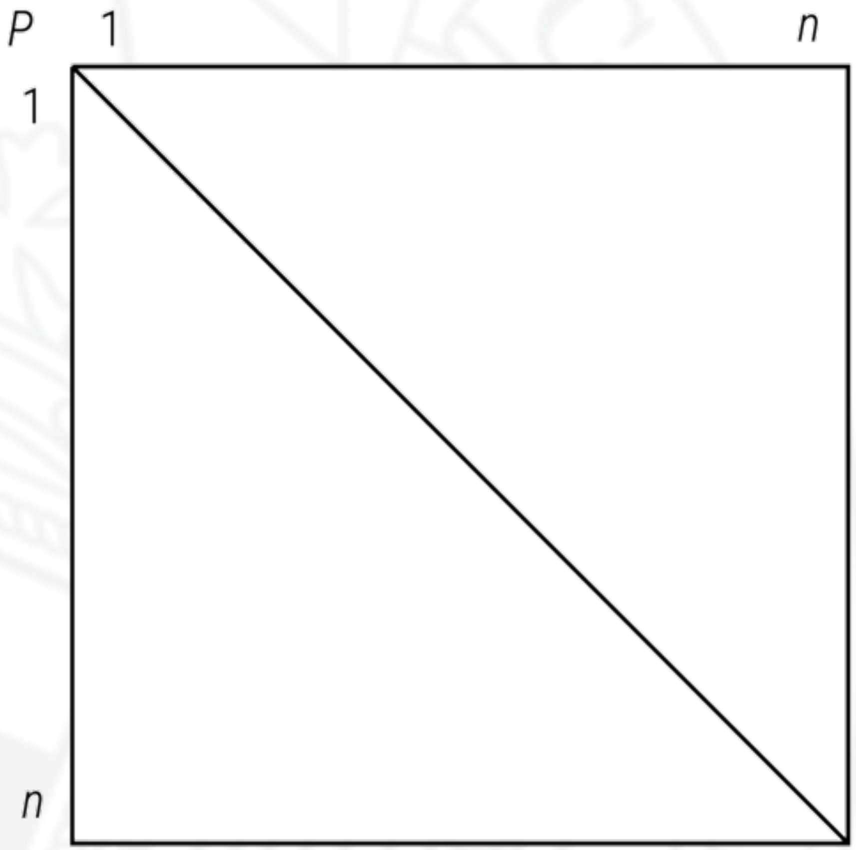


Reconstrucción de la solución



Reconstrucción de la solución





Implementación

```
int multiplica_matrices(vector<int> const& D, Matriz<int> & P) {
  int n = D.size() - 1;
  Matriz<int> matrices(n+1,n+1,0); P = Matriz<int>(n+1,n+1,0);
  for (int d = 1; d <= n-1; ++d)  // recorre diagonales
     for (int i = 1; i <= n - d; ++i) {  // recorre elementos de diagonal
        int j = i + d;
        matrices[i][j] = INF;
        for (int k = i; k <= j-1; ++k) {
           int temp = matrices[i][k] + matrices[k+1][j] + D[i-1]*D[k]*D[j];
           if (temp < matrices[i][j]) { // es mejor partir por k
              matrices[i][j] = temp;  P[i][j] = k;
           }
        }
     }
  return matrices[1][n];
}
```

$$\sum_{d=1}^{n-1}\sum_{i=1}^{n-d} d = \sum_{d=1}^{n-1}(n-d)d = \frac{n(n-1)(n+1)}{6} \in O(n^3)$$

Implementación

```
void escribir_paren(int i, int j, Matriz<int> const& P) {
    if (i == j)
        cout << "M" << i;
    else {
        int k = P[i][j];
        if (k > i) {
            cout << "("; escribir_paren(i, k, P); cout << ")";
        } else cout << "M" << i;
        cout << "*";
        if (k+1 < j) {
            cout << "("; escribir_paren(k+1, j, P); cout << ")";
        } else cout << "M" << j;
    }
}
```