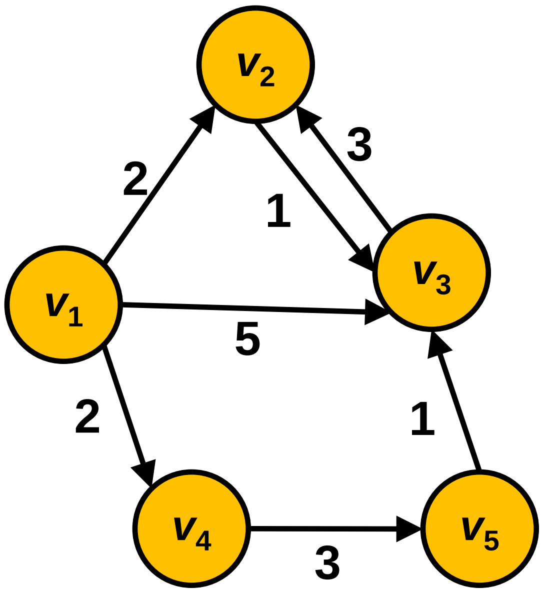
Algoritmer og Datastrukturer 2

Gerth Stølting Brodal

Korteste Veje
[CLRS, kapitel 25.1-25.2]

AARHUS UNIVERSITET

Korteste Veje mellem alle Par af Knude



d_{ij}

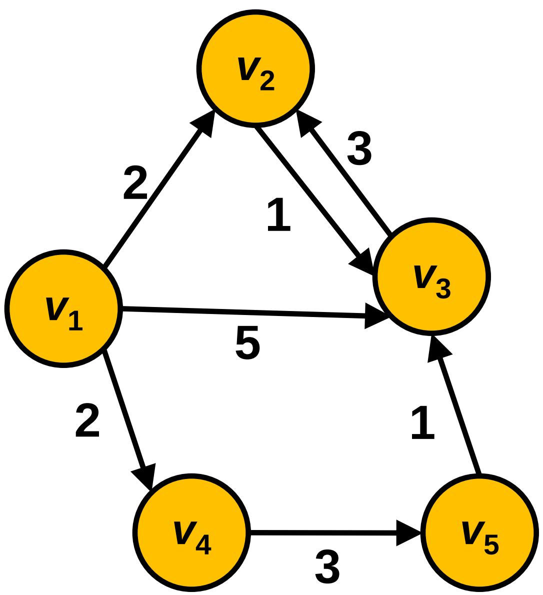
	1	2	3	4	5
1	0	2	3	2	5
2	+∞	0	1	+∞	+∞
3	+∞	3	0	+∞	+∞
4	+∞	7	4	0	3
5	+∞	4	1	+∞	0

π_{ij}

	1	2	3	4	5
1	NIL	1	2	1	4
2	NIL	NIL	2	NIL	NIL
3	NIL	3	NIL	NIL	NIL
4	NIL		5	NIL	4
5	NIL	3	5	NIL	NIL

```
PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)
if i == j
    print i
elseif π_ij == NIL
    print "no path from" i "to" j "exists"
else PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, π_ij)
    print j
```

Tid $O(n)$



d_{ij}

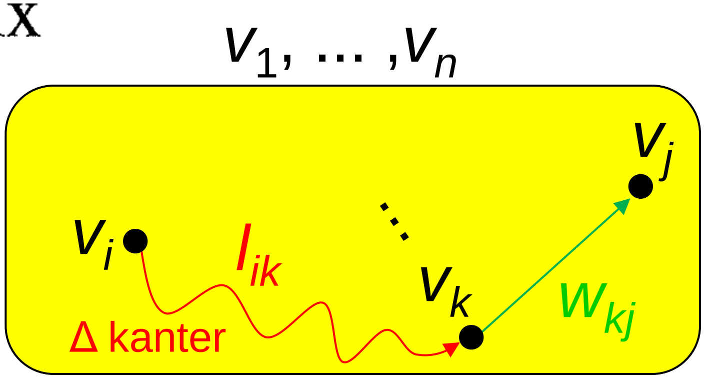
	1	2	3	4	5
1	0	2	3	2	5
2	+∞	0	1	+∞	+∞
3	+∞	3	0	+∞	+∞
4	+∞	7	4	0	3
5	+∞	4	1	+∞	0

π_{ij}

	1	2	3	4	5
1	NIL	1	2	1	4
2	NIL	NIL	2	NIL	NIL
3	NIL	3	NIL	NIL	NIL
4	NIL	3	5	NIL	4
5	NIL	3	5	NIL	NIL

EXTEND-SHORTEST-PATHS(L, W)

$n = L.rows$
let $L' = (l'_{ij})$ be a new $n \times n$ matrix
for $i = 1$ **to** n
    for $j = 1$ **to** n
        $l'_{ij} = \infty$
        for $k = 1$ **to** n
            $l'_{ij} = \min(l'_{ij}, l_{ik} + w_{kj})$
return L'



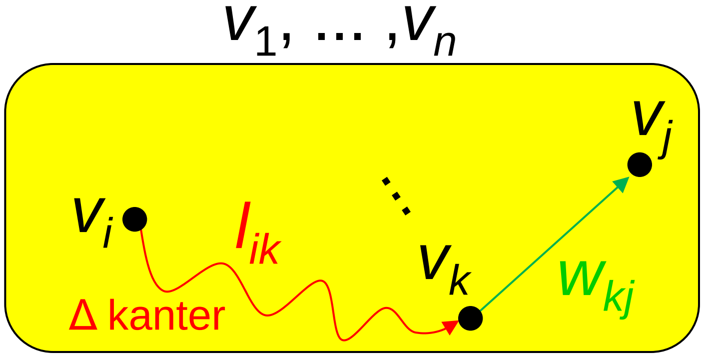
L_{ij} = korteste afstand fra *i* til *j* for stier med Δ kanter
W = incidensmatricen
L'_{ij} = korteste afstand fra *i* til *j* for stier med Δ+1 kanter

Tid $O(n^3)$

SQUARE-MATRIX-MULTIPLY(A, B)

```
n = A.rows
let C be a new n × n matrix
for i = 1 to n
    for j = 1 to n
        c_ij = 0
        for k = 1 to n
            c_ij = c_ij + a_ik · b_kj
return C
```

Tid $O(n^3)$

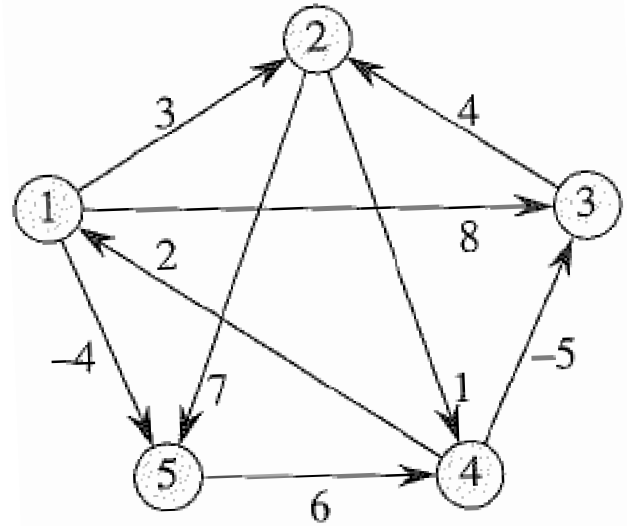


SLOW-ALL-PAIRS-SHORTEST-PATHS(W)

```
n = W.rows
L^(1) = W                                  ← diagonalen = 0
for m = 2 to n − 1
    let L^(m) be a new n × n matrix
    L^(m) = EXTEND-SHORTEST-PATHS(L^(m−1), W)
return L^(n−1)
```

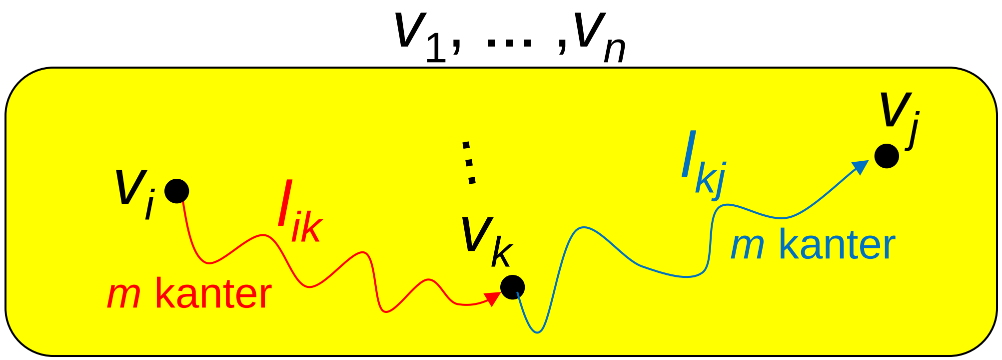
$L^{(m)}_{ij}$ = korteste afstand fra *i* til *j* for stier med *m* kanter
W = incidensmatricen

Tid $O(n^4)$



$$L^{(1)} = \begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \quad L^{(2)} = \begin{pmatrix} 0 & 3 & 8 & 2 & -4 \\ 3 & 0 & -4 & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & \infty & 1 & 6 & 0 \end{pmatrix}$$

$$L^{(3)} = \begin{pmatrix} 0 & 3 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix} \quad L^{(4)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$



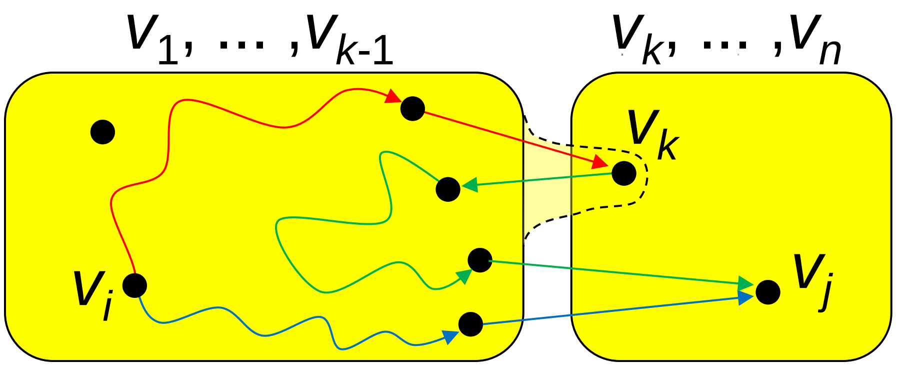
FASTER-ALL-PAIRS-SHORTEST-PATHS(W)

$n = W.rows$
$L^{(1)} = W$
$m = 1$
while $m < n - 1$
  let $L^{(2m)}$ be a new $n \times n$ matrix
  $L^{(2m)} = $ EXTEND-SHORTEST-PATHS$(L^{(m)}, L^{(m)})$
  $m = 2m$
return $L^{(m)}$

$L^{(m)}_{ij}$ = korteste afstand fra i til j for stier med m kanter
W = incidensmatricen

Tid $O(n^3 \cdot \log n)$

Floyd-Warshall



FLOYD-WARSHALL(W)

```
n = W.rows
D^(0) = W
for k = 1 to n
    let D^(k) = (d_ij^(k)) be a new n × n matrix
    for i = 1 to n
        for j = 1 to n
            d_ij^(k) = min(d_ij^(k-1), d_ik^(k-1) + d_kj^(k-1))
return D^(n)
```

$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

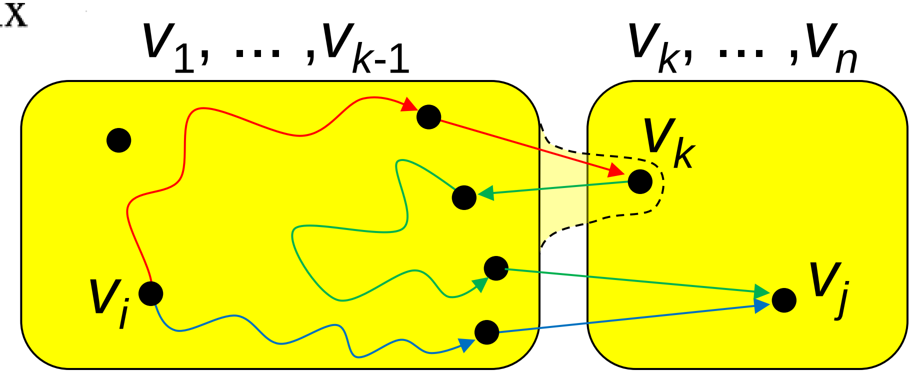
$d^{(k)}_{ij}$ = korteste vej fra *i* til *j* der kun går **via** 1..*k*

Tid $O(n^3)$

Transitive Lukning

(= Floyd-Warshall simplificeret)

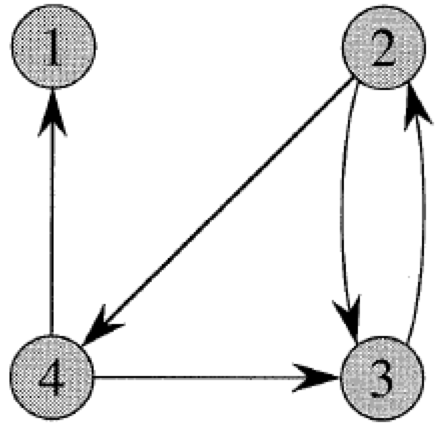
```
TRANSITIVE-CLOSURE(G)
n = |G.V|
let T^(0) = (t_ij^(0)) be a new n × n matrix
for i = 1 to n
    for j = 1 to n
        if i == j or (i, j) ∈ G.E
            t_ij^(0) = 1
        else t_ij^(0) = 0
for k = 1 to n
    let T^(k) = (t_ij^(k)) be a new n × n matrix
    for i = 1 to n
        for j = 1 to n
            t_ij^(k) = t_ij^(k-1) ∨ (t_ik^(k-1) ∧ t_kj^(k-1))
return T^(n)
```



$t^{(k)}_{ij}$ = findes en vej fra *i* til *j* der kun går via 1..*k*

Tid $O(n^3)$

Transitive Lukning: Eksempel



$$T^{(0)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad T^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \quad T^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

$$T^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad T^{(4)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Opsummering Korteste Veje

		SSSP En-til-alle korteste veje	APSP Alle-til-alle korteste veje
Acykliske grafer (positive og negative vægte)		$O(n+m)$	$O(n\cdot(n+m))$
Generelle grafer	Kun positive vægte	**Dijkstra** $O((n+m)\cdot\log n)$ $O(n^2+m)$	**Floyd-Warshall** $O(n^3)$
	Positive og negative vægte	**Bellman-Ford** $O(m\cdot n)$	