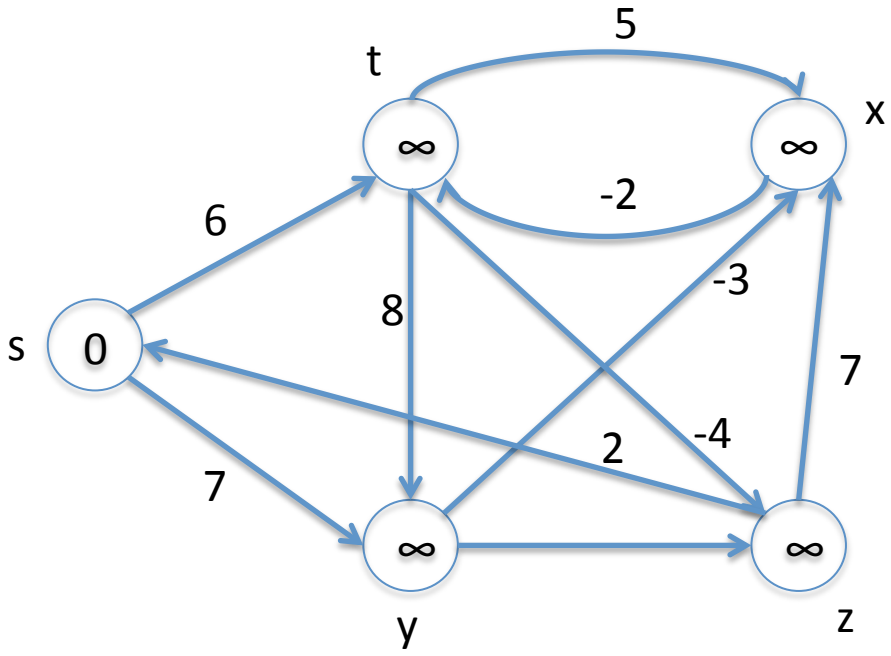
Bellman Ford Algorithm

- How the Bellman Ford algorithm works
- Example from the book, figure 24.4
- Step by step
- v.d is shown in the vertices and shaded edges show predecessors

Initialization

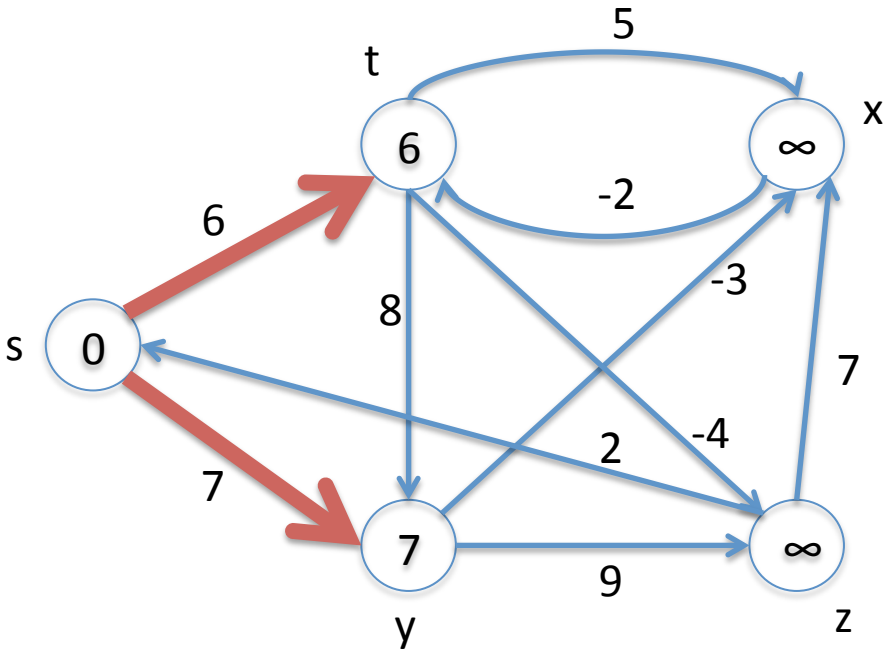



```
BELLMAN-FORD(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  for i = 1 to |G.V| − 1
3      for each edge (u, v) ∈ G.E
4          RELAX(u, v, w)
5  for each edge (u, v) ∈ G.E
6      if v.d > u.d + w(u, v)
7          return FALSE
8  return TRUE
```

Edges considered in this order:

(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)

First pass of the 1st nested loop:



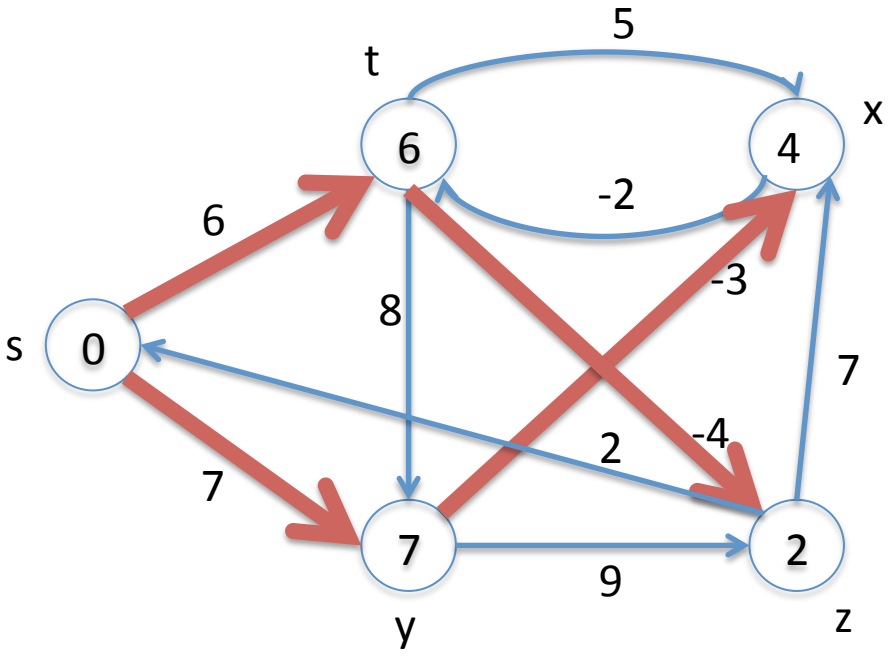
We relax:
(s,t)
(s,y)

The others don't relax because of the infinite values

Edges considered in this order:

(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)

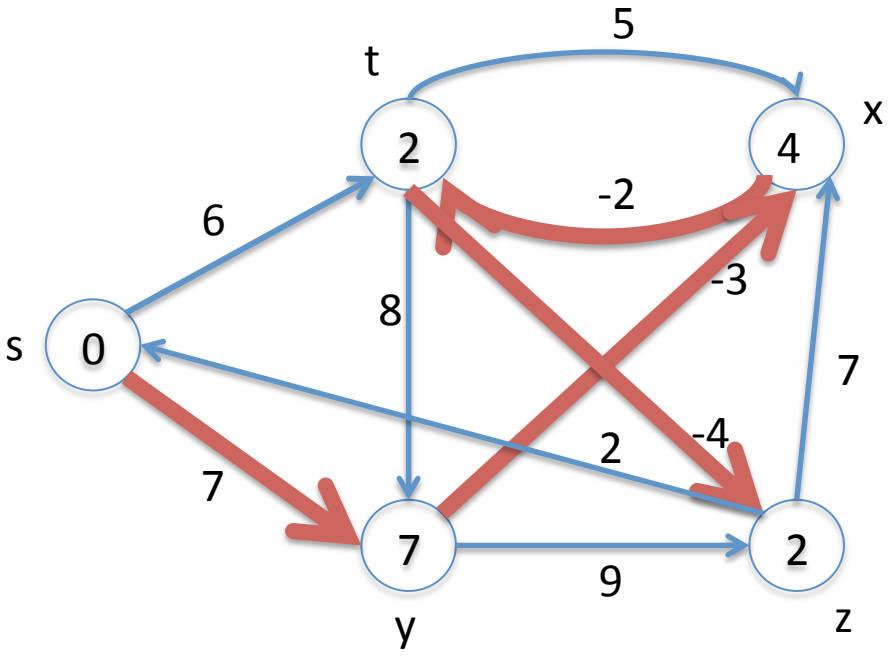
Second pass of the 1st nested loop:



We relax:
(t, x), x-11
(t, y),
(t, z), z-2
(y, x), x-4
(y, z),
(z, x),
(z, s),
(s, t),
(s, y),

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

Third pass of the 1st nested loop:

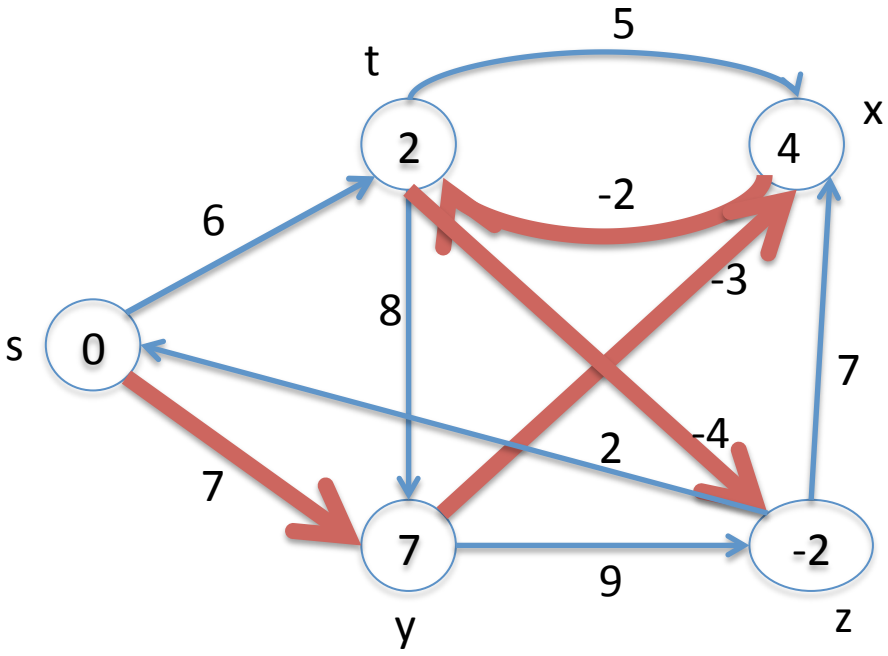


We relax:
(t, x),
(t, y),
(t, z),
(x, t), t-2
(y, x),
(y, z),
(z, x),
(z, s),
(s, t),
(s, y),

Edges considered in this order:

(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)

Fourth pass of the 1st nested loop:

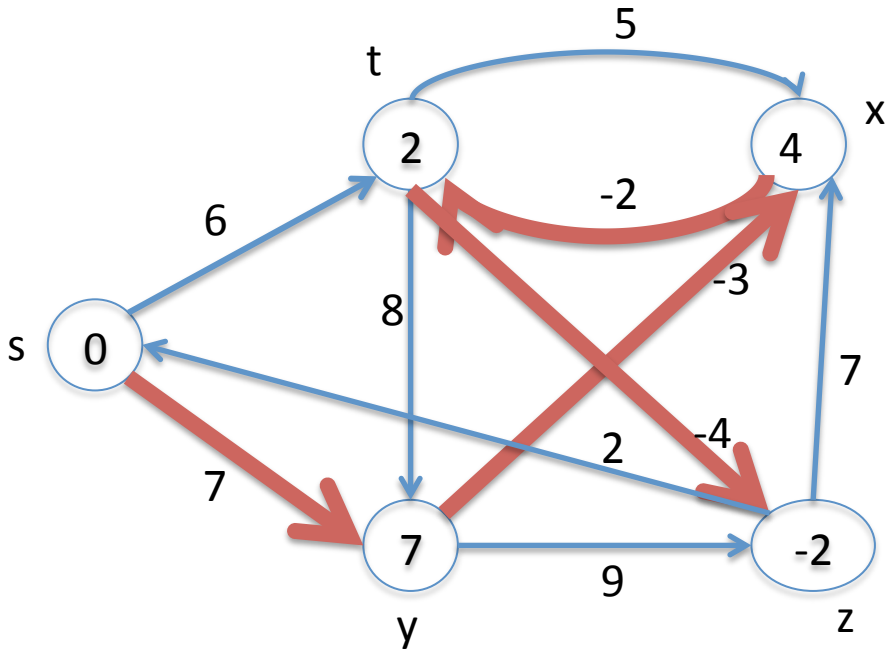


We relax:
(t, x),
(t, y),
(t, z), z- (-2)
(x, t),
(y, x),
(y, z),
(z, x),
(z, s),
(s, t),
(s, y),

Edges considered in this order:

(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)

Second for loop



In the last verification, no comparison makes us return false

```
5   for each edge (u, v) ∈ G.E
6       if v.d > u.d + w(u, v)
7           return FALSE
```

Edges considered in this order:

(t, x), (t, y), (t, z), (x, t), (y, x), (y, z), (z, x), (z, s), (s, t), (s, y)