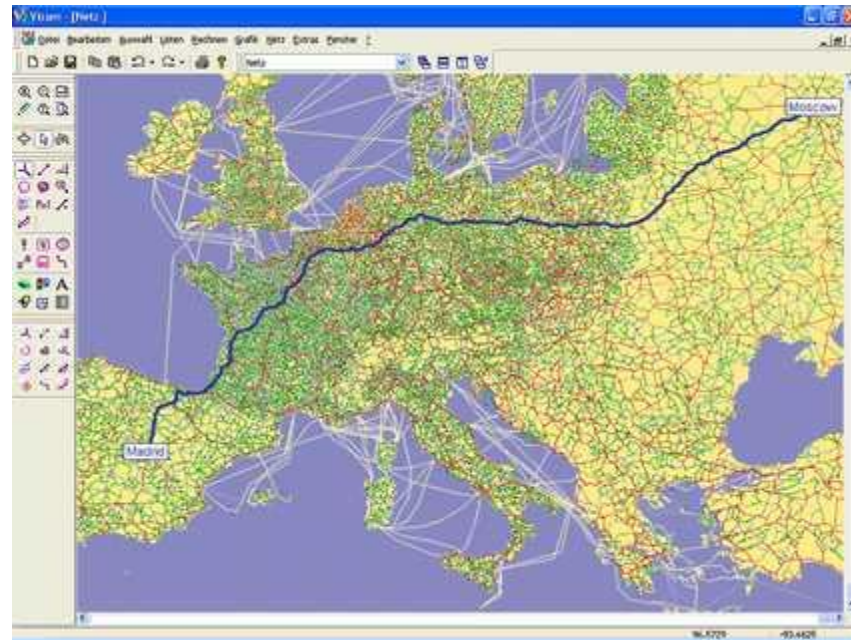


Algoritmer og Datastrukturer

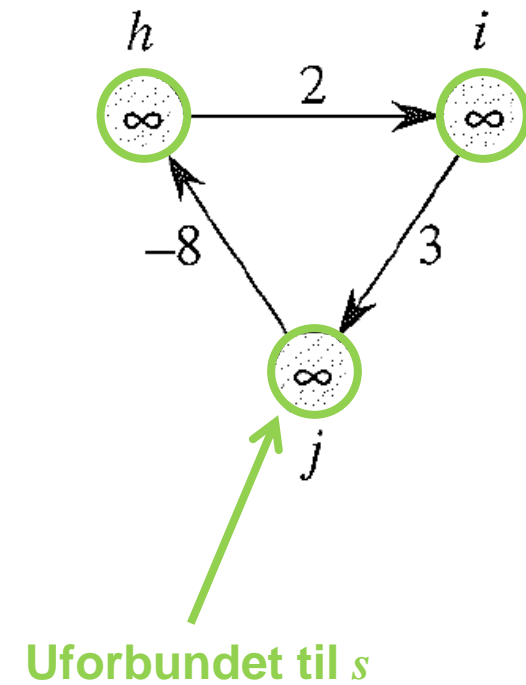
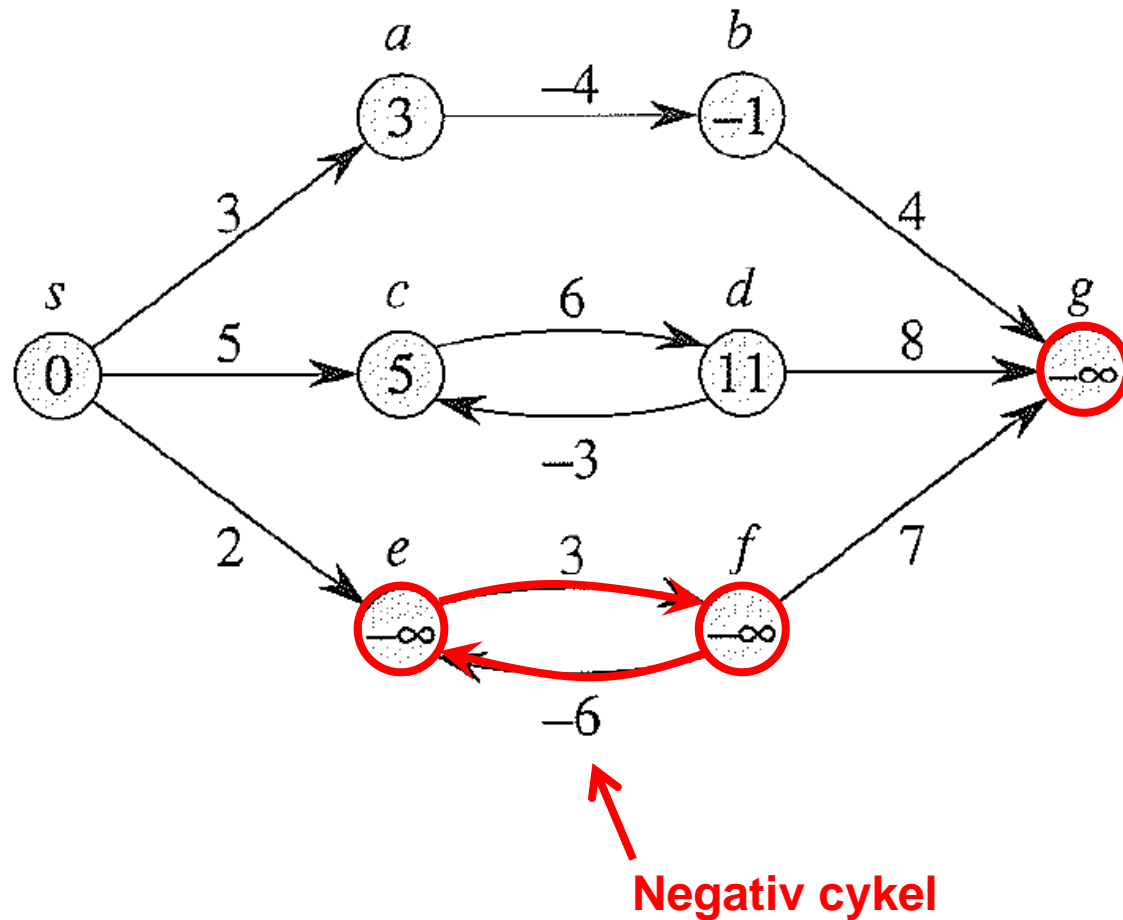
Korteste veje – fra en knude (SSSP)
[CLRS, kapitel 24]

Kort over Vest-Europa

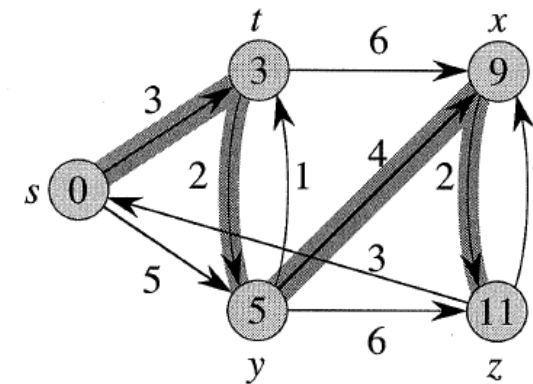
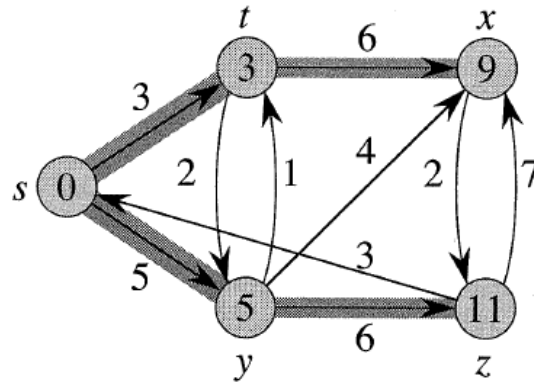
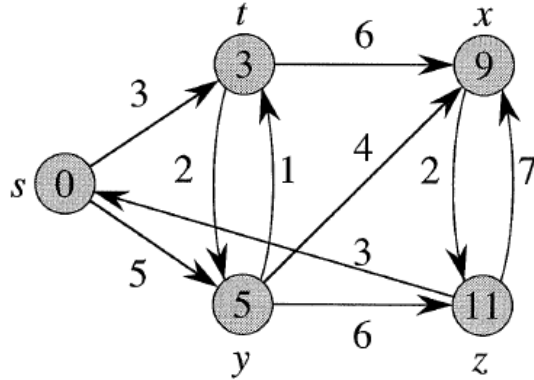
- 18.029.721 knuder
- 42.199.587 orienterede kanter



Eksempel: Korteste veje fra s



Eksempel: Korteste veje træer

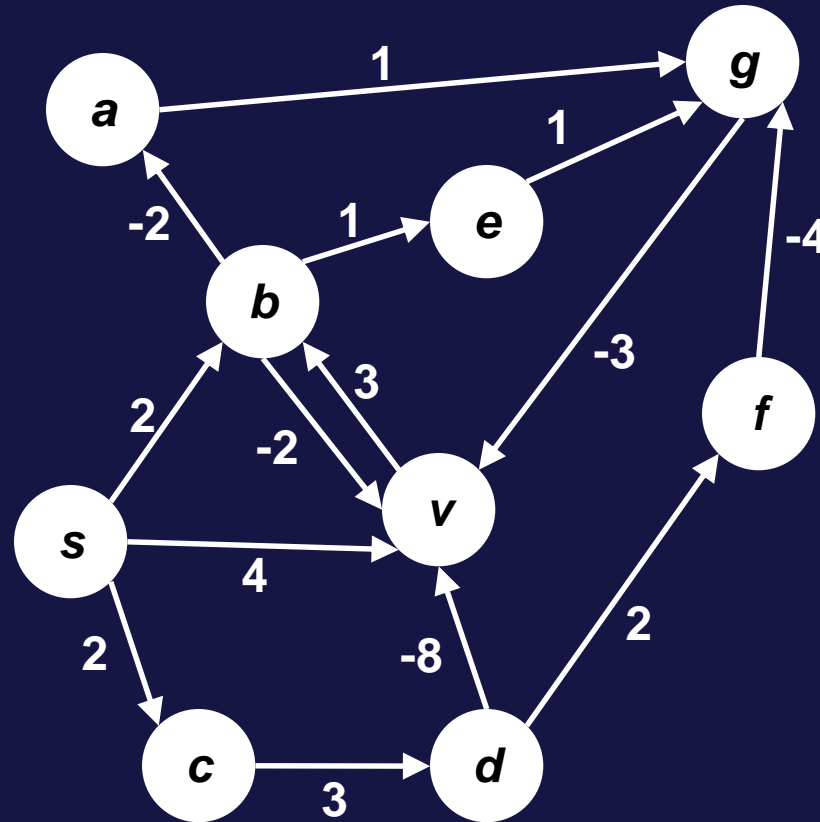


2 forskellige korteste veje træer der repræsenterer stier fra s med samme længde

Hvor lang er den korteste vej fra s til v ?



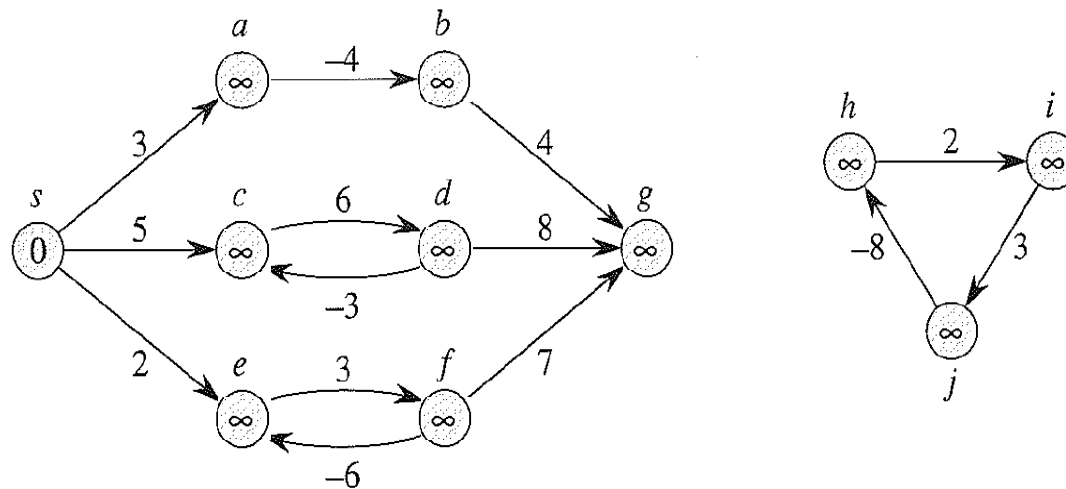
- a) $-\infty$
- b) -3
- c) -1
- d) 0
- e) 4
- f) $+\infty$
- g) Ved ikke



Korteste Veje Estimator : Initialisering

INITIALIZE-SINGLE-SOURCE(G, s)

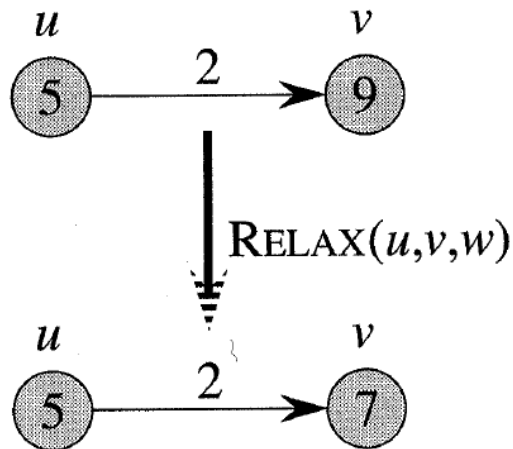
- 1 **for** each vertex $v \in G.V$
- 2 $v.d = \infty$
- 3 $v.\pi = \text{NIL}$
- 4 $s.d = 0$



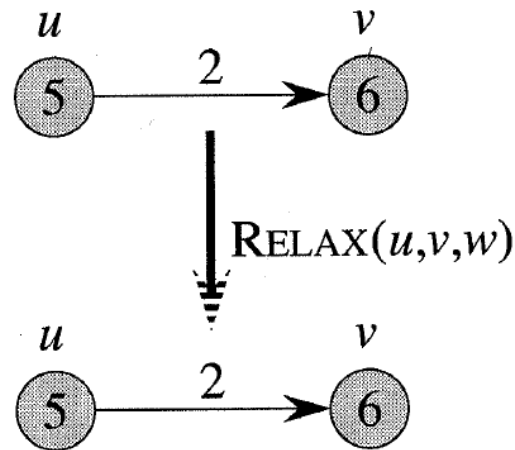
Korteste Veje Estimer : Relax

RELAX(u, v, w)

- 1 **if** $v.d > u.d + w(u, v)$
- 2 $v.d = u.d + w(u, v)$
- 3 $v.\pi = u$



**Kortere afstand til v fundet
(opdater $v.d$ og $v.\pi = u$)**




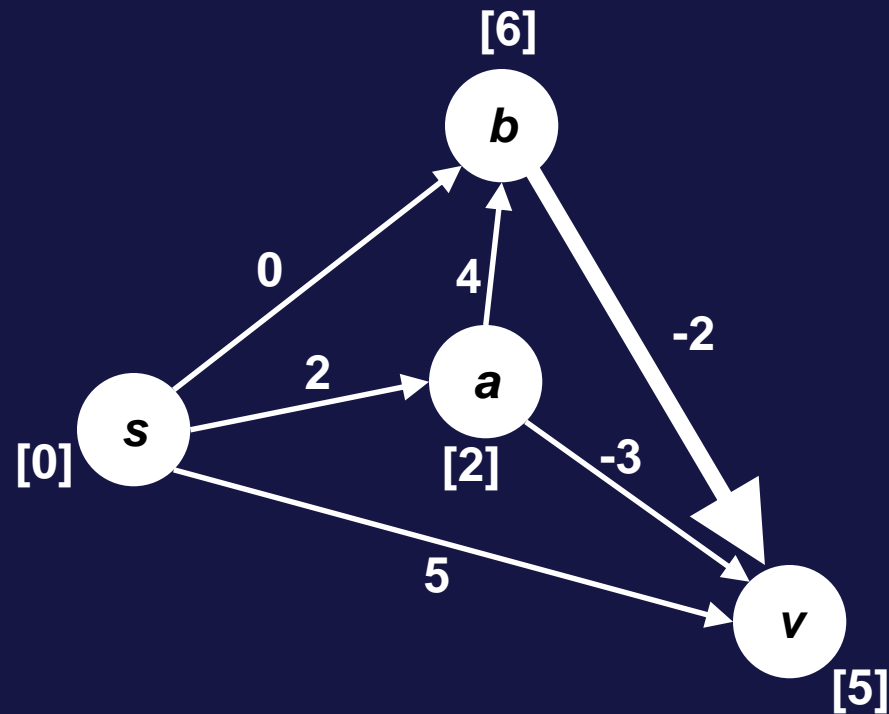
**Forbedrer ikke
afstanden til v**

Invariant

Hvis $v.d < \infty$ så findes en sti fra s til v med afstand $v.d$ og $v.\pi$ som næstsidste knude

Hvad er $v.d$ efter $\text{Relax}(b, v, w)$?
(de aktuelle d værdier er angivet ved $[d]$)

- a) -2
- b) -1
- c) 0
- d) 2
-  e) 4
- f) 5
- g) Ved ikke



Bellman-Ford:

Korteste Veje i Grafer med Negative Vægte

BELLMAN-FORD(G, w, s)

1 INITIALIZE-SINGLE-SOURCE(G, s)

2 **for** $i = 1$ **to** $|G.V| - 1$

3 **for** each edge $(u, v) \in G.E$

4 RELAX(u, v, w)

5 **for** each edge $(u, v) \in G.E$

6 **if** $v.d > u.d + w(u, v)$

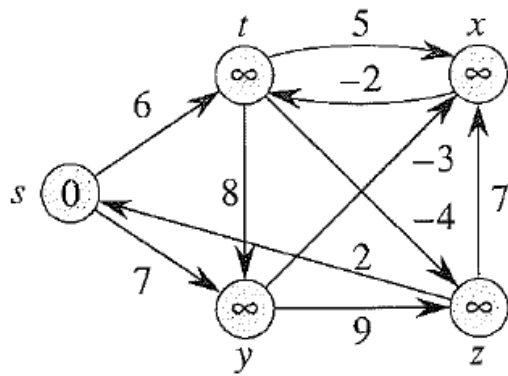
7 **return** FALSE

8 **return** TRUE

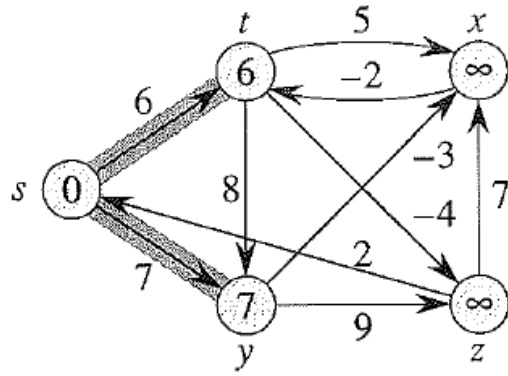
Check for
negativ
cykel

Tid $O(nm)$

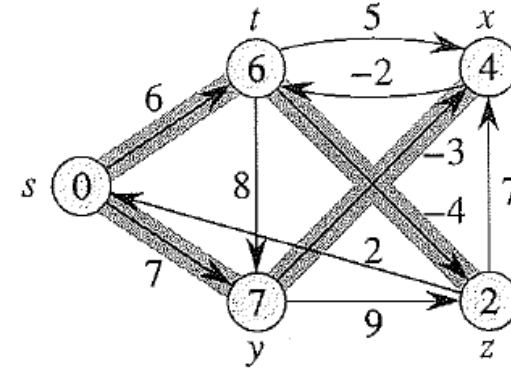
Bellman-Ford: Eksempel



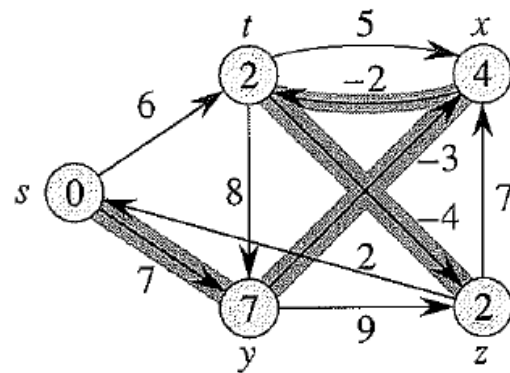
(a)



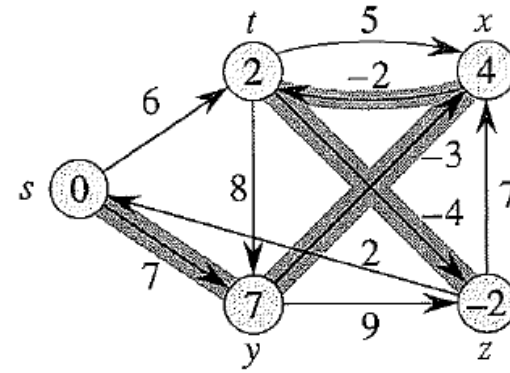
(b)



(c)



(d)



(e)

Antag der findes en **negative cykel**

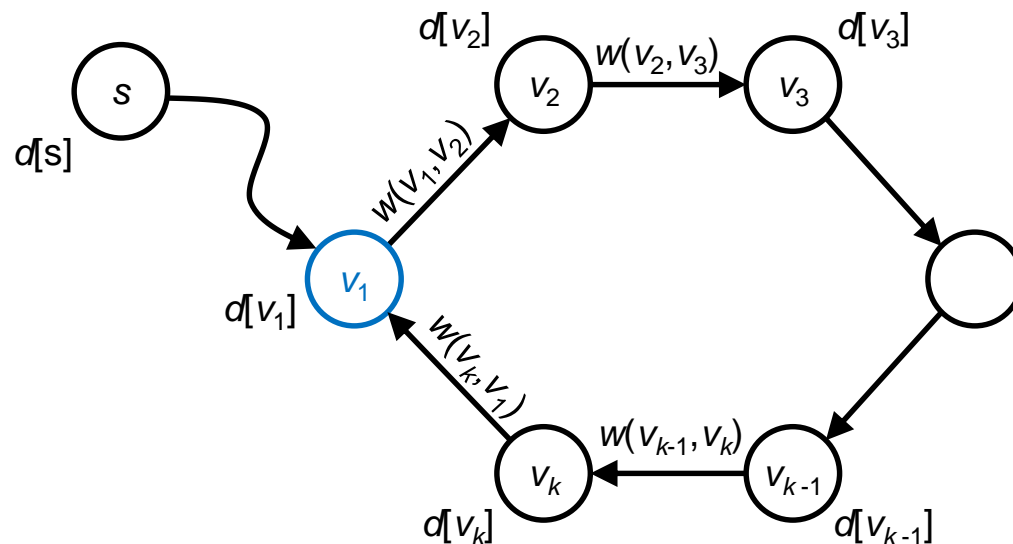
$$w(v_1, v_2) + w(v_2, v_3) + \dots + w(v_{k-1}, v_k) + w(v_k, v_1) < 0$$

og **ikke** muligt at lave **RELAX**

$$d[v_{i+1}] \leq d[v_i] + w(v_i, v_{i+1})$$

heraf følger **modstriden** (hvis ikke alle $d[v_i] = \infty$)

$$\begin{aligned} d[v_1] &\leq d[v_k] + w(v_k, v_1) \leq (d[v_{k-1}] + w(v_{k-1}, v_k)) + w(v_k, v_1) \leq \\ &\dots \leq d[v_1] + w(v_1, v_2) + w(v_2, v_3) + \dots + w(v_{k-1}, v_k) + w(v_k, v_1) < d[v_1] \end{aligned}$$



Sætning

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

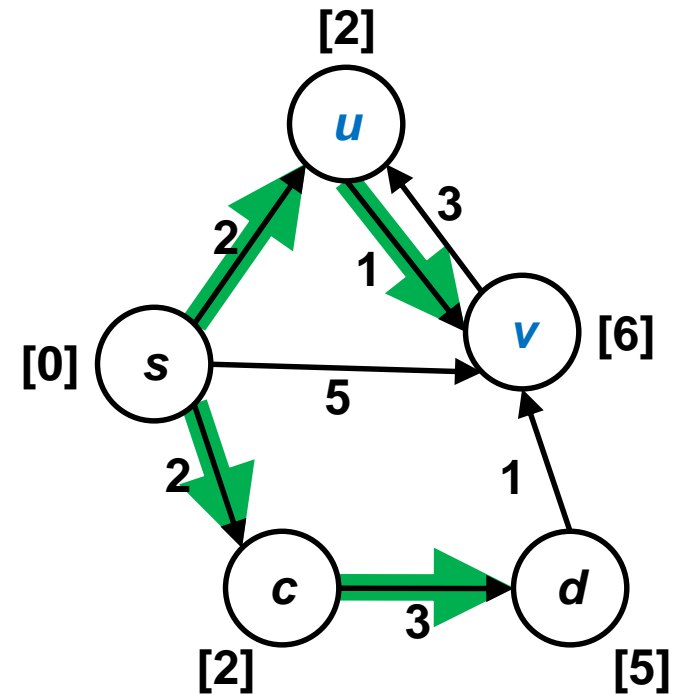
Antag den aktuelle $d[u]$ er den korteste afstand til u .

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



Korollar

Bellman-Ford finder de rigtige korteste afstande efter $n - 1$ $RELAX$ iterationer.

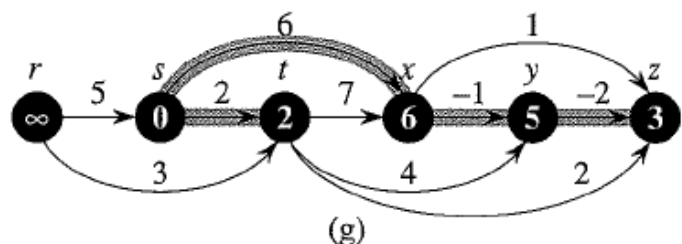
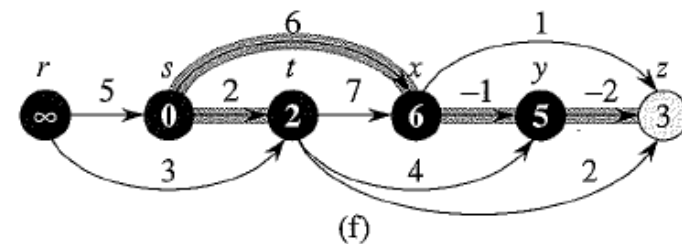
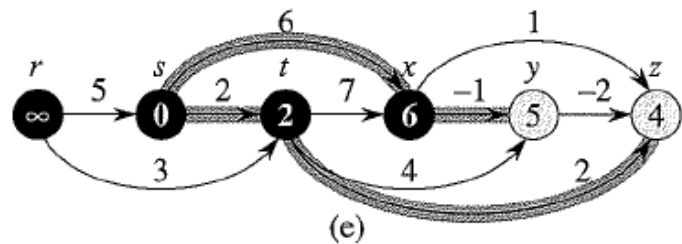
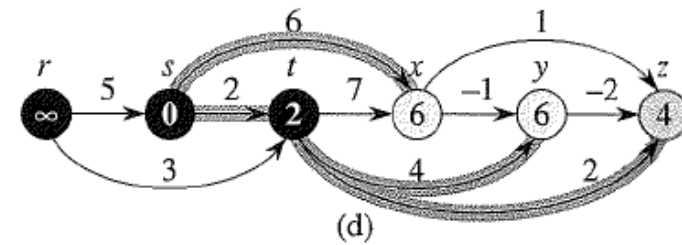
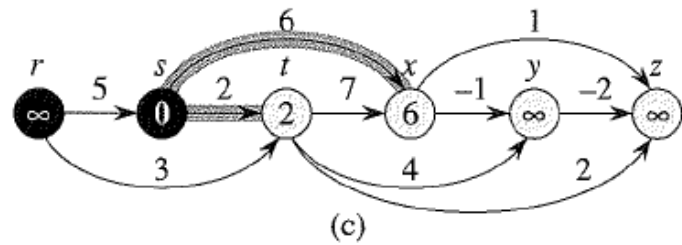
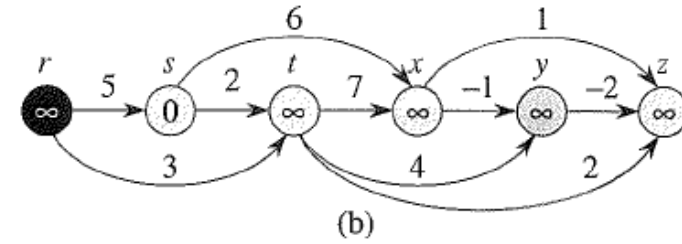
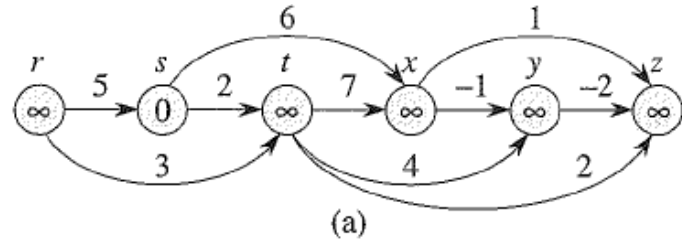


Korteste Veje i Acycliske Grafer

DAG-SHORTEST-PATHS(G, w, s)

- 1 topologically sort the vertices of G
- 2 INITIALIZE-SINGLE-SOURCE(G, s)
- 3 **for** each vertex u , taken in topologically sorted order
- 4 **for** each vertex $v \in G.Adj[u]$
- 5 RELAX(u, v, w)

Acykliske Grafer : Eksempel

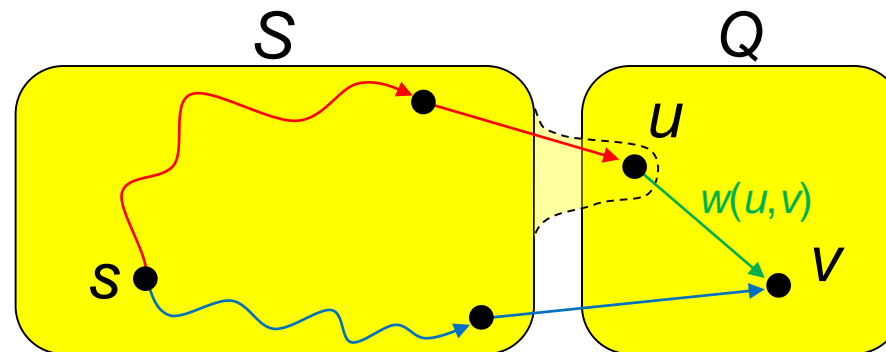


Dijkstra:

Korteste Veje i Grafer uden Negative Vægte

Invarianter

- i) $d[v]$ = korteste afstand fra s til v via knuder i S
- ii) $\forall p \in S, q \in Q : d[p] \leq d[q]$



DIJKSTRA(G, w, s)

1 INITIALIZE-SINGLE-SOURCE(G, s)

2 $S = \emptyset$

3 $Q = G.V$ **Q = prioritets kø, prioritet = d
(besøger knuderne efter stigende afstand fra s)**

4 **while** $Q \neq \emptyset$

5 $u = \text{EXTRACT-MIN}(Q)$

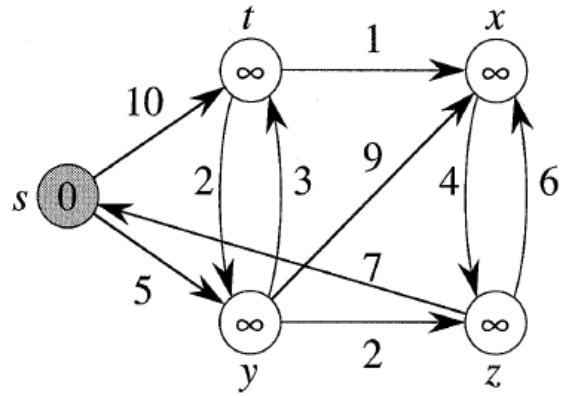
6 $S = S \cup \{u\}$

7 **for** each vertex $v \in G.Adj[u]$

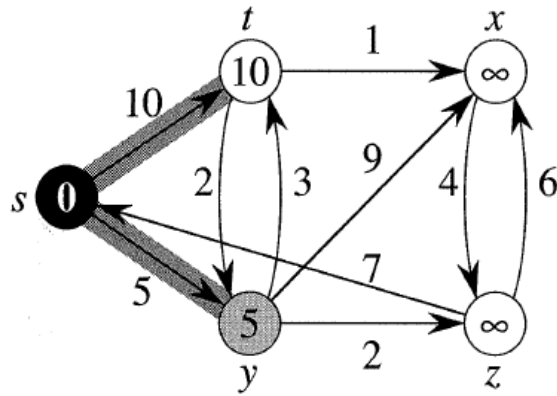
8 RELAX(u, v, w)

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

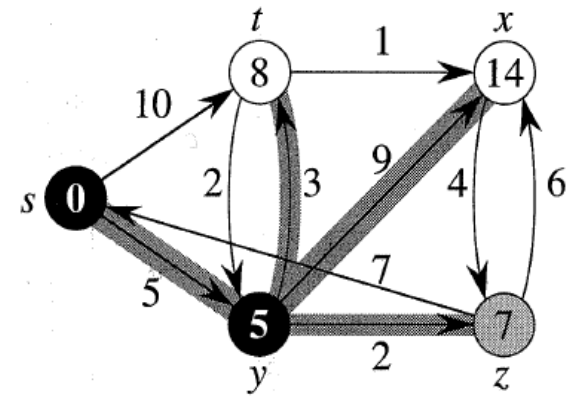
Dijkstra : Eksempel



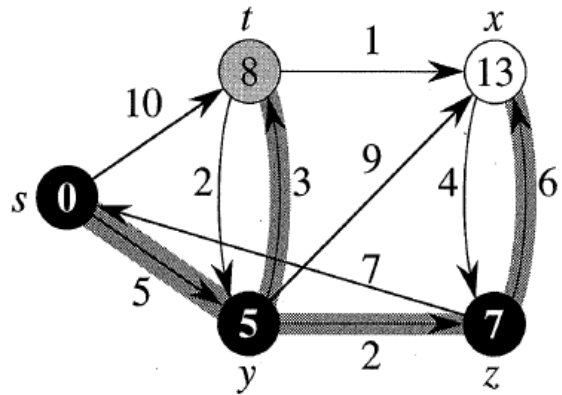
(a)



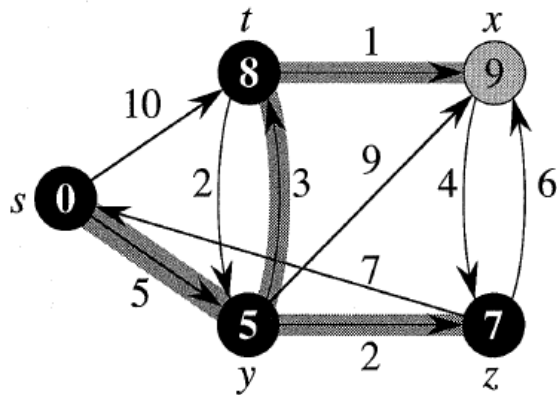
(b)



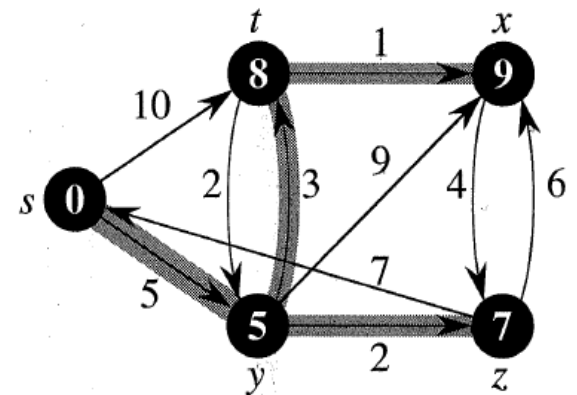
(c)



(d)



(e)



(f)

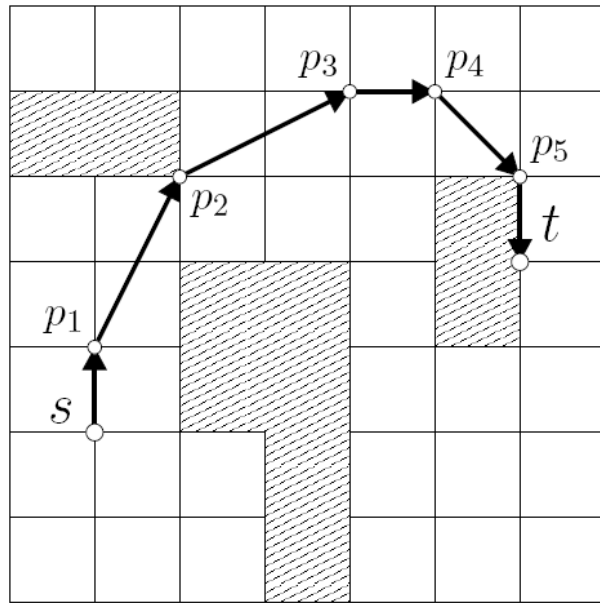
Opsummering

		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 $O(m \cdot n)$

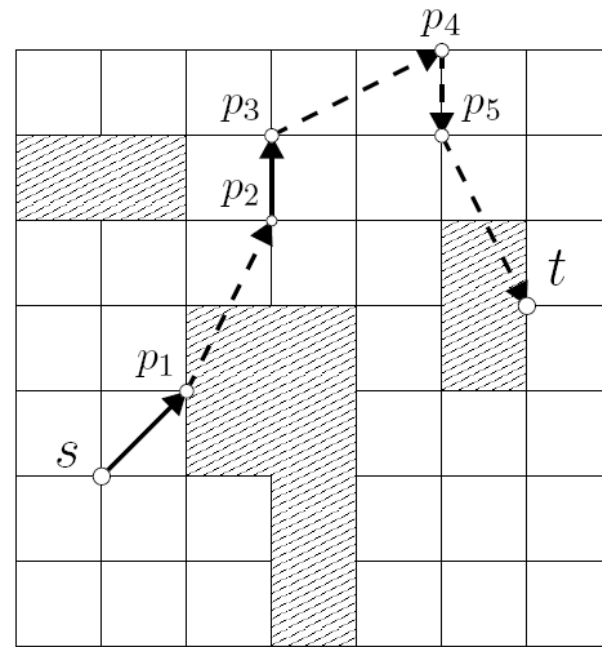
Relaxer
hver kant
præcis
én gang

Vektorrace

(find hurtigste vej fra s til t)



Lovlig sti



Ulovlig sti
(ulovlige stykker af stien er stiplet)

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 højst 3^2 knuder (i', j', x', y') der opfylder de to første krav. Dvs. der er højst $O(n^4)$ potentielle kanter hvor det sidste krav kræver at vi for hver kant checker om $\leq 2n$ krydsende celler er frie. 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 deceleration har vi at den maksimale hastighed k i x og y -retningen skal opfylde $1 + 2 + \dots + k \leq n$, 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 $O(n^3 \sqrt{n})$.