Grundlæggende Algoritmer og Datastrukturer

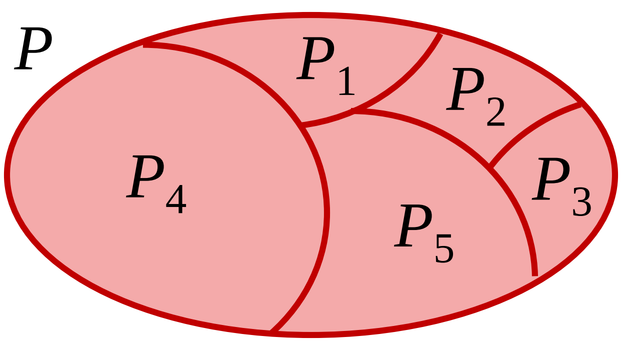
Del-og-kombiner
[CLRS, kapitel 2.3, 4.2-4.5, problem 30.1.c]

Del-og-Kombiner

Algoritme design teknik

Virker for mange problemer (men langt fra alle)

- **Opdel** et problem P i mindre problemer $P_1,..,P_k$, der kan løses uafhængigt
  (små problemer løses direkte)
- Løs delproblemerne $P_1,..,P_k$ **rekursivt**
- **Kombiner** løsningerne for $P_1,..,P_k$ til en løsning for P



Eksempel: Merge-Sort

$\textsc{Merge-Sort}(A, p, r)$

if $p < r$

$\quad q = \lfloor (p + r)/2 \rfloor$

$\quad \textsc{Merge-Sort}(A, p, q)$

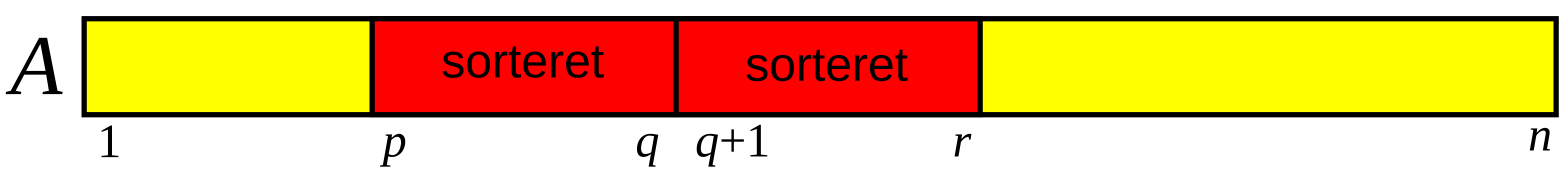
$\quad \textsc{Merge-Sort}(A, q + 1, r)$

$\quad \textsc{Merge}(A, p, q, r)$

① To mindre delproblemer

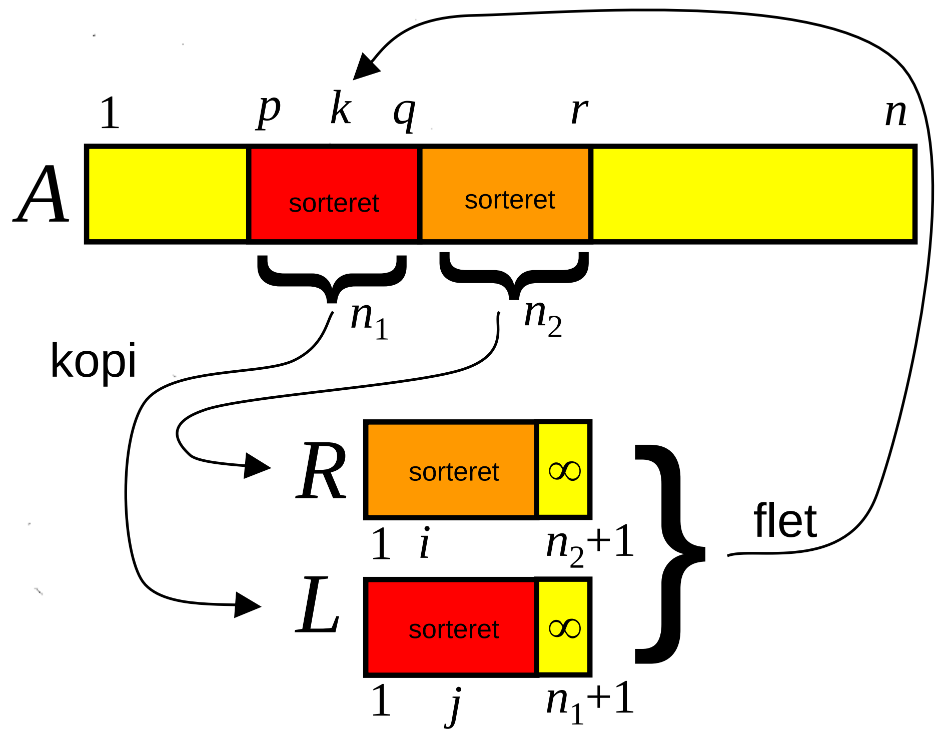
② Løs rekursivt

③ Kombiner



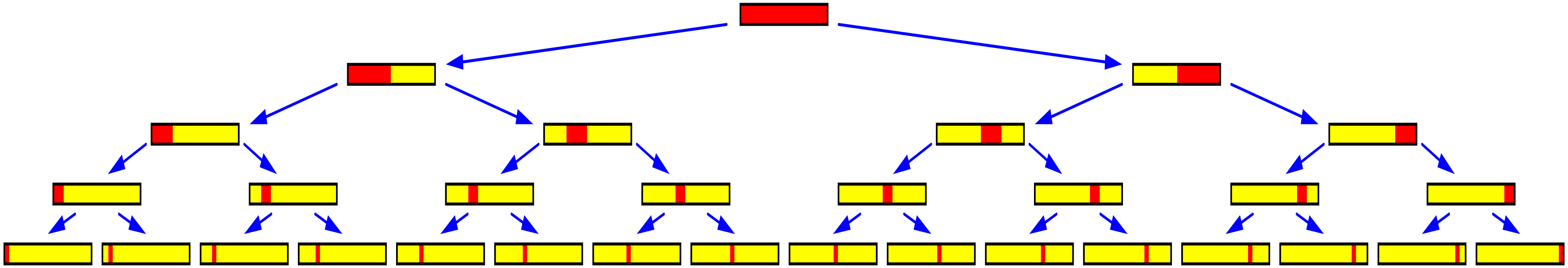
MERGE(A, p, q, r)

$n_1 = q - p + 1$
$n_2 = r - q$
let $L[1\,..\,n_1 + 1]$ and $R[1\,..\,n_2 + 1]$ be new arrays
for $i = 1$ **to** n_1
    $L[i] = A[p + i - 1]$
for $j = 1$ **to** n_2
    $R[j] = A[q + j]$
$L[n_1 + 1] = \infty$
$R[n_2 + 1] = \infty$
$i = 1$
$j = 1$
for $k = p$ **to** r
    if $L[i] \leq R[j]$
        $A[k] = L[i]$
        $i = i + 1$
    else $A[k] = R[j]$
        $j = j + 1$



Merge-Sort : Analyse

Rekursionstræet



Observation

Samlet arbejde per lag er $O(n)$

Arbejde

$$O(n \cdot \#\ \text{lag}) = O(n \cdot \log_2 n)$$

Del-og-kombiner, FADS eksempler:

- **MergeSort**
  - Del op i to lige store dele
  - Rekursiv sortering
  - Kombiner = fletning
- **QuickSort**
  - Opdel efter tilfældigt pivot (**tilfældig opdeling**)
  - Rekursiv sortering
  - Kombiner = ingen (konkatener venstre og højre)
- **QuickSelect**
  - Opdel efter tilfældigt pivot (**tilfældig opdeling**)
  - Rekursiv select
  - Kombiner = ingen

Analyse af Del-og-Kombiner

= analyse af en rekursiv procedure

Essentielt to forskellige måder:

1. Argumenter direkte om **rekursionstræet** (analyser dybde, #knuder på hvert niveau, arbejde i knuderne/niveauerne/træet)
2. Løs en matematisk **rekursionsligning**, f.eks.

$$
\begin{cases}
T(n) \leq a & \text{hvis} \quad n \leq c \\
T(n) \leq 2 \cdot T(n/2) + a \cdot n & \text{ellers}
\end{cases}
$$

Bevises f.eks. vha. induktion.

Hvad er rekursionsformlen for MergeSort hvor man sorterer rekursivt 3 dele af størrelse n/3?

For $n \geq 3$ …

a) $T(n) \leq 3 \cdot T(n) + a \cdot n^3$

b) $T(n) \leq 3 \cdot T(n) + a \cdot n$

c) $T(n) \leq 3 \cdot T(n/3) + a \cdot n$

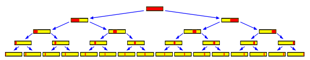
d) $T(n) \leq 3 \cdot T(n/2) + a \cdot n$

e) Ved ikke

… og $T(N) \leq c$ for $n < 3$

Løsning af rekursionsligninger

- Fold rekursionsligningen ud og argumenter om **rekursionstræet**



- Gæt en løsning og vis den ved induktion efter voksende n

$$
\begin{array}{ll}
T(n) \leq a & \text{hvis} \quad n \leq c \\
T(n) \leq 2 \cdot T(n/2) + a \cdot n & \text{ellers}
\end{array}
$$

Rekursionsligninger: Faldgrubber

- Ulige opdelinger glemmes (n ulige, så er de rekursive kald typisk $\lfloor n/2 \rfloor$ og $\lceil n/2 \rceil$)

[CLRS, kapitel 4.6.2]

- Analyserer typiske kun for $n = 2^k$

- Brug aldrig O-udtryk i rekursionsformlen – brug konstanter ($\cancel{T(n)=O(n)+O(T(n/3))}$)

$$T(n) \leq c \cdot n + a \cdot T(n/3)$$

Master Theorem
(Simplificering af [CLRS, Theorem 4.1])

Theorem

Hvis a, b, c, d, p er konstanter som opfylder $a, c, p > 0$, $d \geq 1$, og $b > 1$, så har rekursionsligningen

$$T(n) = \begin{cases} c & \text{hvis} \quad n \leq d \\ a \cdot T(n/b) + c \cdot n^p & \text{hvis} \quad n > d \end{cases}$$

følgende løsning

$$\begin{array}{ccc} \Theta(n^p) & \text{hvis} & a < b^p \\ \Theta(n^p \log n) & \text{hvis} & a = b^p \\ \Theta(n^{\log_b a}) & \text{hvis} & a > b^p \end{array}$$

Løsning til rekursionsligningen?

$$T(n) \leq 4 \cdot T(n / 2) + n^3 \quad \text{for } n > 1$$
$$T(n) \leq c \quad \text{for } n = 1$$

a) $T(n) = O(n^3)$

b) $T(n) = O(n^3 \log n)$

c) $T(n) = O(n^{\log_2 4}) = O(n^2)$

d) Ved ikke

Løsning til rekursionsligningen?

$$T(n) \leq 4 \cdot T(n / 2) + n \quad \text{for } n > 1$$
$$T(n) \leq c \quad \text{for } n = 1$$

a) $T(n) = O(n)$

b) $T(n) = O(n \log n)$

c) $T(n) = O(n^{\log_2 4}) = O(n^2)$

d) Ved ikke

Løsning til rekursionsligningen?

$$T(n) \leq 5 \cdot T(n / 4) + n^2 \quad \text{for } n > 1$$
$$T(n) \leq c \quad \text{for } n = 1$$

a) $T(n) = O(n^2)$

b) $T(n) = O(n^2 \log n)$

c) $T(n) = O(n^{\log_4 5}) = O(n^{1.161})$

d) Ved ikke

Løsning til rekursionsligningen?

$$T(n) \le 9 \cdot T(n/3) + n^2 \quad \text{for } n > 1$$
$$T(n) \le c \quad \text{for } n = 1$$

a) $T(n) = O(n^2)$

b) $T(n) = O(n^2 \log n)$

c) $T(n) = O(n^{\log_3 9}) = O(n^2)$

d) Ved ikke

Løsning til rekursionsligningen?

$$T(n) \le 7 \cdot T(n/2) + c \cdot n^2 \quad \text{for } n \ge 2$$

$$T(n) \le c \quad \text{for } n = 1$$

a) $T(n) = O(n^2)$

b) $T(n) = O(n^2 \cdot \log n)$

c) $T(n) = O(n^{7/2})$

d) $T(n) = O(n^{\log_2 7}) = O(n^{2.81})$

e) $T(n) = O(n^7)$

f) Ved ikke

Dybden af rekursionen?

$$T(n) = \begin{cases} c & \text{hvis} \quad n \leq d \\ a \cdot T(n/b) + c \cdot n^p & \text{hvis} \quad n > d \end{cases}$$

a) $\log n$

b) $\log_b n$

c) $\log_b (n/d)$

d) $\log_d (n/b)$

e) Ved ikke

Dybde	$i = 0..\log_b(n/d) - 1$	$\log_b(n/d)$
# delproblemer	a^i	$a^{\log_b(n/d)}$
Størrelse af delproblemer	n/b^i	d
Tid per delproblem	$c \cdot (n/b^i)^p$	c
Tid per lag	$a^i \cdot c \cdot (n/b^i)^p$	$c \cdot a^{\log_b(n/d)}$

$$T(n) = \Theta\left(\underbrace{c \cdot a^{\log_b(n/d)}}_{\text{(bunden af rekursionen)}} + \sum_{i=0}^{\log_b(n/d)-1} \underbrace{a^i \cdot c \cdot (n/b^i)^p}_{\text{(lag } i = 0..\log_b(n/d) - 1\text{)}} \right)$$

$$= \Theta\left(c \cdot a^{\log_b n} + c \cdot n^p \cdot \sum_{i=0}^{\log_b n - 1} (a/b^p)^i \right)$$

$$\frac{(a/b^p)^{\log_b n} - 1}{a/b^p - 1} \quad \text{for} \quad a \neq b^p$$

$$= \Theta\left(n^{\log_b a} + n^p \cdot \begin{cases} 1 & \text{for } a < b^p \\ \log n & \text{for } a = b^p \\ (a/b^p)^{\log_b n} & \text{for } a > b^p \end{cases} \right) = \Theta\left(\begin{cases} n^p & \text{for } a < b^p \\ n^p \cdot \log n & \text{for } a = b^p \\ n^{\log_b a} & \text{for } a > b^p \end{cases} \right)$$

$$a^{\log_b n} = n^{\log_b a}$$

$$(b^p)^{\log_b n} = n^p$$

Multiplikation af lange heltal

[CLRS, problem 30.1.c] **Karatsuba 1960**

- I og J hver heltal med n bits
- Naive implementation kræver $O(n^2)$ bit operationer
- Lad $I = I_h \cdot 2^{n/2} + I_l$ og $J = J_h \cdot 2^{n/2} + J_l$
- $I \cdot J = I_h \cdot J_h \cdot 2^n + ((I_h - I_l) \cdot (J_l - J_h) + I_l \cdot J_l + I_h \cdot J_h) \cdot 2^{n/2} + I_l \cdot J_l$

$$\boxed{\begin{array}{ll} T(n) \leq 3 \cdot T(n/2) + c \cdot n & \text{for } n \geq 2 \\ T(n) \leq c & \text{for } n = 1 \end{array}}$$

- $T(n) = O(n^{\log_2 3}) = O(n^{1.58})$

Multiplikation af lange heltal

$$I = \boxed{I_h}\underbrace{\boxed{I_l}}^{n/2} \qquad J = \boxed{J_h}\boxed{J_l}$$

$$I \cdot J = I_h \cdot J_h \cdot 2^n + (I_h \cdot J_l + I_l \cdot J_h) \cdot 2^{n/2} + I_l \cdot J_l$$

a) T(n) ≤ 2·T(n/2) + a·n^2

b) T(n) ≤ 4·T(n/4) + a·n

c) T(n) ≤ 4·T(n/2) + a·n

d) T(n) ≤ 4·T(n/2) + a·n^2

e) Ved ikke

… og $T(N) \leq c$ for $n = 1$

Multiplikation af lange heltal

+4000 år	$O(n^2)$
Del-og-kombiner Karatsuba 1960	$O(n^{\log_2 3})$
Schönhage-Strassen, 1971	$O(n \cdot \log n \cdot \log\log n)$
Fürer, 2007 Harvey, Hoeven, 2018	$O(n \cdot \log n \cdot 2^{O(\log^* n)})$ $O(n \cdot \log n \cdot 2^{2 \cdot \log^* n})$

Ikke pensum: Karatsuba har skrevet en artikel om historien af multiplikation.

- Anatolii A. Karatsuba: *The Complexity of Computations*, Proceedings of the Steklov Institute of Mathematics, 1995, vol. 211, 169–183 (russisk original)

Matricer

$m = 3$
søjler / kolonner

repræsenterer
lineær transformation $\mathbb{R}^3 \longrightarrow \mathbb{R}^4$

$n = 4$
rækker

$$\begin{pmatrix} 3 & 2 & -1 \\ 5 & 0 & 4 \\ 7 & -2 & 0 \\ 1 & 3 & 2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3x_1 + 2x_2 - x_3 \\ 5x_1 + 4x_3 \\ 7x_1 - 2x_2 \\ x_1 + 3x_2 + 2x_3 \end{pmatrix}$$

$n \times m$ matrix

søjlevektor
$m \times 1$ matrix

søjlevektor
$n \times 1$ matrix

Regneregler

Matrix addition
($n \times m$ matricer)

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} + \begin{pmatrix} 5 & 6 \\ 4 & 1 \\ 3 & 3 \end{pmatrix} = \begin{pmatrix} 6 & 8 \\ 7 & 5 \\ 8 & 9 \end{pmatrix}$$

$$(A + B) + C = A + (B + C)$$
$$A + B = B + A$$

Matrix subtraktion
($n \times m$ matricer)

$$\begin{pmatrix} 6 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} - \begin{pmatrix} 2 & 6 \\ 4 & 1 \\ 3 & 3 \end{pmatrix} = \begin{pmatrix} 4 & -4 \\ -1 & 3 \\ 2 & 3 \end{pmatrix}$$

$$A - (B + C) = (A - B) - C$$

Multiplikation med
konstant

$$3\begin{pmatrix} 1 & 2 \\ -2 & 3 \\ 4 & -1 \end{pmatrix} = \begin{pmatrix} 3 & 6 \\ -6 & 9 \\ 12 & -3 \end{pmatrix}$$

$$c(dA) = (cd)A$$
$$c(A + B) = (cA) + (cB)$$

Matrix Multiplikation

= komposition af lineære transformationer

$$
\begin{array}{cc}
 & \begin{array}{cccc} \phantom{a_{11}} & & & \end{array} \\
i & \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}
\end{array}
\begin{array}{c}
\begin{array}{cccc} j & \phantom{b_{12}} & & \phantom{b_{1p}} \end{array} \\
\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mp} \end{pmatrix}
\end{array}
=
\begin{array}{cc}
 & \begin{array}{cccc} j & \phantom{c_{12}} & & \phantom{c_{1p}} \end{array} \\
i & \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{pmatrix}
\end{array}
$$

$n \times m$ matrix $\quad$ $m \times p$ matrix $\quad$ $n \times p$ matrix

=

$$c_{ij} = \Sigma_{k=1..m}\, a_{ik} \cdot b_{kj}$$

Regneregler

$$(AB)C = A(BC)$$
$$A(B+C) = (AB) + (AC)$$
$$(A+B)C = (AC) + (BC)$$

$$AB \neq BA$$

$$\begin{pmatrix} 2 & 1 & 4 \\ 1 & 3 & 2 \end{pmatrix} \begin{pmatrix} 3 & 1 \\ 2 & 2 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 12 & 20 \\ 11 & 15 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 1 \\ 2 & 2 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} 2 & 1 & 4 \\ 1 & 3 & 2 \end{pmatrix} = \begin{pmatrix} 7 & 6 & 14 \\ 6 & 8 & 12 \\ 6 & 13 & 12 \end{pmatrix}$$

Matrix Multiplikation

$$\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{np} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mp} \end{pmatrix}$$

```
MATRIX-MULTIPLY(A, B)
if A.columns ≠ B.rows
    error "incompatible dimensions"
else let C be a new A.rows × B.columns matrix
    for i = 1 to A.rows
        for j = 1 to B.columns
            c_ij = 0
            for k = 1 to A.columns
                c_ij = c_ij + a_ik · b_kj
    return C
```

Naive implementation: tid $O(npm)$

(Kvadratisk) Matrix Multiplikation

[CLRS, kapitel 4.2]

$\frac{n}{2} \times \frac{n}{2}$ matrix

$$\underbrace{\begin{pmatrix} I & J \\ K & L \end{pmatrix}}_{n \times n \text{ matrix}} = \underbrace{\begin{pmatrix} A & B \\ C & D \end{pmatrix}}_{n \times n \text{ matrix}} \underbrace{\begin{pmatrix} E & F \\ G & H \end{pmatrix}}_{n \times n \text{ matrix}}$$

$$\begin{aligned} I &= A \cdot E + B \cdot G \\ J &= A \cdot F + B \cdot H \\ K &= C \cdot E + D \cdot G \\ L &= C \cdot F + D \cdot H \end{aligned}$$

- $A, B, \ldots, K, L$ er $n/2$ x $n/2$-matricer
- I, J, K, L kan beregnes med 8 rekursive multiplikationer og 4 matrix additioner på $n/2$ x $n/2$ -matricer
- $T(n) \leq 8 \cdot T(n/2) + c \cdot n^2$ for $n \geq 2$

  $T(n) \leq c$ for $n = 1$
- $T(n) = O(n^{\log_2 8}) = O(n^3)$

Strassen's Matrix Multiplikation

1969

$$\begin{pmatrix} I & J \\ K & L \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

$$\begin{aligned}
I &= S_5 + S_6 + S_4 - S_2 \\
&= (A+D)(E+H) + (B-D)(G+H) + D(G-E) - (A+B)H \\
&= AE + DE + AH + DH + BG - DG + BH - DH + DG - DE - AH - BH \\
&= AE + BG.
\end{aligned}$$

$$\begin{aligned}
J &= S_1 + S_2 \\
&= A(F-H) + (A+B)H \\
&= AF - AH + AH + BH \\
&= AF + BH.
\end{aligned}$$

$$\begin{aligned}
K &= S_3 + S_4 \\
&= (C+D)E + D(G-E) \\
&= CE + DE + DG - DE \\
&= CE + DG.
\end{aligned}$$

$$\begin{aligned}
L &= S_1 - S_7 - S_3 + S_5 \\
&= A(F-H) - (A-C)(E+F) - (C+D)E + (A+D)(E+H) \\
&= AF - AH - AE + CE - AF + CF - CE - DE + AE + DE + AH + DH \\
&= CF + DH.
\end{aligned}$$

$$\boxed{\begin{aligned}
S_1 &= A \cdot (F-H) \\
S_2 &= (A+B) \cdot H \\
S_3 &= (C+D) \cdot E \\
S_4 &= D \cdot (G-E) \\
S_5 &= (A+D) \cdot (E+H) \\
S_6 &= (B-D) \cdot (G+H) \\
S_7 &= (A-C) \cdot (E+F)
\end{aligned}}$$

7 rekursive multiplikationer

Strassen's Matrix Multiplikation

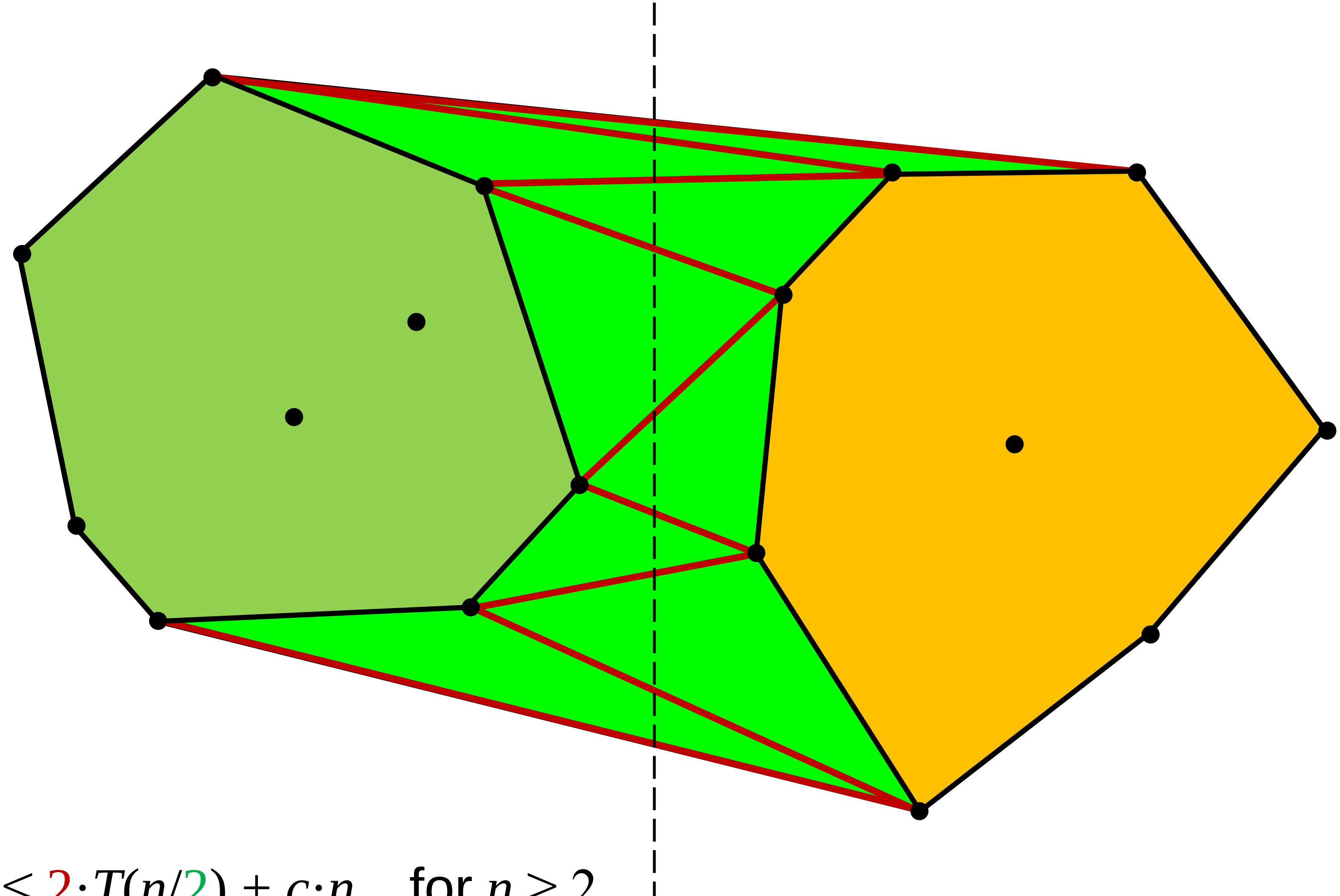
- Bruger 18 matrix additioner (tid $O(n^2)$) og 7 rekursive matrix multiplikationer

$$
\boxed{\begin{array}{ll}
T(n) \le 7 \cdot T(n/2) + c \cdot n^2 & \text{for } n \ge 2 \\
T(n) \le c & \text{for } n = 1
\end{array}}
$$

- $T(n) = \mathrm{O}(n^{\log_2 7}) = O(n^{2.81})$

Bedste resultat for matix multiplikation $\mathrm{O}(n^{2.3729})$: Virginia Vassilevska Williams, *Multiplying matrices faster than Coppersmith-Winograd*, STOC 2012

Konveks Hylster



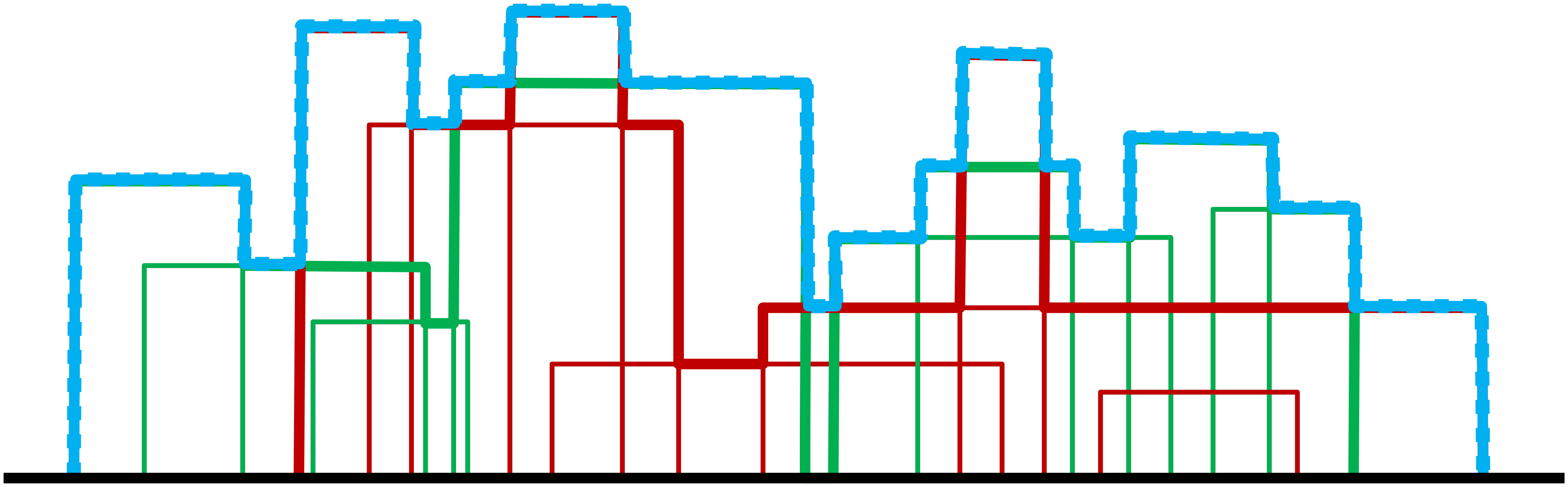
$T(n) \leq 2 \cdot T(n/2) + c \cdot n$ for $n \geq 2$

$T(n) \leq c$ for $n = 1$

$T(n) = O(n \cdot \log n)$

Silhuet

(afleveringsopgave)



$$T(n) \leq ? \cdot T(n/?) + ? \quad \text{for } n \geq 2$$

$$T(n) \leq c \quad \text{for } n = 1$$