Bellman-Ford Single source shortest path algorithm

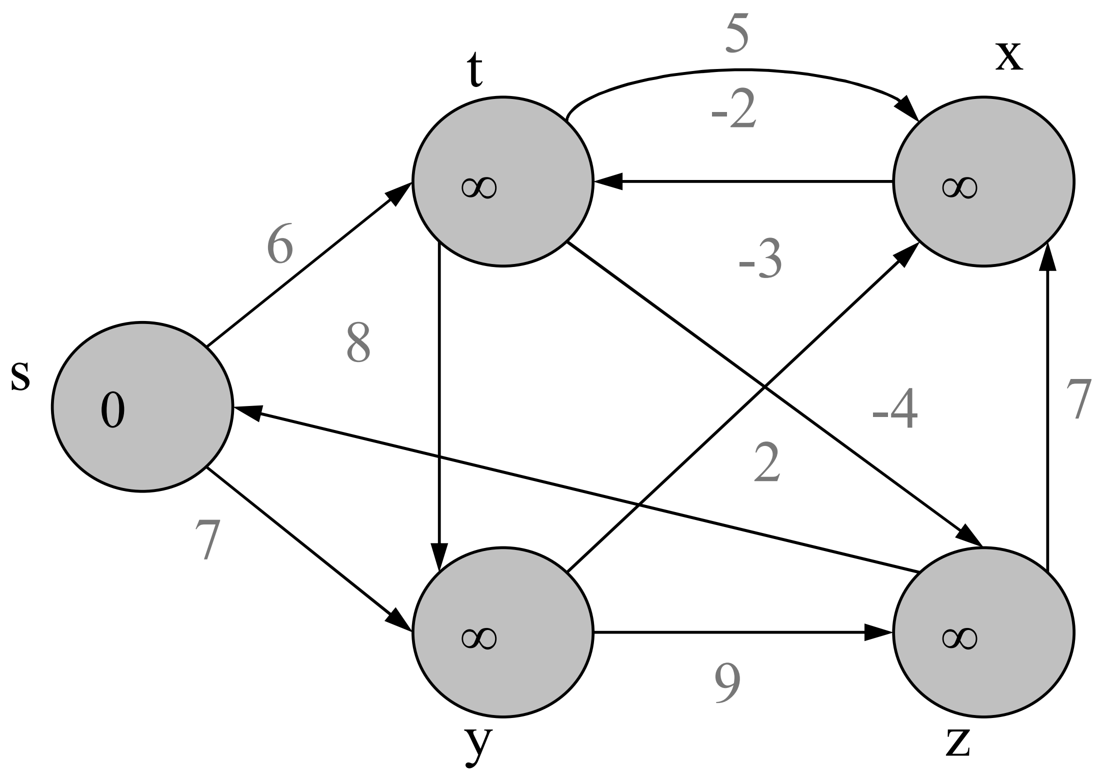
Bellman-Ford Algorithm for Single Source Shortest Paths

- More general than Dijkstra's algorithm:
  - Edge-weights can be negative

- Detects the existence of negative-weight cycle(s) reachable from source S.

- Follows Dynamic programming strategy (Try out all possible solutions and pickup the best solution)

Bellman-Ford Algorithm Example



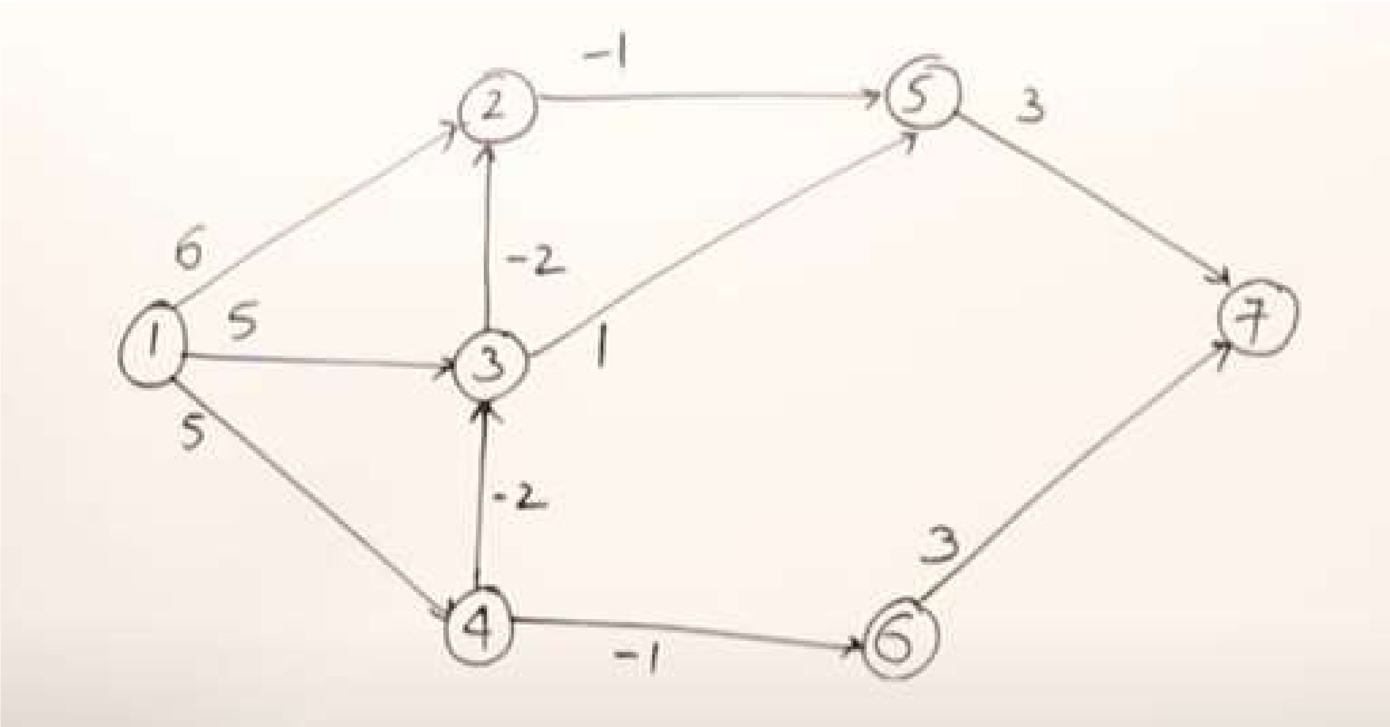
Procedure

- The Bellman ford algorithm starts with computing the path from source to each node i in *v*.
- Since there are n nodes in a graph, the path to any node can contain at most n-1 nodes.
- Then for each node u, remaining n-2 nodes are required to be examined.
- If <j, u>€E then the condition dist[u] is compared with dist[j] + cost[j][u].
- If dist[u] is greater than dist[j]+dist[j][u] then dist[u] is updated to dist[j]+cost[j][u].

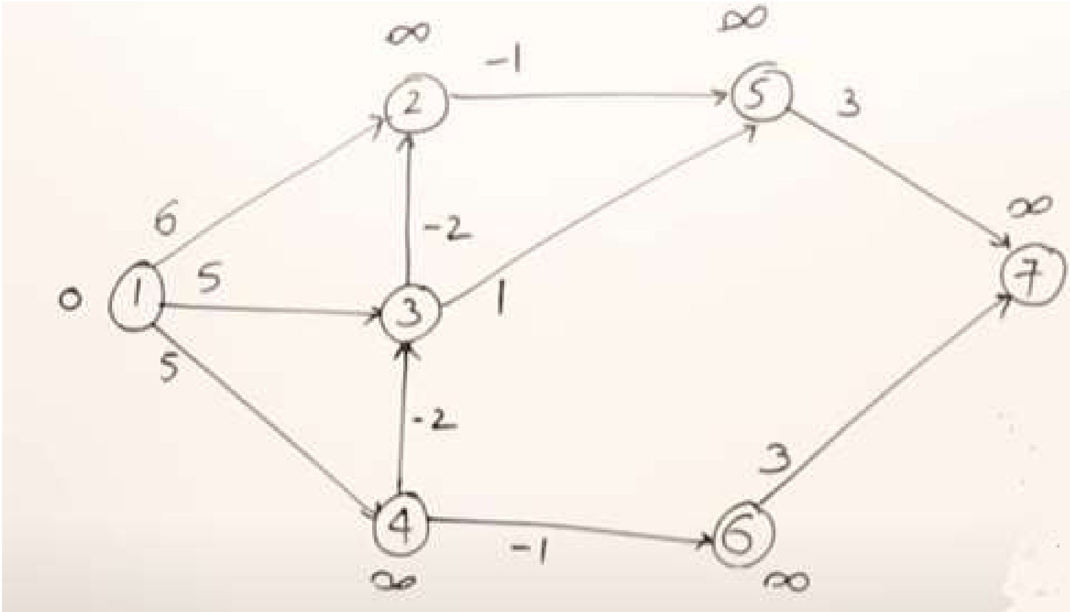
- Idea:
  - Each edge is relaxed |V–1| times by making |V-1| passes over the whole edge set.
  - To make sure that each edge is relaxed exactly |V – 1| times, it puts the edges in an unordered list and goes over the list |V – 1| times.

Relaxation:

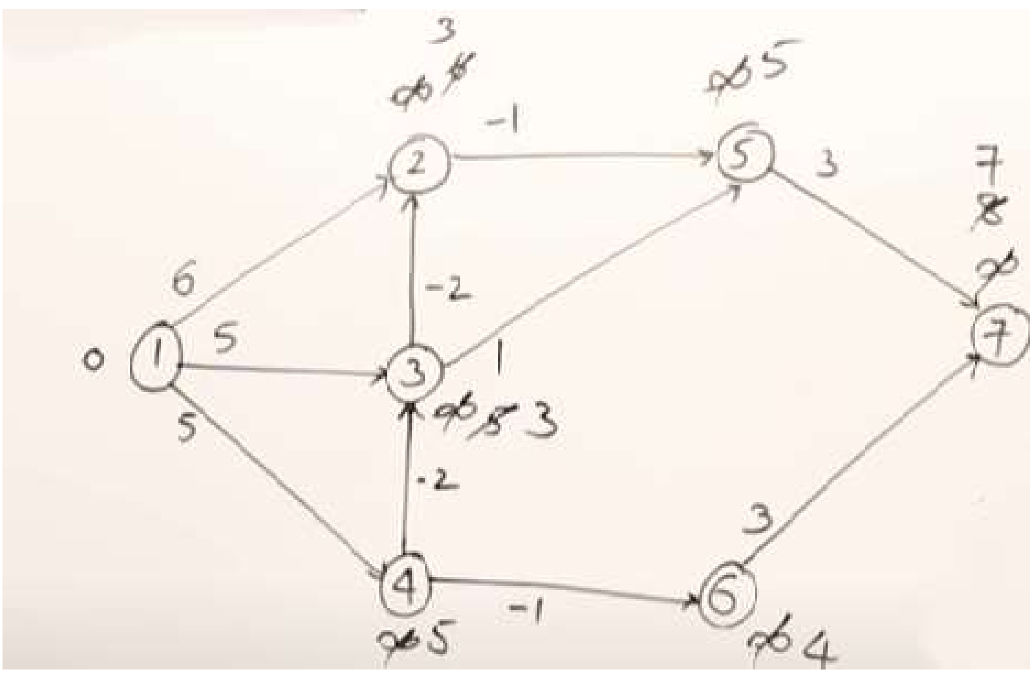
$$
\begin{aligned}
&\textbf{for } \text{all edges } (u,v) \in E \\
&\qquad\qquad \textbf{if } dist[v] > dist[u] + w(u,v) \\
&\qquad\qquad\qquad dist[v] \leftarrow dist[u] + w(u,v)
\end{aligned}
$$



List of Edges:
(1,2), (1,3), (1,4), (2,5), (3,2), (3,5), (4,3), (5,7), (6,7)



Initiation

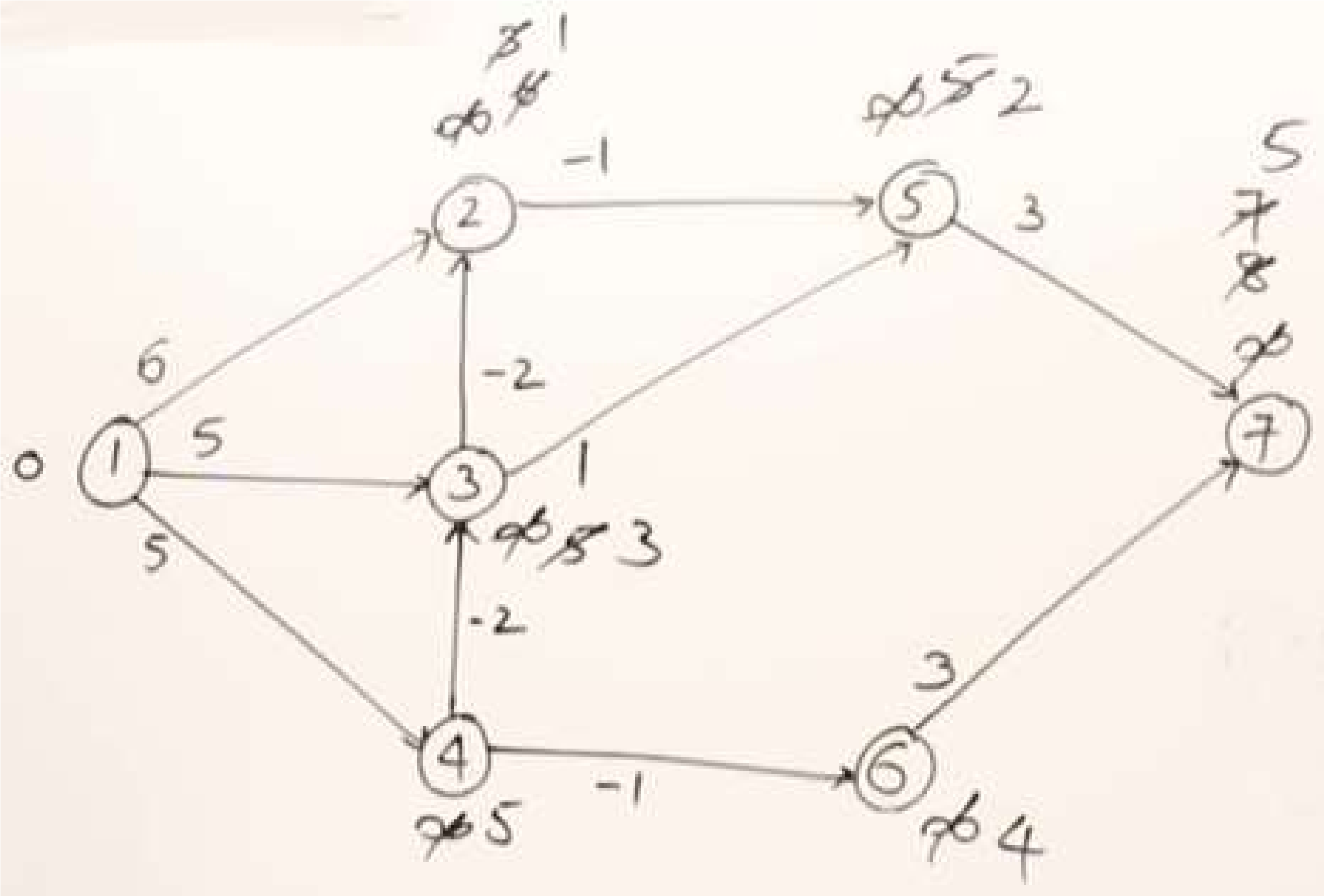


Iteration – 1

List of Edges:
(1,2), (1,3), (1,4), (2,5),
(3,2), (3,5), (4,3), (5,7), (6,7)

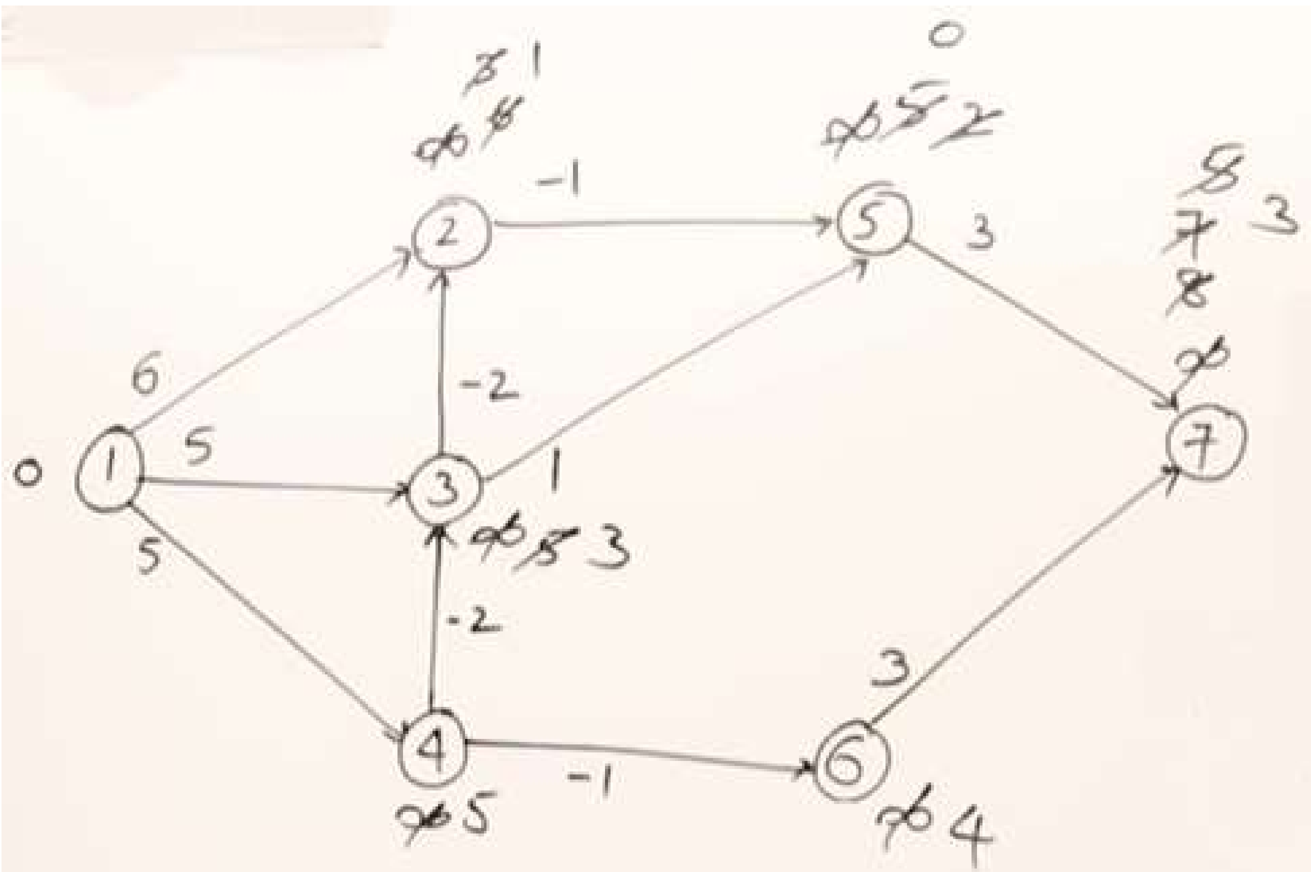
Relaxation:
If (d[u]+c(u,v) < d[v]
d[v]=d[u]+c(u,v)

Iteration - 2
(1,2), (1,3), (1,4), (2,5), (3,2), (3,5), (4,3), (5,7), (6,7)



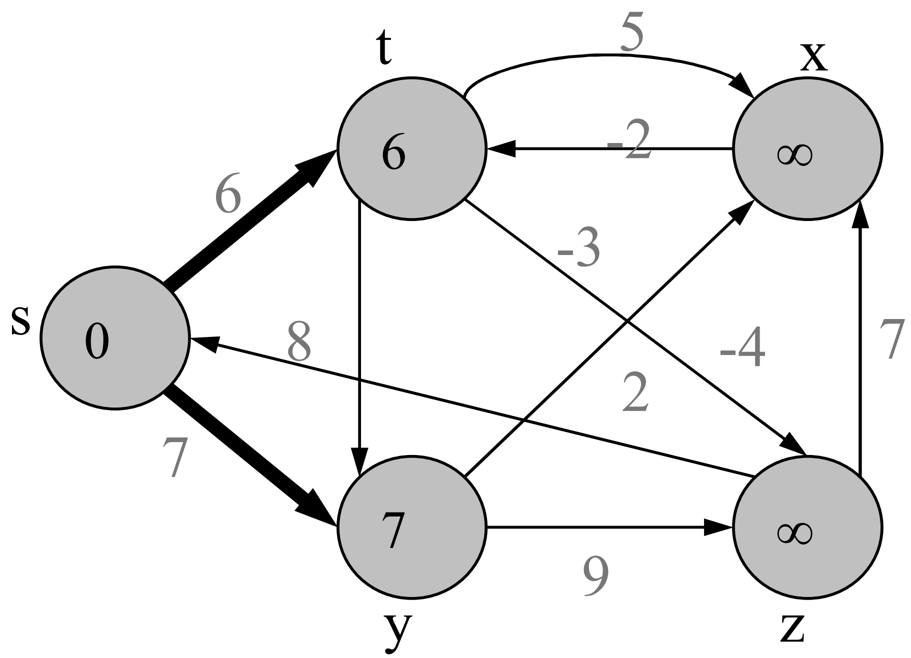
Iteration - 3

(1,2), (1,3), (1,4), (2,5), (3,2), (3,5), (4,3), (5,7), (6,7)



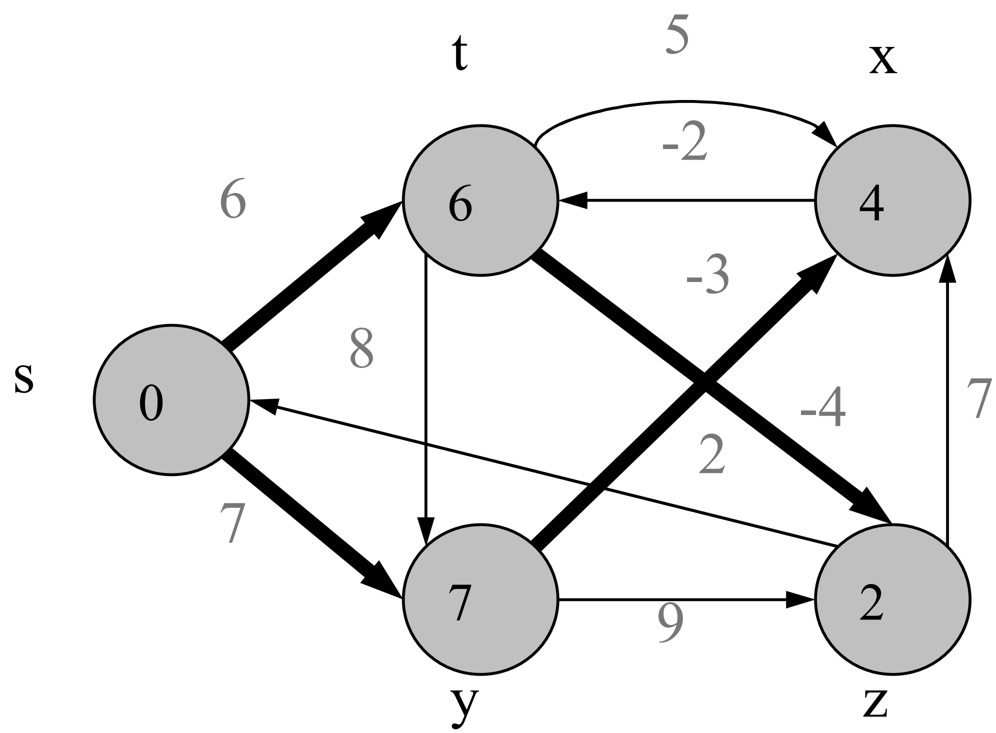
1 - 0
2 - 1
3 - 3
4 - 5
5 - 0
6 - 4
7 - 3

Bellman-Ford Algorithm Example 2



List of Edges:
(s,t), (s,y), (t,x), (t,y), (t,z), (x,t), (y,x), (y,z), (z,x), (z,s)

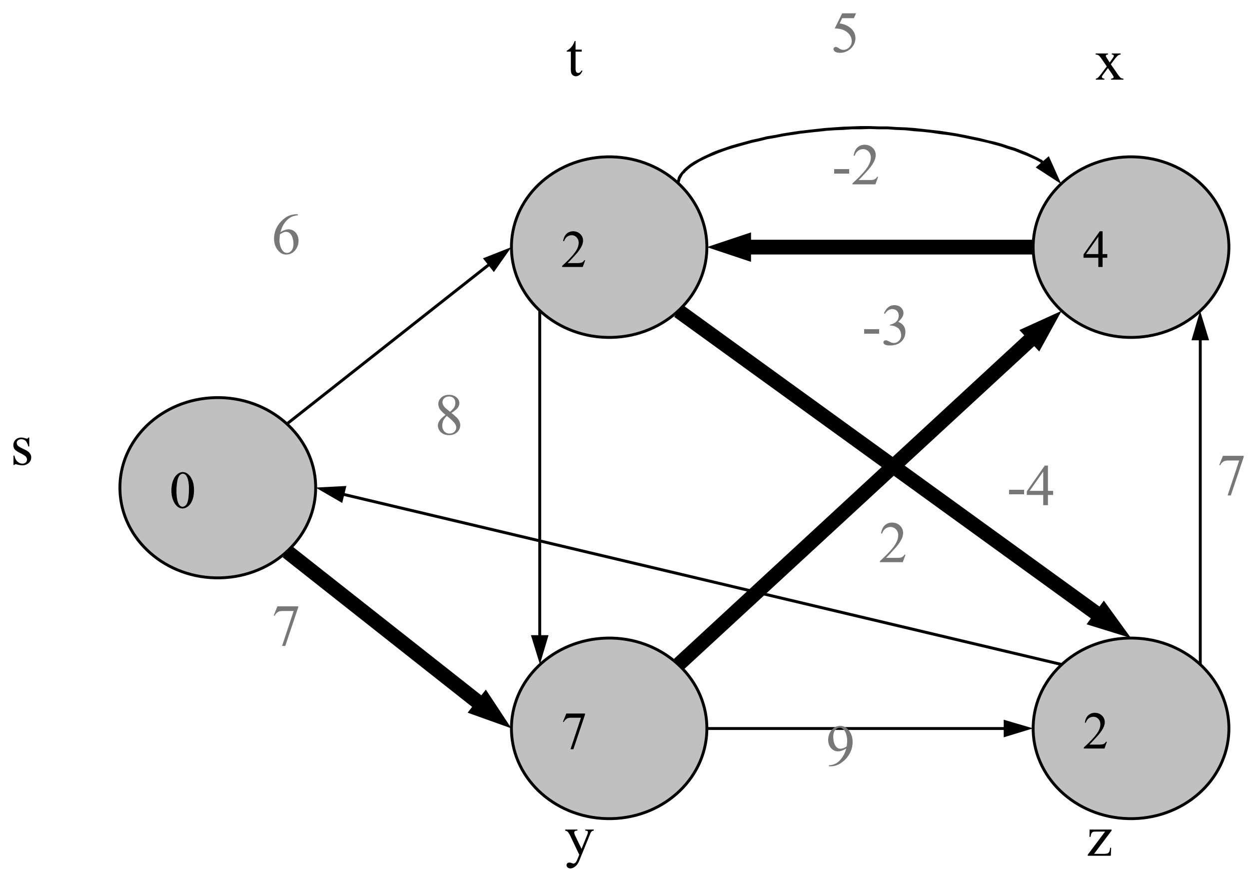
Bellman-Ford Algorithm Example



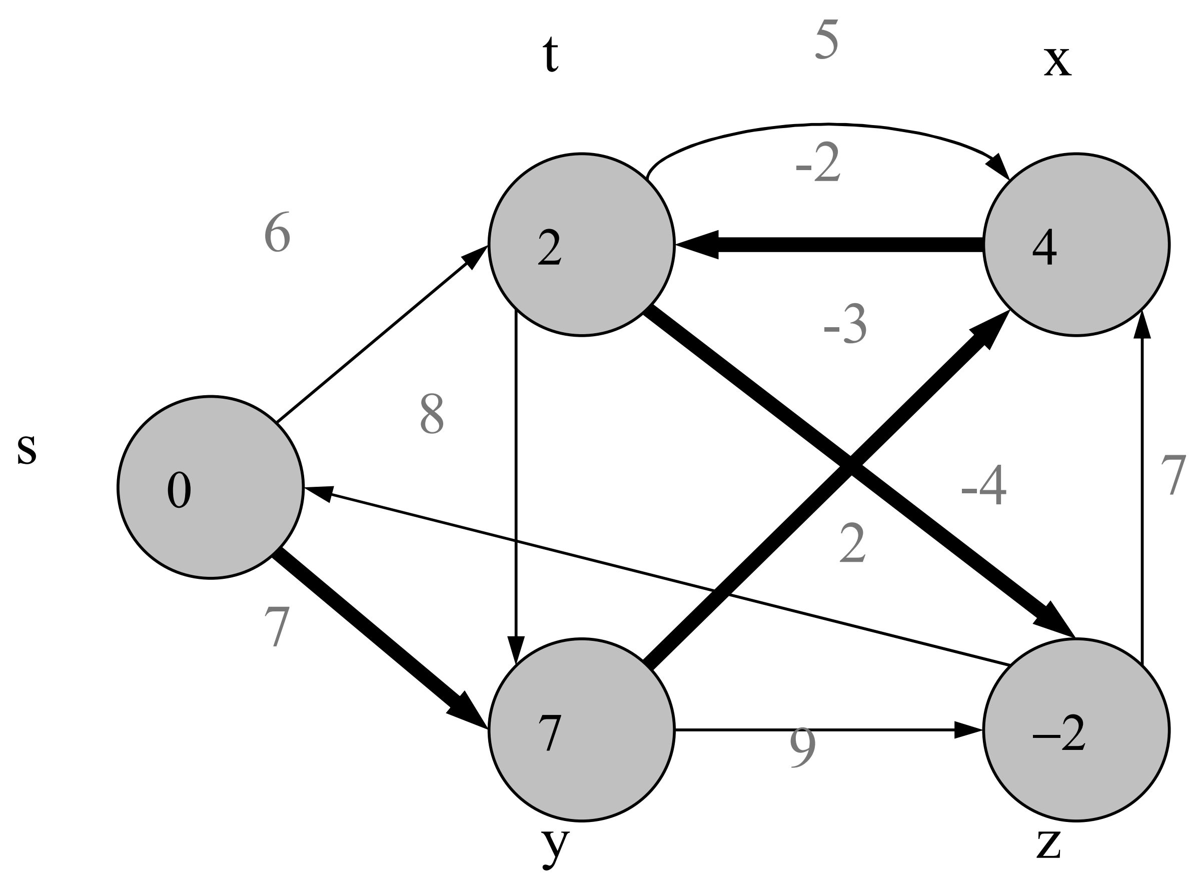
Iteration - 1

List of Edges:
(s,t), (s,y), (t,x), (t,y), (t,z), (x,t), (y,x), (y,z), (z,x), (z,s)

Bellman-Ford Algorithm Example



Bellman-Ford Algorithm Example



$\textsc{Bellman-Ford}(G, s)$

for all $v \in V$
    $dist[v] \leftarrow \infty$
    $prev[v] \leftarrow null$
$dist[s] \leftarrow 0$
for $i \leftarrow 1$ **to** $|V| - 1$
    for all edges $(u, v) \in E$
        if $dist[v] > dist[u] + w(u, v)$
            $dist[v] \leftarrow dist[u] + w(u, v)$
            $prev[v] \leftarrow u$
for all edges $(u, v) \in E$
    if $dist[v] > dist[u] + w(u, v)$
        return $false$

Application

Assembly Line Scheduling

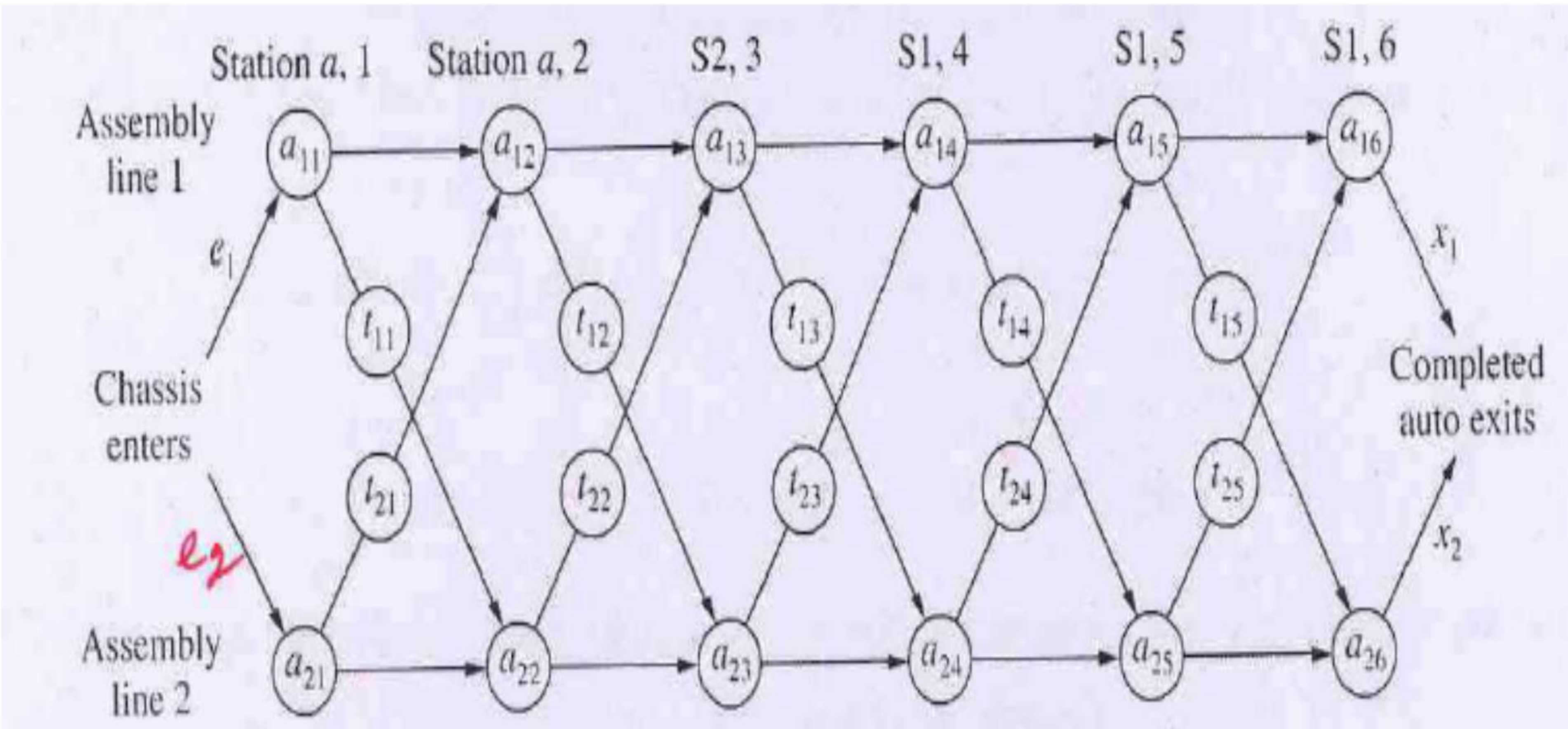
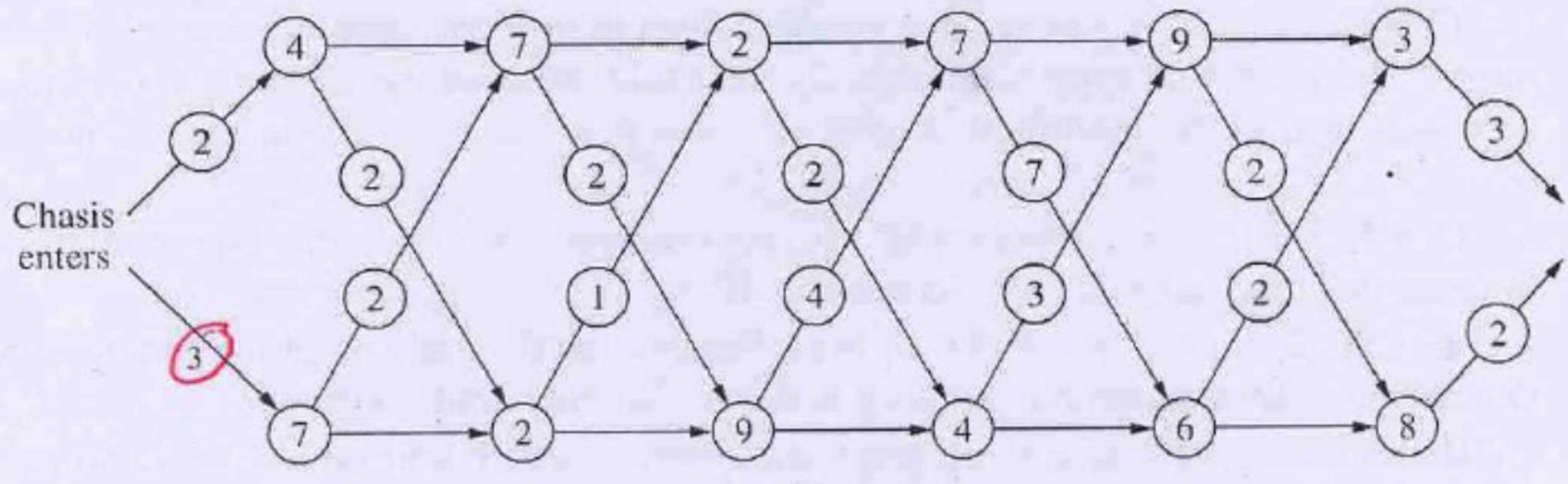


Illustration of Assembly line Scheduling



(a)

j	1	2	3	4	5	6
$f_1[j]$	6	13	13	20	29	30
$f_2[j]$	10	10	19	19	25	33

$f^* = 33$

(b)

j	2	3	4	5	6
$l_1[j]$	1	2	1	1	2
$l_2[j]$	1	2	1	2	2

$l^* = 1$

(c)