# Paradigma de Divisão-e-Conquista

Problema: Multiplicação de Matrizes Quadradas
e Algoritmo de Strassen

# Multiplicação de Matrizes

- Dadas duas matrizes quadradas nxn, $A=(a_{ij})$ e $B=(b_{ij})$, então no produto $C = A \times B$, definimos as entradas cij, para i,j = 1,2, ..., n, por

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

# Multiplicação de Matrizes

```
algoritmo square_matrix_multiply(A, B)
    n=A.linhas
    declare C uma matriz nxn
    para i = 1 até n
        para j = 1 até n
            C[i,j] = 0
            para k = 1 até n
                C[i,j] = C[i,j] + A[i,k] * B[k,j]
retorne(C)
```

# Multiplicação de Matrizes

- Um algoritmo divisão e conquista simples.
  - Sejam as matrizes quadradas A e B, 2x2.

$$A = \begin{matrix} A11 & A12 \\ A21 & A22 \end{matrix} \qquad B = \begin{matrix} B11 & B12 \\ B21 & B22 \end{matrix} \qquad C = \begin{matrix} C11 & C12 \\ C12 & C22 \end{matrix} \qquad (1)$$

Quais são as equações que calculam C11, C12, C21, C22 são:

C11 = A11*B11 + A21*B12 (2)

C12 = A11*B12 + A12*B22 (3)

C21 = A21*B11 + A22*B21 (4)

C22 = A21*B12 + A22*B22 (5)

# Multiplicação de Matrizes

Cada uma das 4 equações anteriores determina a multiplicação de matrizes (n/2 x n/2) e a adição de seus n/2 x n/2 produtos.

Podemos usar essas equações para criar um algoritmo de divisão e conquista simples e recursivo.

# Multiplicação de Matrizes

```
algoritmo Square-Matrix-Multiply-Recursive(A,B)
    n=A.rows
    seja C uma nova matriz nXn.
    se n==1 então
        c11 = a11*b11
    senão
        particione as matrizes como a equação (1)
        C11 = Square-Matrix-Multiply-Recursive(A11,B11) +
              Square-Matrix-Multiply-Recursive(A12,B21)
        C12 = Square-Matrix-Multiply-Recursive(A11,B12) +
              Square-Matrix-Multiply-Recursive(A12,B22)
        C21 = Square-Matrix-Multiply-Recursive(A21,B11) +
              Square-Matrix-Multiply-Recursive(A22,B21)
        C22 = Square-Matrix-Multiply-Recursive(A21,B12) +
              Square-Matrix-Multiply-Recursive(A22,B22)
retorne(C)
```

# Multiplicação de Matrizes

Esse pseudocódigo encobre um detalhe de implementação sutil, mas importante.

Como particionamos as matrizes na linha 5? Se fôssemos criar 12 novas matrizes n/2 x n/2, gastaríamos $\Theta(n^2)$ tempo copiando entradas. Na verdade, podemos particionar as matrizes sem copiar entradas.

Não é preciso copiar as entradas, o truque é usar cálculos de índice, identificar uma submatriz por um intervalo de índices de linha e um intervalo de índices de coluna da matriz original. Assim, acabamos representando uma submatriz um pouco diferente de como representamos a matriz original, que é a sutileza que estamos encobrindo.

# Multiplicação de Matrizes

A vantagem é que, como podemos especificar submatrizes por cálculos de índice, a execução da linha 5 leva apenas $\Theta(1)$ tempo (embora veremos que não faz diferença assintoticamente no tempo de execução geral se copiamos ou particionamos no local).

# Método Strassen

O método de Strassen não é nada óbvio. Ele tem quatro etapas:

1. Divida as matrizes de entrada A e B e a matriz de saída C em n/2 x n/2 submatrizes, como na equação (1). Este passo leva $\Theta(1)$ tempo pelo cálculo do índice, assim como em SQUARE-M ATRIX-MULTIPLY-RECURSIVE .

2. Crie 10 matrizes S1,S2,...,S10,cada uma das quais é n/2xn/2 e é a soma ou diferença de duas matrizes criadas na etapa 1. Podemos criar todas as 10 matrizes em $\Theta(n^2)$ tempo.

# Método Strassen

3. Usando as submatrizes criadas na etapa 1 e as 10 matrizes criadas na etapa 2, calcule recursivamente sete produtos de matriz P1, , P2 , .... ,P7 . Cada matriz Pi é n/2 x n/2.

4. Calcule as submatrizes desejadas C11, C12, C21, C22 da matriz de resultado C adicionando e subtraindo várias combinações das matrizes Pi. Podemos calcular todas as quatro submatrizes em tempo $\Theta(n^2)$.

# Método de Strassen

- Detalhes do passo 2: são criadas as seguintes 10 matrizes auxiliares:

S1 = B12 – B22
S2 = A11 + A12
S3 = A21 + A22
S4 = B21 – B11
S5 = A11 + A22
S6 = B11 + B22
S7 = A12 – A22
S8 = B21 + B22
S9 = A11 – A21
S10 = B11 + B12

aqui subtraímos ou adicionamos matrizes 10 vezes, este passo realmente leva $\Theta(n^2)$.

# Método de Strassen

- Detalhes passo 3: recursivamente multiplicamos 7 vezes matrizes n/2 x n/2 , onde cada uma é a soma ou a diferença dos produtos das submatrizes A e B:

| | |
|---|---|
| P1 = A11 * S1 | = A11*B12 – A11*B22 |
| P2 = S2 * B22 | = A11*B22 + A12*B22 |
| P3 = S3 * B11 | = A21*B11 + A22*B11 |
| P4 = A22 * S4 | = A22*B21 – A22*B11 |
| P5 = S5 * S6 | = A11*B11 + A11*B22 +A22*B11 + A22*B22 |
| P6 = S7 * S8 | = A12*B21 + A12*B22 – A22*B21 – A22*B22 |
| P7 = S9 * S10 | = A11*B11 + A11*B12 – A21*B11 - A21*B12 |

# Método de Strassen

- Detalhes passo 4: adicionamos e subtraímos as Pi matrizes criadas no passo 3 para construir as 4 submatrizes n/2 x n/2 o produto:

$$C11 = P5 + P4 - P2 + P6$$
$$C12 = P1 + P2$$
$$C21 = P3 + P4$$
$$C22 = P5 + P1 - P3 - P7$$

  - Custo de $\Theta(n^2)$.

# Método Strassen

- Recorrência:

$$T(n) = 1 \quad \text{se } n = 1; \quad \text{e}$$

$$T(n) = \Theta(1) + \Theta(n^2) + 7T(n/2) + \Theta(n^2) \quad \text{se } n > 1$$

$$= 7T(n/2) + \Theta(n^2) \quad \text{se } n > 1$$

Logo $T(n) = \Theta(n^{\log 7}) = \Theta(n^{2,81})$