# Algoritmer og Datastrukturer 2

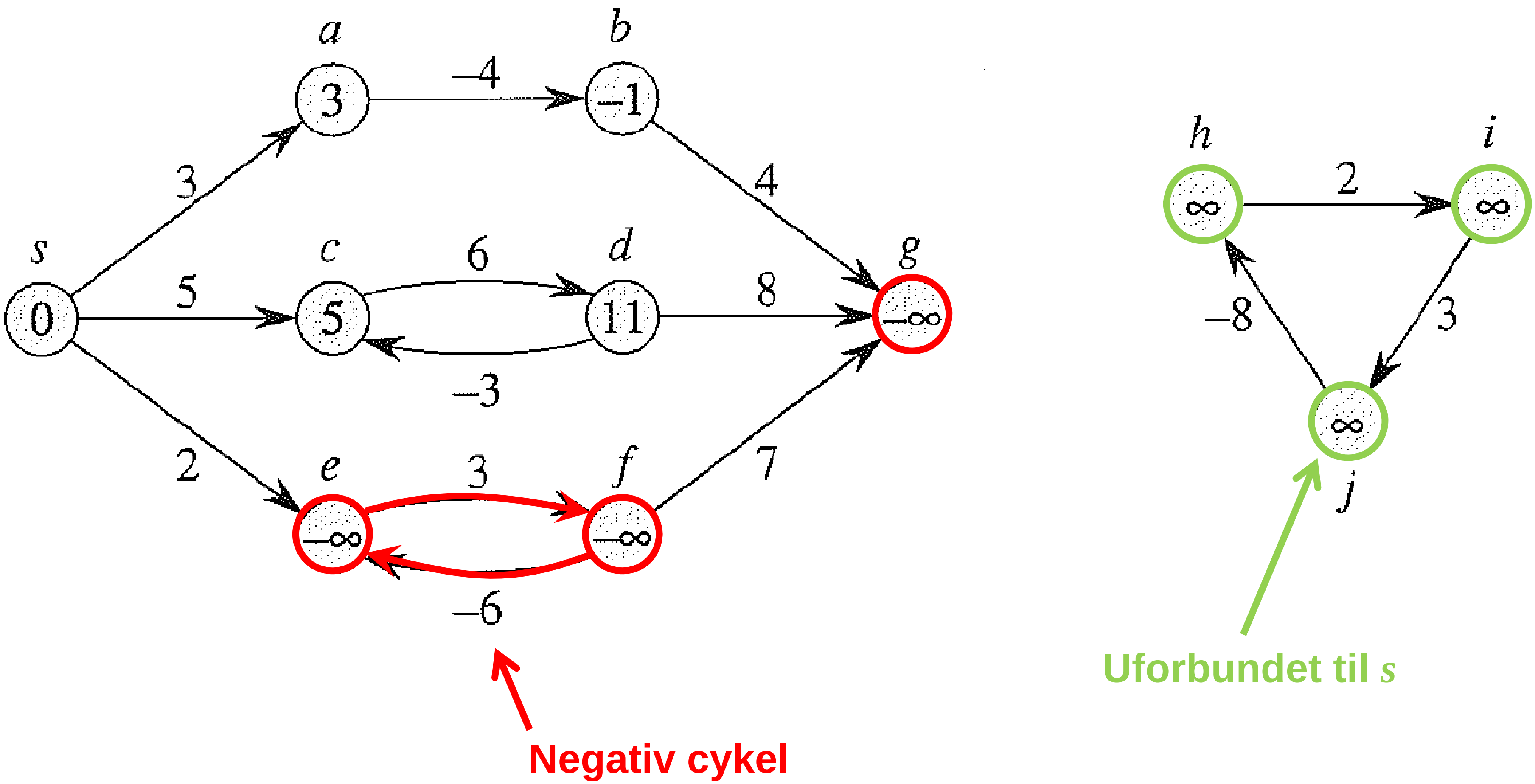
## Korteste Veje

[CLRS, kapitel 24, 25.1-25.2]

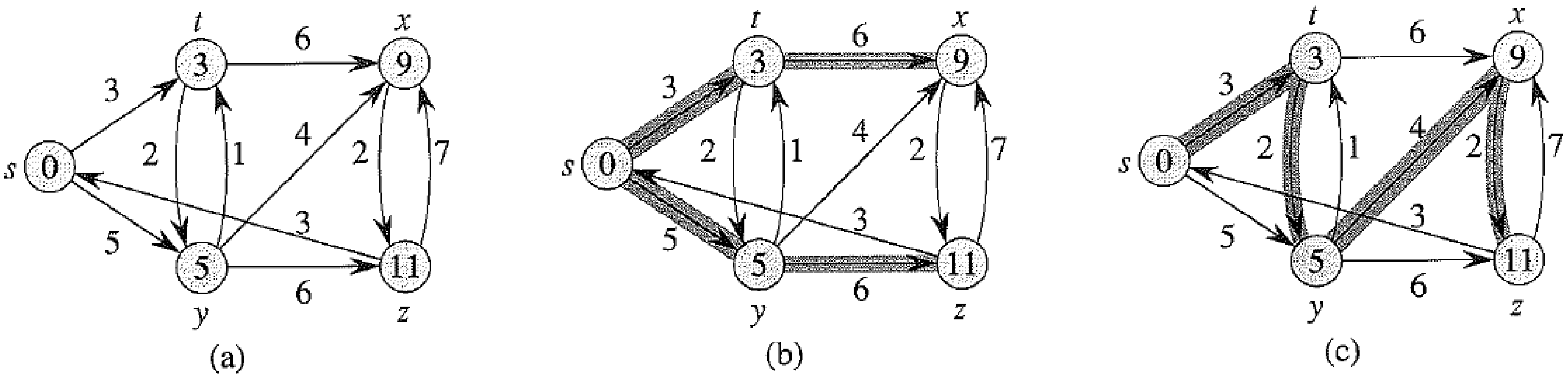

**Gerth Stølting Brodal**

**Aarhus Universitet**

# Eksempel: Korteste veje fra *s*



# Eksempel: Korteste veje træer



# Korteste Veje Estimater : Initialisering

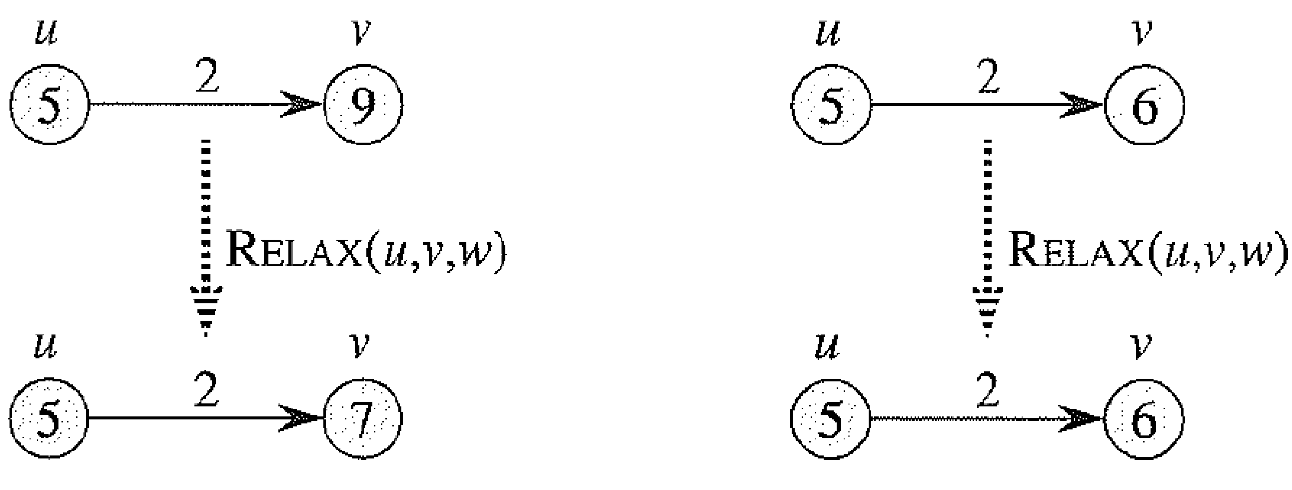
INITIALIZE-SINGLE-SOURCE$(G, s)$

**for** each vertex $v \in V[G]$
    **do** $d[v] \leftarrow \infty$
        $\pi[v] \leftarrow$ NIL
$d[s] \leftarrow 0$

# Korteste Veje Estimater : Relax

RELAX$(u, v, w)$

**if** $d[v] > d[u] + w(u, v)$
    **then** $d[v] \leftarrow d[u] + w(u, v)$
        $\pi[v] \leftarrow u$




(a) (b)

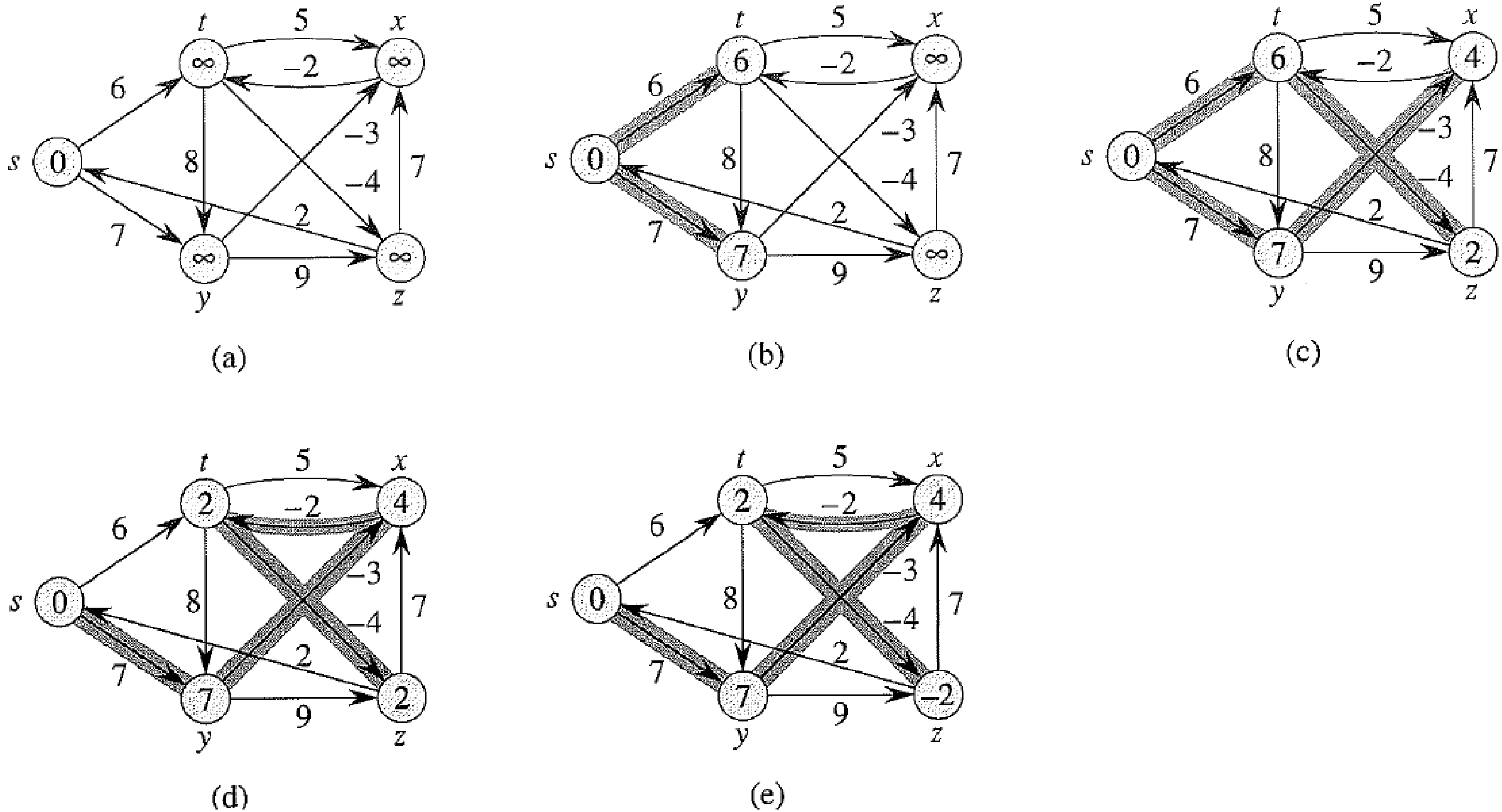
# Bellman-Ford:

## Korteste Veje i Grafer med Negative Vægte

```
BELLMAN-FORD(G, w, s)
    INITIALIZE-SINGLE-SOURCE(G, s)
    for i ← 1 to |V[G]| − 1
        do for each edge (u, v) ∈ E[G]
            do RELAX(u, v, w)
    for each edge (u, v) ∈ E[G]
        do if d[v] > d[u] + w(u, v)
            then return FALSE
    return TRUE
```

**Tid** $O(nm)$

# Bellman-Ford: Eksempel

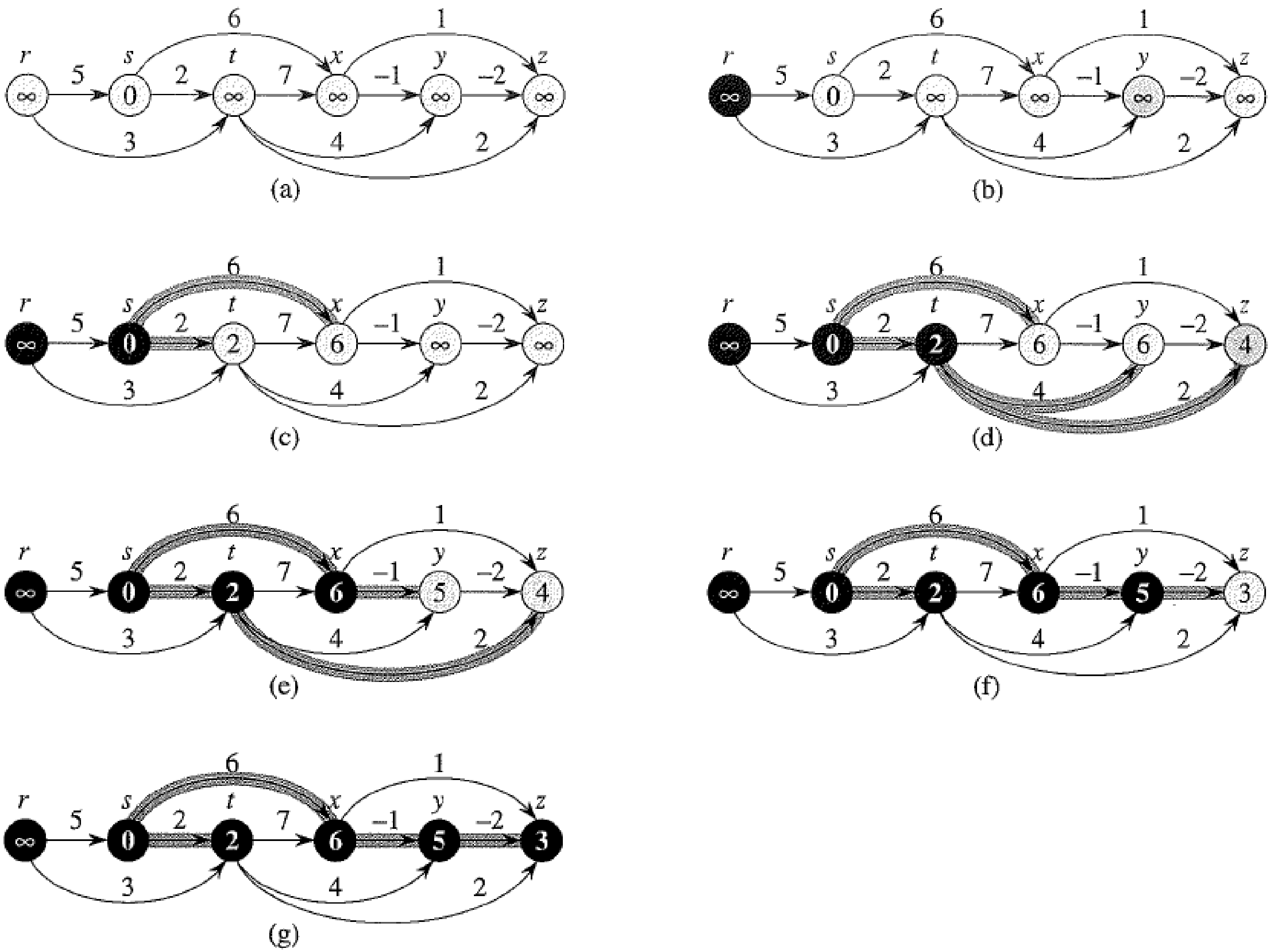


# Korteste Veje i Acycliske Grafer

```
DAG-SHORTEST-PATHS(G, w, s)
    topologically sort the vertices of G
    INITIALIZE-SINGLE-SOURCE(G, s)
    for each vertex u, taken in topologically sorted order
        do for each vertex v ∈ Adj[u]
            do RELAX(u, v, w)
```

**Tid** $O(n+m)$

# Acykliske Grafer : Eksempel



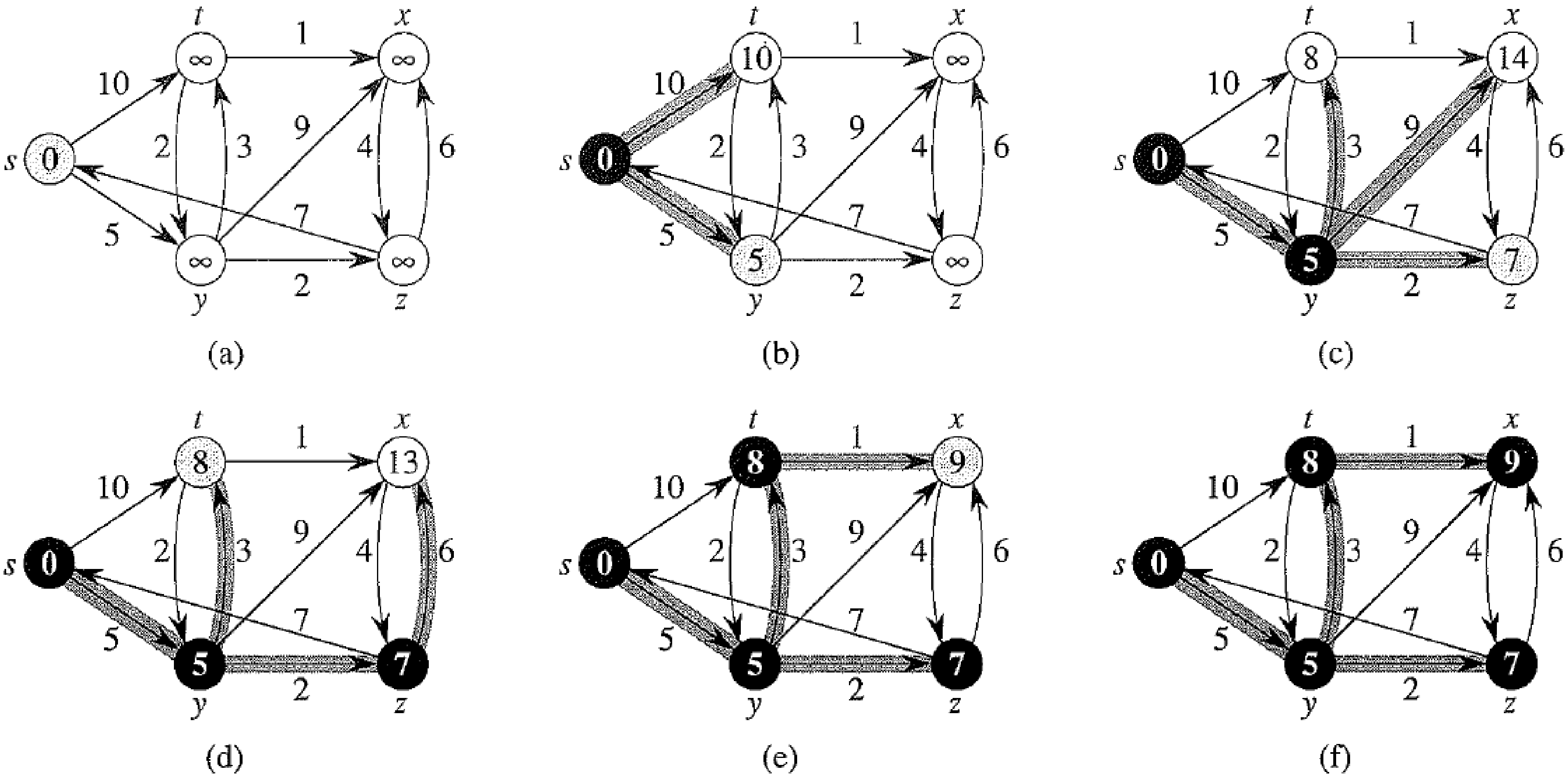
# Dijkstra:

## Korteste Veje i Grafer uden Negative Vægte

```
DIJKSTRA(G, w, s)
INITIALIZE-SINGLE-SOURCE(G, s)
S ← Ø
Q ← V[G]
while Q ≠ Ø
    do u ← EXTRACT-MIN(Q)
       S ← S ∪ {u}
       for each vertex v ∈ Adj[u]
           do RELAX(u, v, w)
```

**Tid** $O((n+m)\cdot\log n)$
**eller** $O(n^2+m)$

# Dijkstra : Eksempel



# Opsummering

| | | SSSP<br>En-til-alle korteste veje |
|---|---|---|
| Acykliske grafer (positive og negative vægte) | | $O(n+m)$ |
| Generelle grafer | Kun positive vægte | **Dijkstra**<br>$O((n+m)\cdot\log n)$ |
| | Positive og negative vægte | **Bellman-Ford**<br>$O(m\cdot n)$ |

# Korteste Veje mellem alle Par af Knude

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

EXTEND-SHORTEST-PATHS$(L, W)$

$n \leftarrow rows[L]$<br>
let $L' = (l'_{ij})$ be an $n \times n$ matrix<br>
**for** $i \leftarrow 1$ **to** $n$<br>
    **do for** $j \leftarrow 1$ **to** $n$<br>
        **do** $l'_{ij} \leftarrow \infty$<br>
           **for** $k \leftarrow 1$ **to** $n$<br>
               **do** $l'_{ij} \leftarrow \min(l'_{ij}, l_{ik} + w_{kj})$<br>
**return** $L'$

**Tid** $O(n^3)$

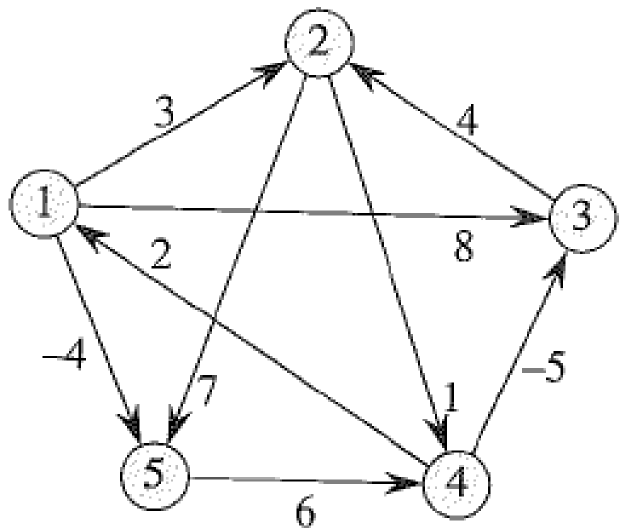
```
MATRIX-MULTIPLY(A, B)
n ← rows[A]
let C be an n × n matrix
for i ← 1 to n
    do for j ← 1 to n
        do c_ij ← 0
            for k ← 1 to n
                do c_ij ← c_ij + a_ik · b_kj
return C
```

**Tid** $O(n^3)$

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

$n \leftarrow rows[W]$
$L^{(1)} \leftarrow W$
**for** $m \leftarrow 2$ **to** $n - 1$
    **do** $L^{(m)} \leftarrow$ EXTEND-SHORTEST-PATHS$(L^{(m-1)}, W)$
**return** $L^{(n-1)}$

**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$(W)$

$n \leftarrow rows[W]$
$L^{(1)} \leftarrow W$
$m \leftarrow 1$
**while** $m < n - 1$
  **do** $L^{(2m)} \leftarrow$ EXTEND-SHORTEST-PATHS$(L^{(m)}, L^{(m)})$
    $m \leftarrow 2m$
**return** $L^{(m)}$

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

# Floyd-Warshall

FLOYD-WARSHALL$(W)$

$n \leftarrow rows[W]$
$D^{(0)} \leftarrow W$
**for** $k \leftarrow 1$ **to** $n$
    **do for** $i \leftarrow 1$ **to** $n$
        **do for** $j \leftarrow 1$ **to** $n$
            **do** $d_{ij}^{(k)} \leftarrow \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$
**return** $D^{(n)}$

**Tid** $O(n^3)$

# Transitive Lukning

TRANSITIVE-CLOSURE$(G)$

$n \leftarrow |V[G]|$
**for** $i \leftarrow 1$ **to** $n$
  **do for** $j \leftarrow 1$ **to** $n$
    **do if** $i = j$ or $(i, j) \in E[G]$
      **then** $t_{ij}^{(0)} \leftarrow 1$
      **else** $t_{ij}^{(0)} \leftarrow 0$
**for** $k \leftarrow 1$ **to** $n$
  **do for** $i \leftarrow 1$ **to** $n$
    **do for** $j \leftarrow 1$ **to** $n$
      **do** $t_{ij}^{(k)} \leftarrow t_{ij}^{(k-1)} \vee \left(t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)}\right)$
**return** $T^{(n)}$

**Tid** $O(n^3)$

# Opsummering

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