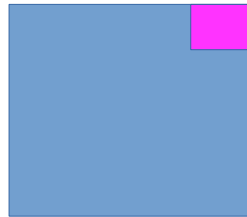


## TD02 : calculs distribués : Multiplication des matrices

$$\mathbf{R} = \mathbf{P} * \mathbf{Q}$$

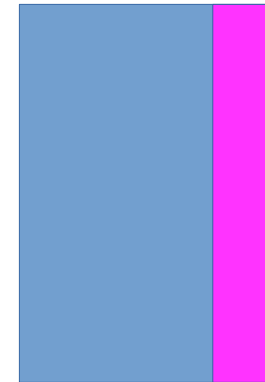
(m,p)



(m,n)



(n,p)



$$\forall i, j : r_{i,j} = p_{i,1} * q_{1,j} + p_{i,2} * q_{2,j} + \dots + p_{i,n} * q_{n,j} = \sum_{k=1}^n p_{i,k} * q_{k,j}$$

Complexité :  $m * n * p$  multiplications

## TD calculs distribués : Multiplication des matrices

### Calcul de base

```
final public class Matrix {  
    private final int M;           // number of rows  
    private final int N;           // number of columns  
    ...  
    public Matrix times(Matrix Q) { // return R = this * Q  
        Matrix.complexity = 0;  
        Matrix P = this;  
        if (P.N != Q.M) throw new RuntimeException("Illegal matrix dimensions.");  
        Matrix R = new Matrix(P.M, Q.N);  
        for (int i = 0; i < R.M; i++)  
            for (int j = 0; j < R.N; j++)  
                for (int k = 0; k < P.N; k++) {  
                    R.data[i][j] += P.data[i][k] * Q.data[k][j];  
                    Matrix.complexity++; // naturellement, on sait que c'est C.M*C.N*A.N  
                }  
        return R;  
    }  
}
```

## TD calculs distribués : Multiplication des matrices

$$R = P * Q$$

*Avec un découpage des opérandes*

$A * A' + B * C'$	$A * B' + B * D'$
$C * A' + D * C'$	$C * B' + D * D'$

A	B
C	D

A'	B'
C'	D'

Il y a 8 threads (un thread pour chaque multiplication)

Complexité :  $m/2 * n/2 * p/2$  multiplications

## TD calculs distribués : Multiplication des matrices

# Organisation des calculs

*Avec un découpage des opérandes*

Une opérande est une **sous-matrice** :

$$X = (Y, l_{\min}, c_{\min}, nbl, nbc)$$

P (M,N)

A	B
C	D

Par exemple :

$$A = (P, 0, 0, P.M/2, P.N/2)$$

$$B = (P, 0, P.N/2, P.M/2, P.N/2)$$

$$C = (P, P.M/2, 0, P.M/2, P.N/2)$$

$$D = (P, P.M/2, P.N/2, P.M/2, P.N/2)$$