Föreläsning 8 i ADK

Algoritmkonstruktion: dekomposition

Stefan Nilsson

KTH

Metod 3 - Dekomposition
(Divide and conquer, Söndra och härska)

- Dela upp i mindre problem
- Lös delproblemen rekursivt
- Kombinera resultaten

Analys: Använd en rekursionsrelation

Exempel: Matrismultiplikation

Metod 3 - Dekomposition (Divide and conquer, Söndra och härska)

Matrismultiplikation

$$C = AB \Leftrightarrow \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

function $\textsc{Mul}(A,B,n)$
- **if** $n = 1$ **then return** $A \cdot B$
- $C_{11} = \textsc{Add}(\ \textsc{Mul}(A_{11}, B_{11}, \frac{n}{2}), \textsc{Mul}(A_{12}, B_{21}, \frac{n}{2})\ )$
- $C_{12} = \textsc{Add}(\ \textsc{Mul}(A_{11}, B_{12}, \frac{n}{2}), \textsc{Mul}(A_{12}, B_{22}, \frac{n}{2})\ )$
- $C_{21} = \textsc{Add}(\ \textsc{Mul}(A_{21}, B_{11}, \frac{n}{2}), \textsc{Mul}(A_{22}, B_{21}, \frac{n}{2})\ )$
- $C_{22} = \textsc{Add}(\ \textsc{Mul}(A_{21}, B_{12}, \frac{n}{2}), \textsc{Mul}(A_{22}, B_{22}, \frac{n}{2})\ )$
- **return** C

Fungerar om $n = 2^k$ för något k

Analys: $$\left.\begin{aligned} T(1) &= 1 \\ T(n) &= 8T\left(\frac{n}{2}\right) + 4\left(\frac{n}{2}\right)^2 \end{aligned}\right\} \Rightarrow T(n) = \mathcal{O}(n^3)$$

Snabb multiplikation av 2×2-matriser (strassen)

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

Kan beräknas genom:

$$\begin{aligned}
m_1 &= (a_{12} - a_{22}) \cdot (b_{21} + b_{22}) \\
m_2 &= (a_{11} + a_{22}) \cdot (b_{11} + b_{22}) \\
m_3 &= (a_{11} - a_{21}) \cdot (b_{11} + b_{12}) \\
m_4 &= (a_{11} + a_{12}) \cdot b_{22} \\
m_5 &= a_{11} \cdot (b_{12} + b_{22}) \\
m_6 &= a_{22} \cdot (b_{21} + b_{11}) \\
m_7 &= (a_{21} + a_{22}) \cdot b_{11}
\end{aligned}$$

$$\begin{aligned}
c_{11} &= m_1 + m_2 - m_4 + m_6 \\
c_{12} &= m_4 + m_5 \\
c_{21} &= m_6 + m_7 \\
c_{22} &= m_2 - m_3 + m_8 - m_7
\end{aligned}$$

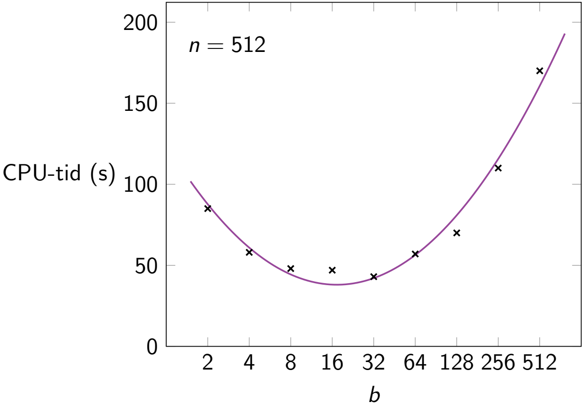
Sammanlagt 7 multiplikationer samt 18 additioner och subtraktioner

Multiplikation av två $n \times n$-matriser tar tid:

$$\left.\begin{aligned} T(1) &= 1 \\ T(n) &= 7T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2 \end{aligned}\right\} \Rightarrow T(n) = \mathcal{O}(n^{\log_2 7}) \sim O(n^{2.81})$$

Strassen i praktiken

- Använd Strassens multiplikationsalgoritm för stora matriser ($n > b$)
- Använd vanlig matrismultiplikation för mindre matriser ($n \leqslant b$)
- Hur ska brytpunkten b väljas?



Multiplikation av binära tal

- $x = \underbrace{x_{n-1}x_{n-2}\cdots x_{\frac{n}{2}}}_{a}\underbrace{x_{\frac{n}{2}-1}\cdots x_1x_0}_{b} = a \cdot 2^{\frac{n}{2}} + b$
- $y = \underbrace{y_{n-1}y_{n-2}\cdots y_{\frac{n}{2}}}_{c}\underbrace{y_{\frac{n}{2}-1}\cdots y_1y_0}_{d} = c \cdot 2^{\frac{n}{2}} + d$
- $xy = ac \cdot 2^n + (ad + bc)2^{\frac{n}{2}} + bd$
- Anta att $n = 2^k$

Multiplikation av binära tal

function MULT(x,y,k)
  if k = 0 **then return** x·y
  else
    [a,b] ← x
    [c,d] ← y
    p_1 ← MULT(a,c,k-1)
    p_2 ← MULT(b,d,k-1)
    p_3 ← MULT(a,d,k-1)
    p_4 ← MULT(b,c,k-1)
    return $p_1 \cdot 2^n + (p_3 + p_4)2^{\frac{n}{2}} + p_2$

$$\left.\begin{array}{l} T(1) = \Theta(1) \\ T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n) \end{array}\right\} \Rightarrow T(n) = \mathcal{O}(n^{\log_2 4}) = \mathcal{O}(n^2)$$

Smartare multiplikation (Karatsuba)

- $A = a \cdot c$
- $B = b \cdot d$
- $C = (a + b) \cdot (c + d)$
- $D = A \cdot 2^n + (C - A - B) \cdot 2^{\frac{n}{2}} + B$
- $D = ac \cdot 2^n + (ad + bc)2^{\frac{n}{2}} + bd = x \cdot y$

Smartare multiplikation (Karatsuba)

```
function SMARTMULT(x,y,k)
    if k ⩽ 4 then return x·y
    else
        [a,b] ← x
        [c,d] ← y
        A ← SMARTMULT(a,c,k-1)
        B ← SMARTMULT(b,d,k-1)
        C ← SMARTMULT(a+b,c+d,k-1)
        return A · 2^n + (C − A − B)2^(n/2) + B
```

$$\left.\begin{aligned} T(1) &= \Theta(1) \\ T(n) &= 3T\left(\tfrac{n}{2}\right) + \Theta(n) \end{aligned}\right\} \Rightarrow T(n) = \mathcal{O}(n^{\log_2 3}) = \mathcal{O}(n^{1{,}58})$$

(Bästa kända algoritmen: $\mathcal{O}(n \log n)$)