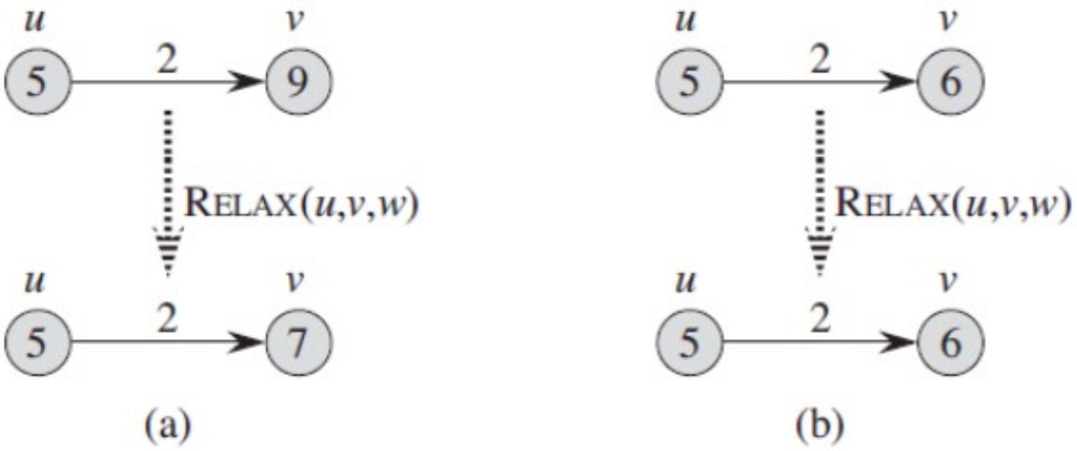
SHORTEST PATHS: BELLMAN FORD AND DAGS

CS340

Relaxing an edge

- Can we improve the shortest-path estimate for v by going through u and taking (u,v)?



Relaxing an edge

```
RELAX(u, v, w)
if v.d > u.d + w(u, v)
    v.d = u.d + w(u, v)
    v.π = u
```

Properties of shortest paths and relaxation

- Path-relaxation property
  - If $p = <v_0, v_1,\ldots,v_k>$ i is a shortest path from $s = v_0$ to v_k, and we relax the edges of p in the order $(v_0, v_1), (v_1,v_2),\ldots(v_{k-1},v_k)$. then $v_k.d = \delta(s, v_k)$. This property holds **regardless** of any other relaxation steps that occur, even if they are intermixed with relaxations of the edges of p.

Bellman-Ford Algorithm

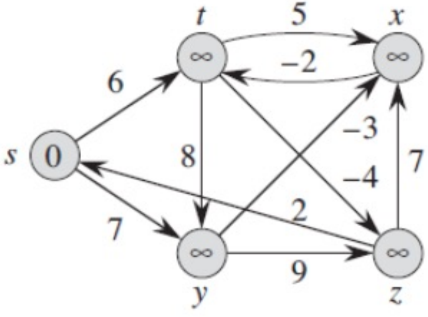
- Allows negative-weight edges.
- Computes v.d and π.d for all v ∈ V.
- Returns TRUE if no negative-weight cycles reachable from s, FALSE otherwise.

Bellman-Ford Algorithm

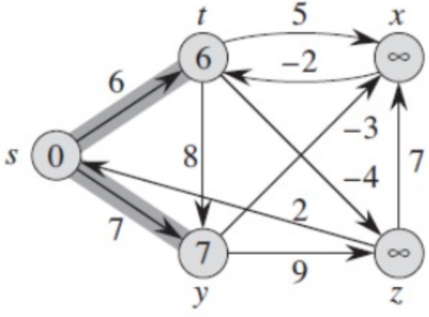
- Relies on the **Path Relaxation Property**
- Relaxes each edge |V|-1 times (why?)
- Relaxes each edge 1 more time to detect negative-weight cycles. Returns true if none.
- Time Complexity:
  nested for loop
  outer loop = |v|-1
  inner loop = E
  Total = Θ(VE)

```
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
```

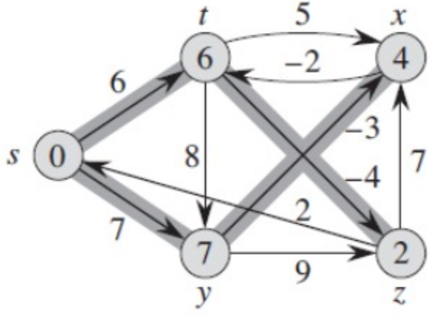
Bellman-Ford



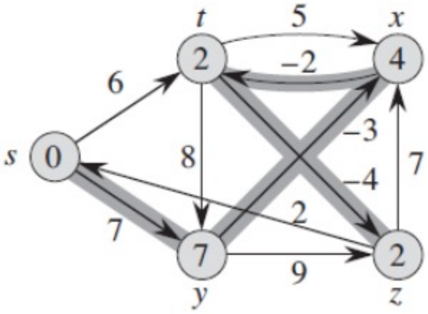
(a)



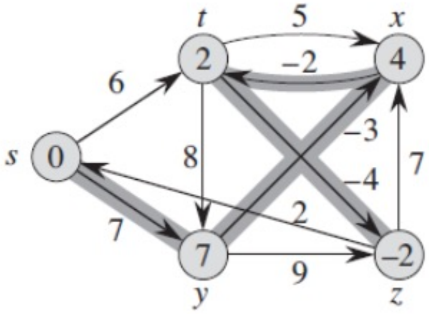
(b)



(c)



(d)



(e)

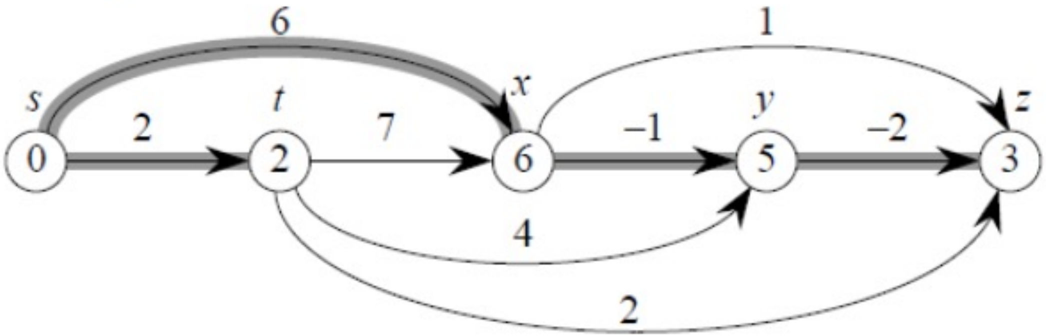
Edges relaxed in this order:
(t,x) (t,y) (t,z) (x,t) (y,x) (y,z)
(z,x) (z,s) (s,t) (s,y)

Single-source shortest paths in a dag

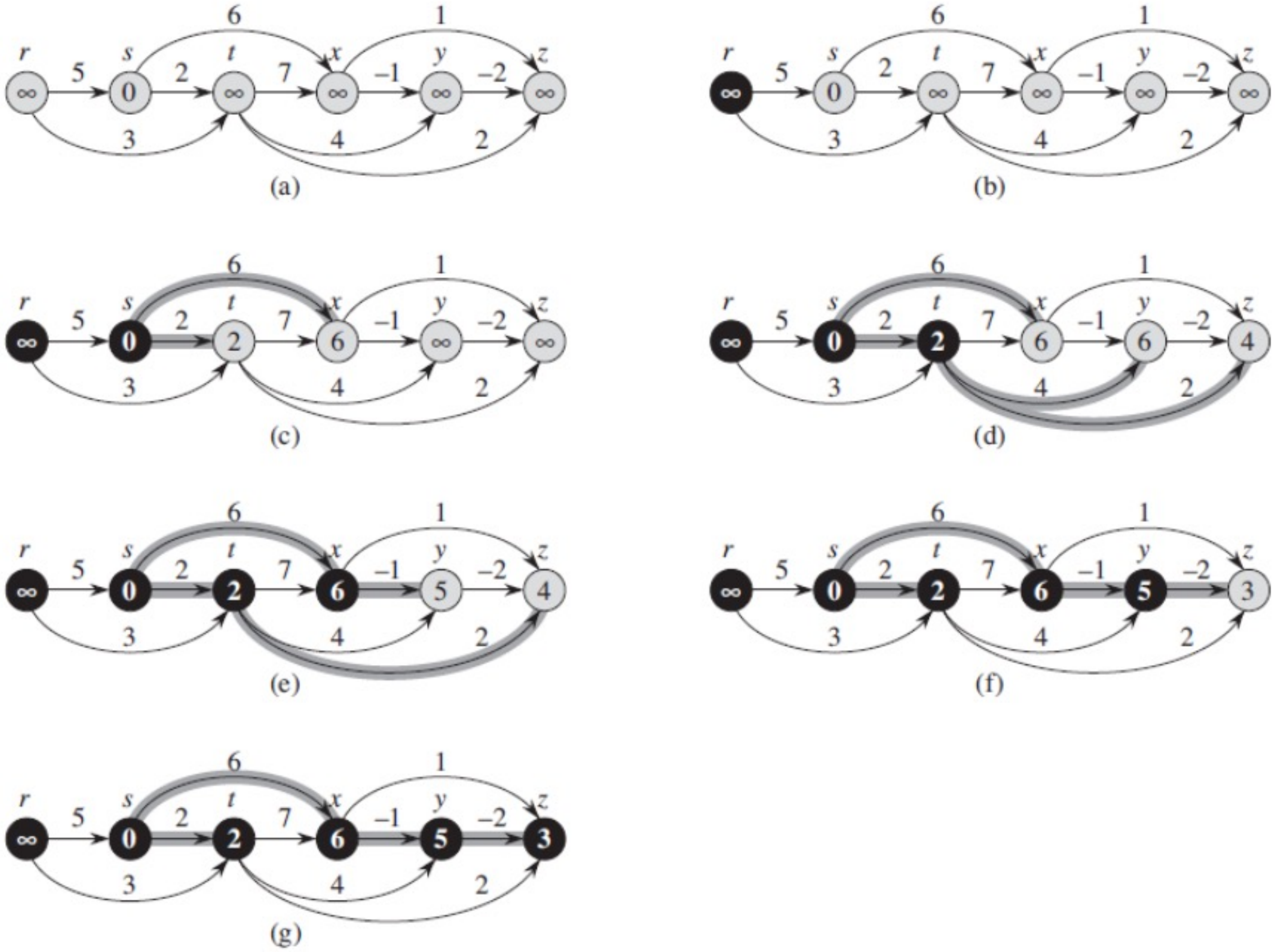
```
DAG-SHORTEST-PATHS(G, w, s)
topologically sort the vertices of G
INITIALIZE-SINGLE-SOURCE(G, s)
for each vertex u, taken in topologically sorted order
    for each vertex v ∈ G.Adj[u]
        RELAX(u, v, w)
```

Time complexity Θ(V+E)

Single-source shortest paths in a dag



Single-source shortest paths in a dag



(a) (b) (c) (d) (e) (f) (g)