Algoritmer og Datastrukturer 2

Gerth Stølting Brodal

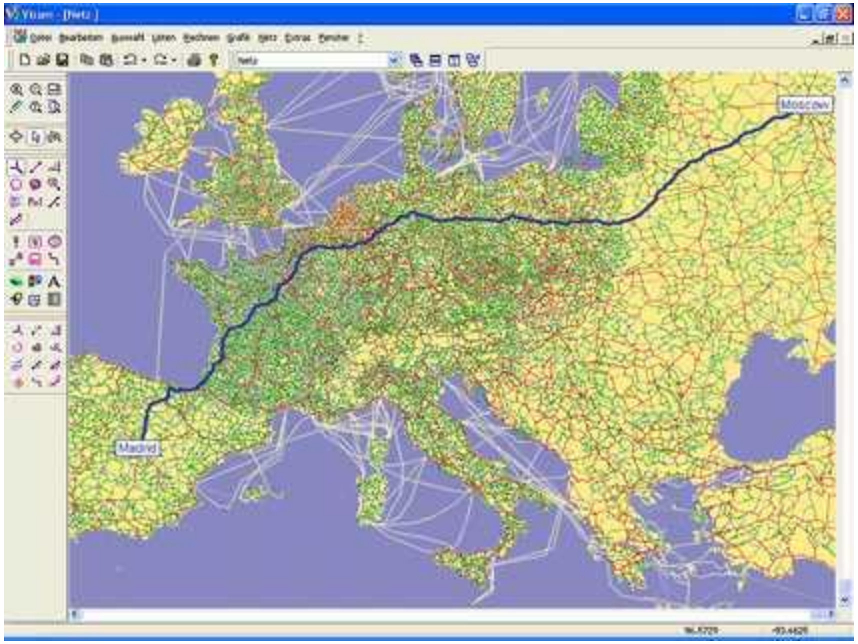
Korteste Veje
[CLRS, kapitel 24]

AARHUS UNIVERSITET

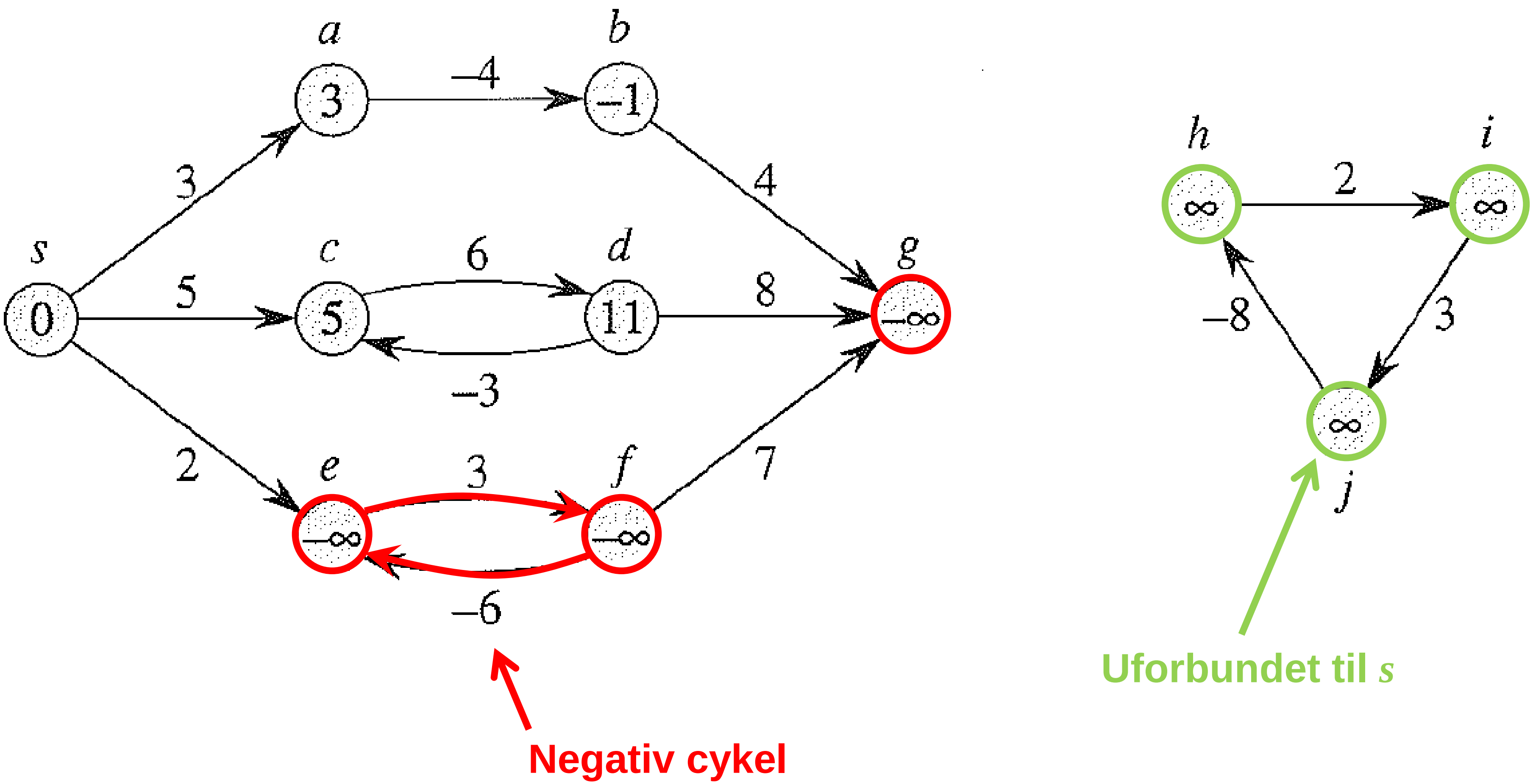
Kort over Vest-Europa

- 18.029.721 knuder
- 42.199.587 orienterede kanter

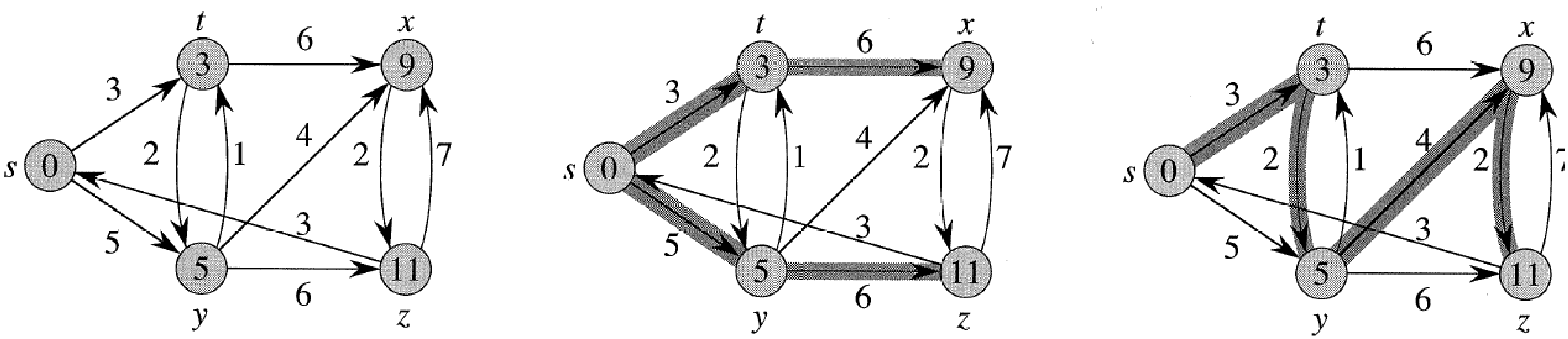
traffic mobility logistics.



Eksempel: Korteste veje fra *s*



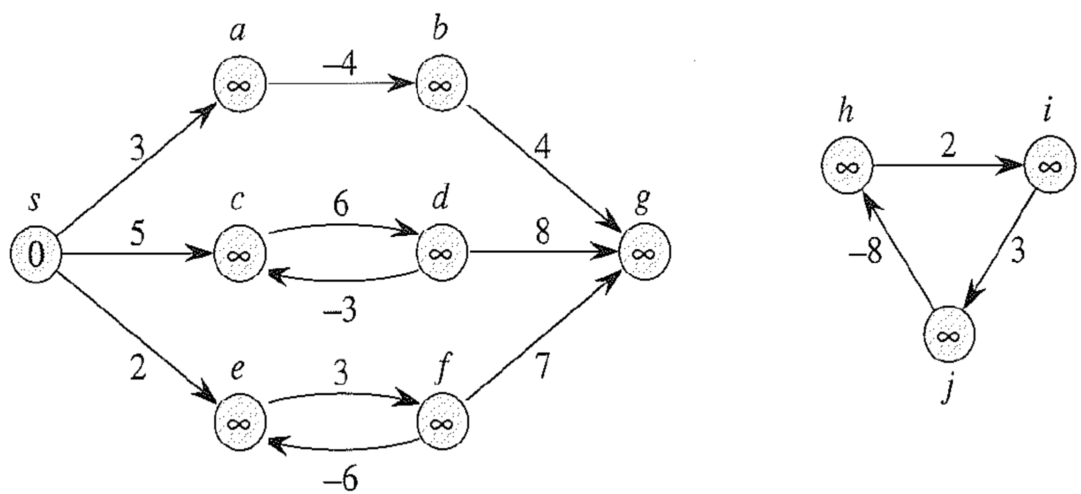
Eksempel: Korteste veje træer



2 forskellige korteste veje træer der repræsenterer stier fra s med samme længde

Korteste Veje Estimater : Initialisering

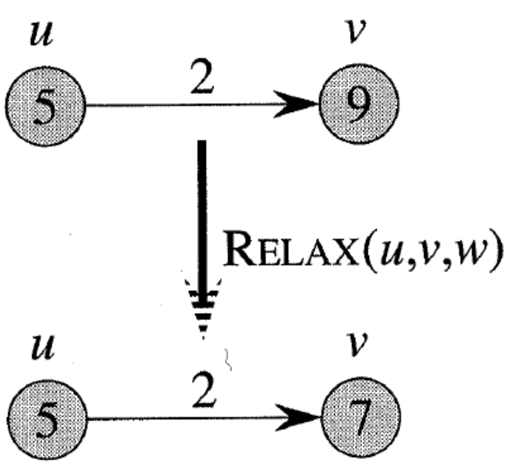
```
INITIALIZE-SINGLE-SOURCE(G, s)
for each vertex v ∈ G.V
    v.d = ∞
    v.π = NIL
s.d = 0
```



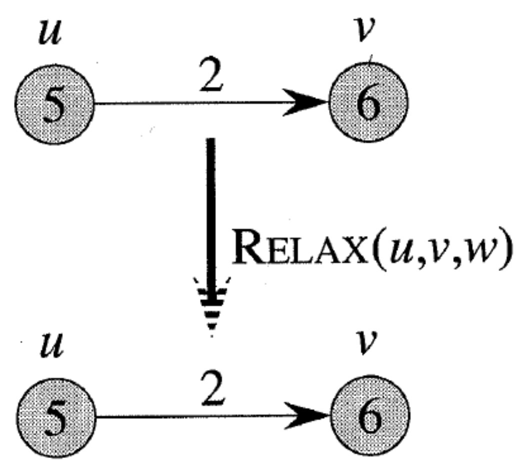
Korteste Veje Estimater : Relax

$\text{RELAX}(u, v, w)$

$$\textbf{if } v.d > u.d + w(u, v)$$
$$\quad v.d = u.d + w(u, v)$$
$$\quad v.\pi = u$$



Kortere afstand til *v* fundet



Forbedrer ikke afstanden til *v*

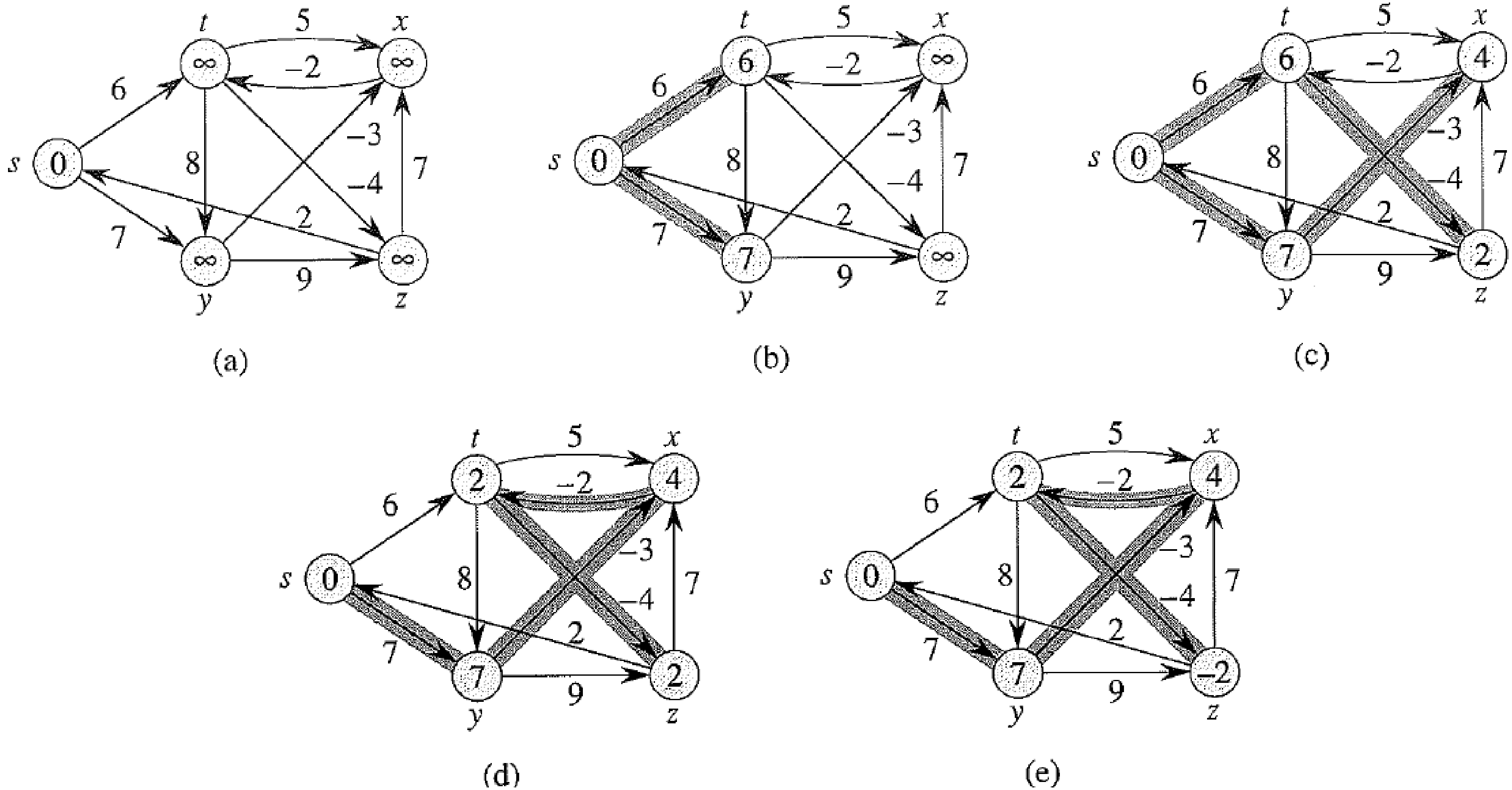
Bellman-Ford:
Korteste Veje i Grafer med Negative Vægte

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

Check for negativ cykel

Tid $O(nm)$

Bellman-Ford: Eksempel

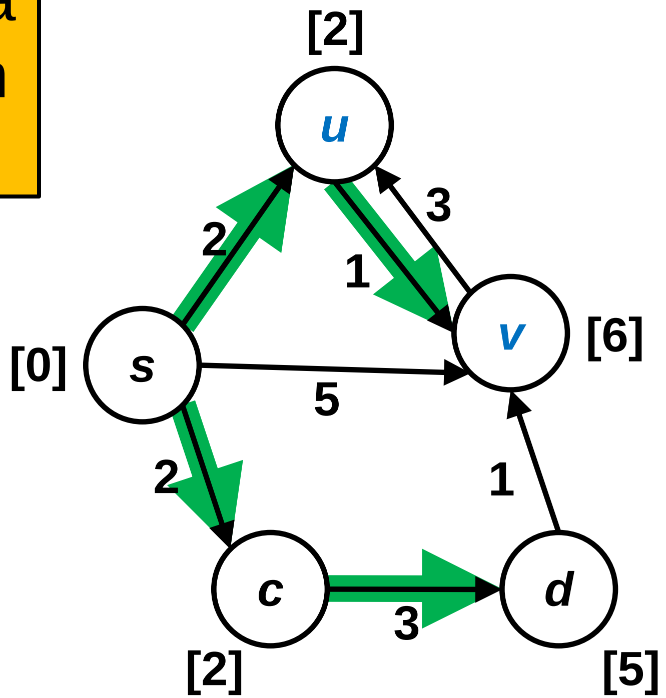


Sætning

Betragt et (ukendt) korteste veje træ *T* hvori ***(u,v)*** er en kant.

Antag den aktuelle *d*[*u*] er den korteste afstand til *u*.

Relax(*u*,*v*,*w*) medfører at *d*[*v*] også er en kortest afstand til *v* (hvis den ikke allerede var det).

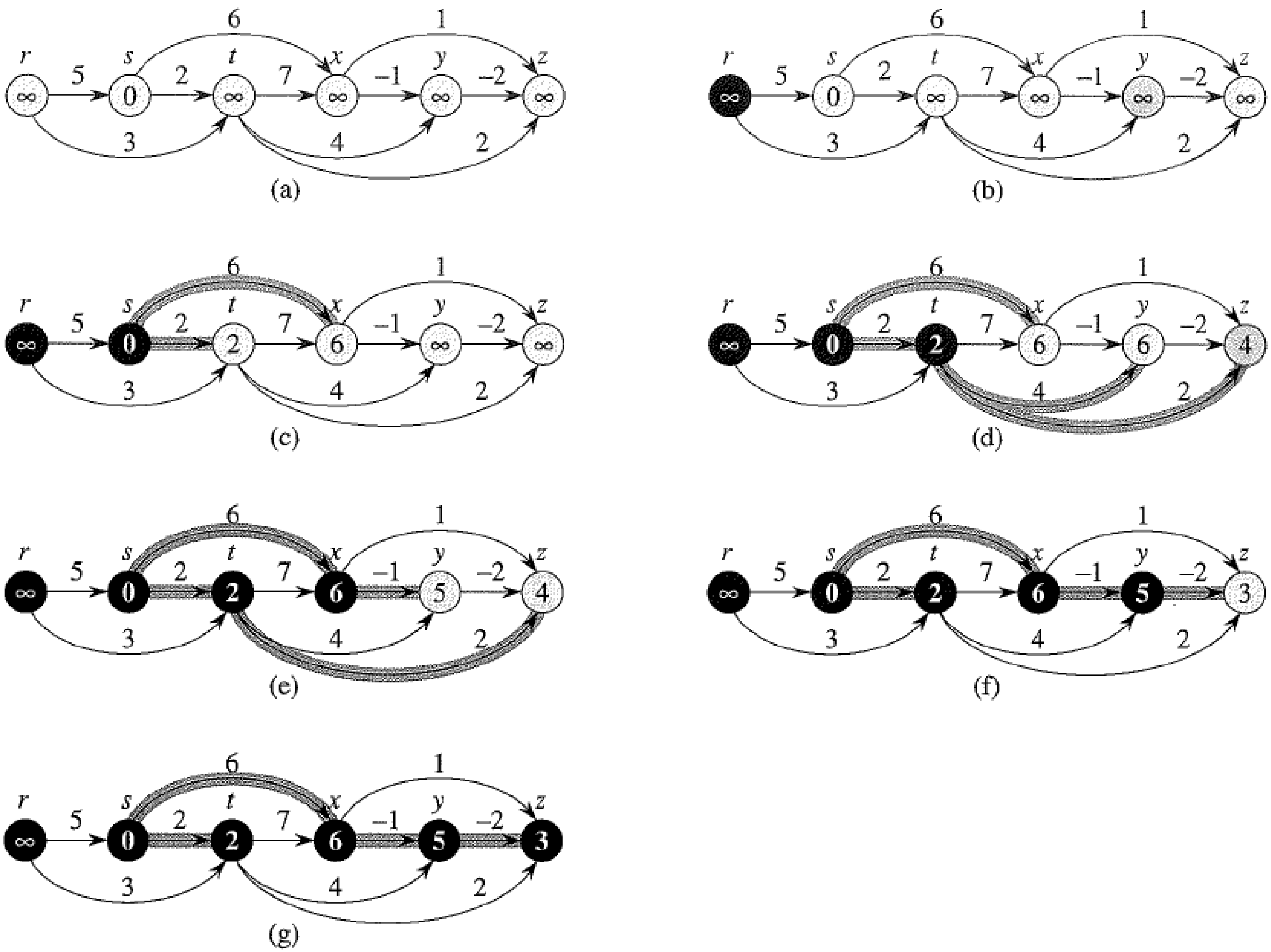


Korteste Veje i Acycliske Grafer

```
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
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Tid $O(n+m)$

Acykliske Grafer : Eksempel

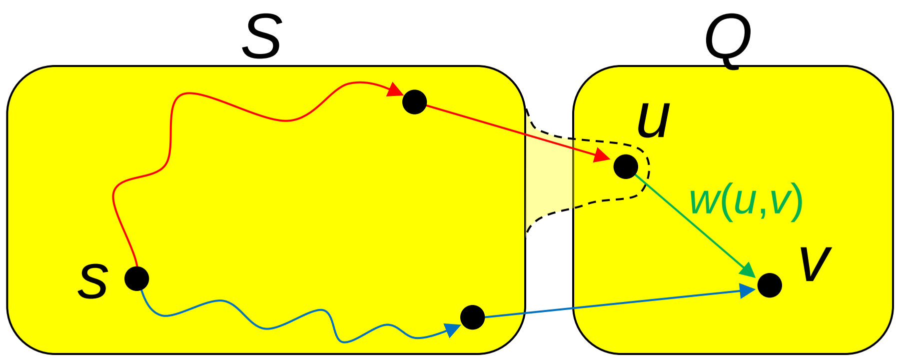


Dijkstra:
Korteste Veje i Grafer uden Negative Vægte

Invarianter

i) $d[v]$ = korteste afstand fra s til v **via knuder i *S***

ii) $\forall p \in S,\ q \in Q : d[p] \leq d[q]$



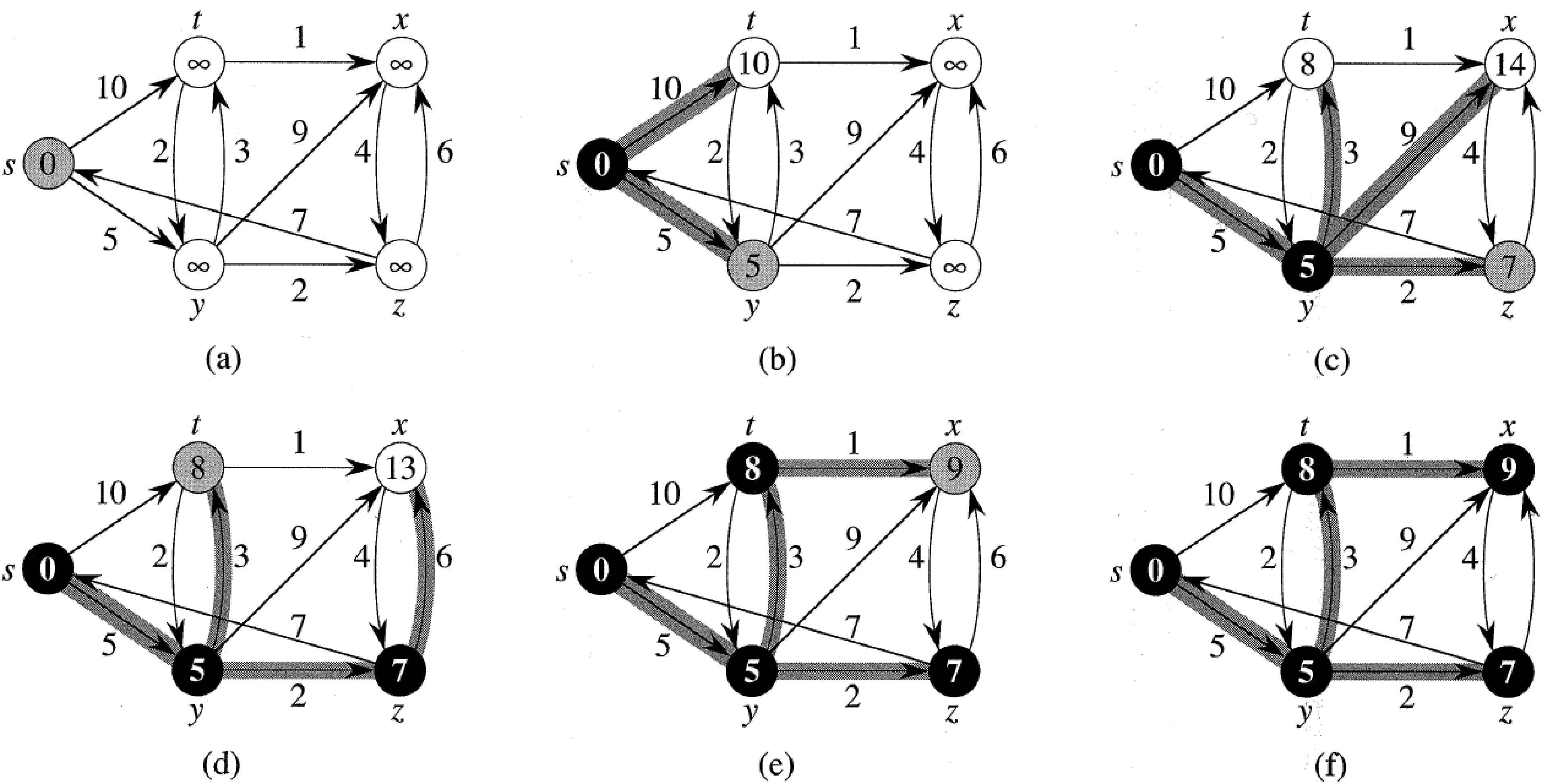
$\text{DIJKSTRA}(G, w, s)$

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

Q = prioritets kø, prioritet = d
(besøger knuderne efter stigende afstand fra *s*)

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

Dijkstra : Eksempel



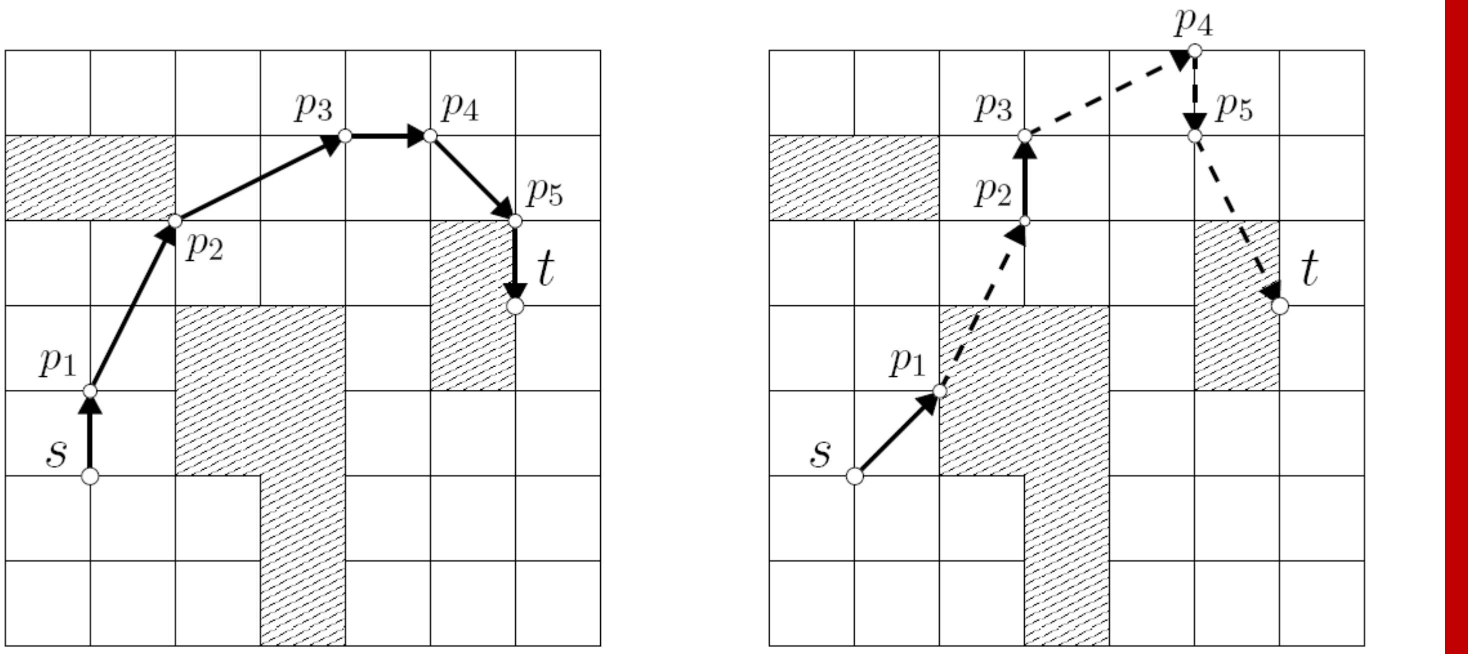
Opsummering

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

Relaxer hver kant præcis én gang

Vektorrace

(find hurtigste vej fra *s* til *t*)



Lovlig sti

Ulovlig sti
(ulovlige stykker af stien er stiplet)

Eksamensopgave Sommeren 2009, opgave 3

Konstruer en uorienteret graf hvor man har en knude (i, j, x, y) for hvert par af et gitterpunkt (i, j) og hastighedsvektor (x, y), hvor hastighedsvektoren (x, y) fører til gitterpunktet (i, j) fra gitterpunktet $(i - x, j - y)$, og hvor $0 \leq i \leq n$, $0 \leq j \leq n$, $i - n \leq x \leq i$ og $j - n \leq y \leq j$. Dvs. grafen har $O(n^4)$ knuder. For alle par af knuder $v = (i, j, x, y)$ og $v' = (i', j', x', y')$ laver vi en kant imellem v og v' hvis og kun hvis følgende er opfyldt:

- $i' = i + x'$ og $j' = j + y'$
- $-1 \leq x' - x \leq 1$ og $-1 \leq y' - y \leq 1$
- alle cellerne der skæres af liniestykket fra (i, j) til (i', j') er frie

Givet (i, j, x, y) findes der højest 3^2 knuder (i', j', x', y') der opfylder de to første krav. Dvs. der er højest $O(n^4)$ potentielle kanter hvor det sidste krav kræver at vi for hver kant checker om $\leq 2n$ krydsende celler er fri. Grafen kan derfor konstrueres i tid $O(n^5)$. Endeligt checkes der vha. BFS om der en sti fra $s = (i_s, j_s, 0, 0)$ til $t = (i_t, j_t, 0, 0)$ i tid $O(n^4)$, hvor (i_s, j_s) og (i_t, j_t) er hhv. start og slut gitterpunktet. Total tid bliver $O(n^5)$.

Note: Da man ikke må køre ud af gitteret og har begrænset acceleration og deacceleration har vi at den maksimale hastighed k i x og y-retningen skal opfylde $1 + 2 + \cdots + k \leq n$, dvs. $k = O(\sqrt{n})$. Dette medfører at antal knuder kan begrænses til $O(n \cdot n \cdot \sqrt{n} \cdot \sqrt{n}) = O(n^3)$ og konstruktionstiden til $O(n^3 \sqrt{n})$.