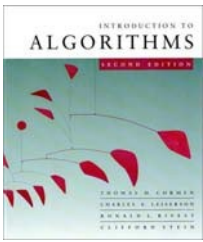# CS60020: Foundations of Algorithm Design and Machine Learning

Sourangshu Bhattacharya

Dijkstra's algorithm

$d[s] \leftarrow 0$
for each $v \in V - \{s\}$
    do $d[v] \leftarrow \infty$
$S \leftarrow \varnothing$
$Q \leftarrow V$ $\triangleright Q$ is a priority queue maintaining $V - S$

while $Q \neq \varnothing$
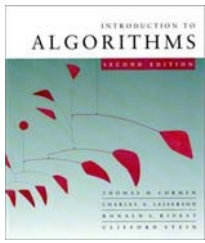   **do** $u \leftarrow \text{EXTRACT-MIN}(Q)$
        $S \leftarrow S \cup \{u\}$
        for each $v \in Adj[u]$
            do if $d[v] > d[u] + w(u, v)$ ***relaxation***
                then $d[v] \leftarrow d[u] + w(u, v)$ ***step***
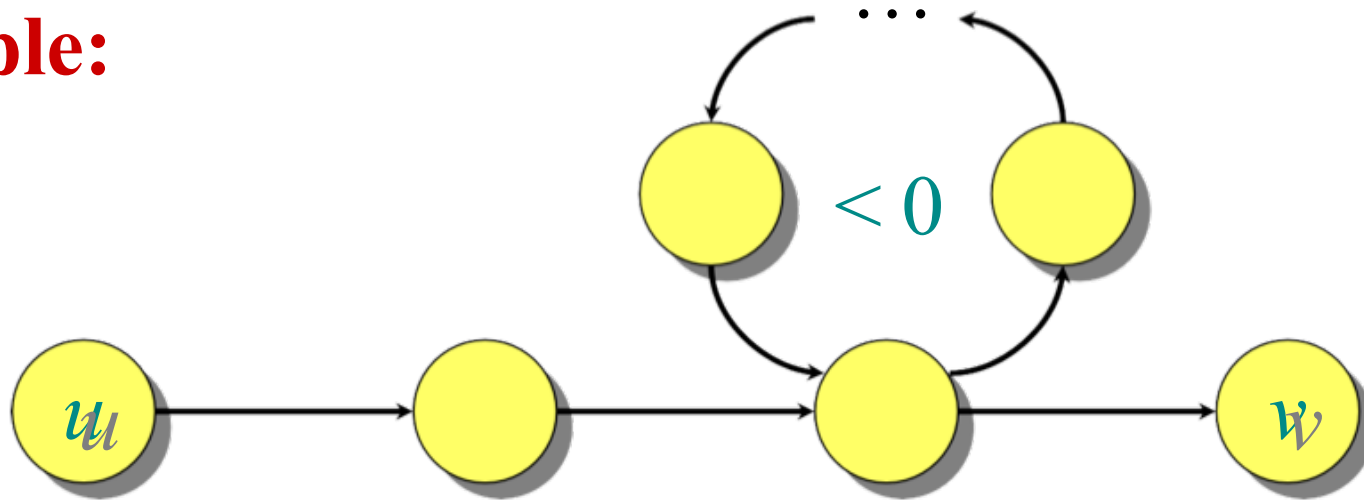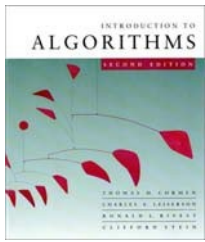
Implicit DECREASE-KEY

Negative-weight cycles

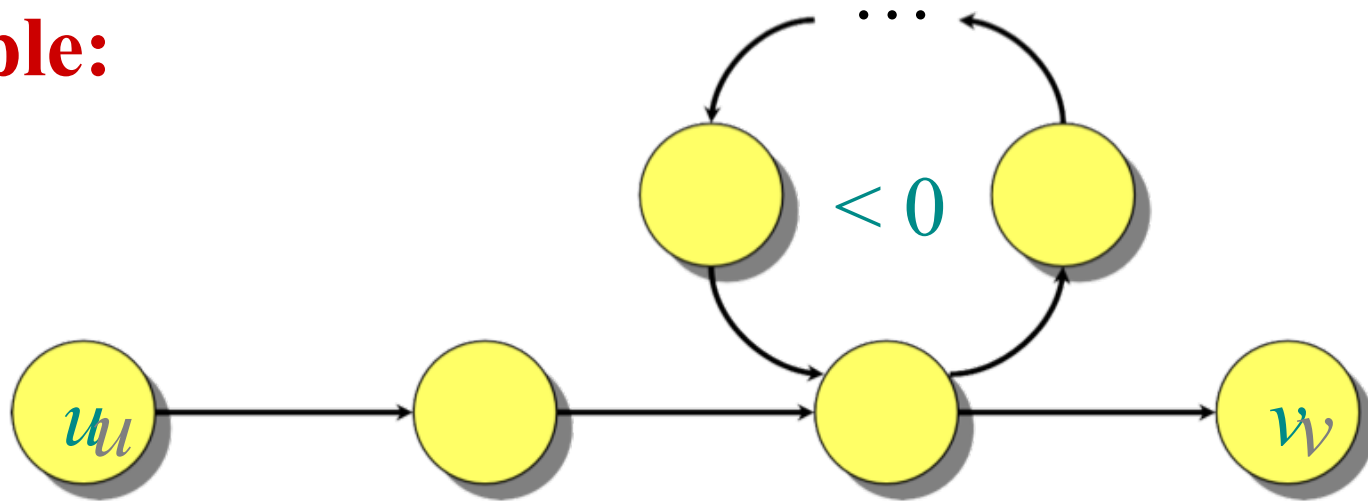**Recall:** If a graph $G = (V, E)$ contains a negative-weight cycle, then some shortest paths may not exist.

Example:

Negative-weight cycles

Recall: If a graph $G = (V, E)$ contains a negative-weight cycle, then some shortest paths may not exist.

Example:

Bellman-Ford algorithm: Finds all shortest-path lengths from a **source** $s \in V$ to all $v \in V$ or determines that a negative-weight cycle exists.
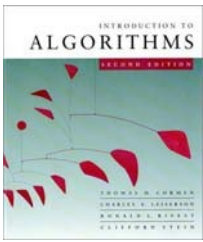
Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson
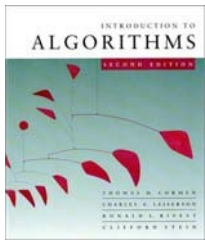
Bellman-Ford algorithm

$d[s] \leftarrow 0$
for each $v \in V - \{s\}$ ⎫ initialization
    do $d[v] \leftarrow \infty$ ⎭

for $i \leftarrow 1$ **to** $|V| - 1$
    do for each edge $(u, v) \in E$
        do if $d[v] > d[u] + w(u, v)$ ⎫ *relaxation*
            then $d[v] \leftarrow d[u] + w(u, v)$ ⎭ *step*

for each edge $(u, v) \in E$
    do if $d[v] > d[u] + w(u, v)$
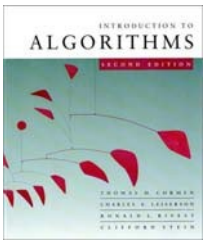   **then** report that a negative-weight cycle exists

At the end, $d[v] = \delta(s, v)$, if no negative-weight cycles.
Time = $O(VE)$.

Example of Bellman-Ford

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson
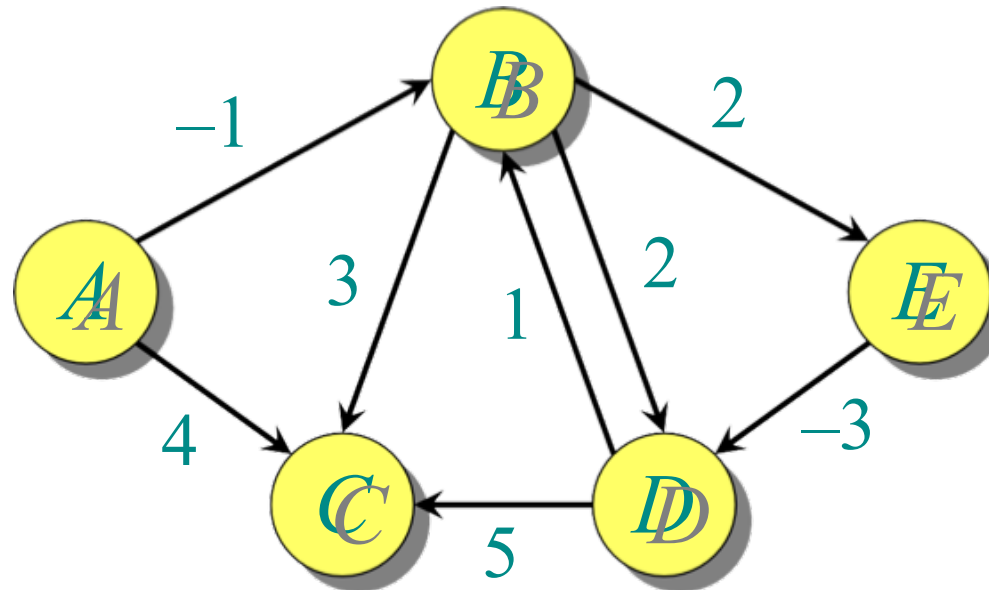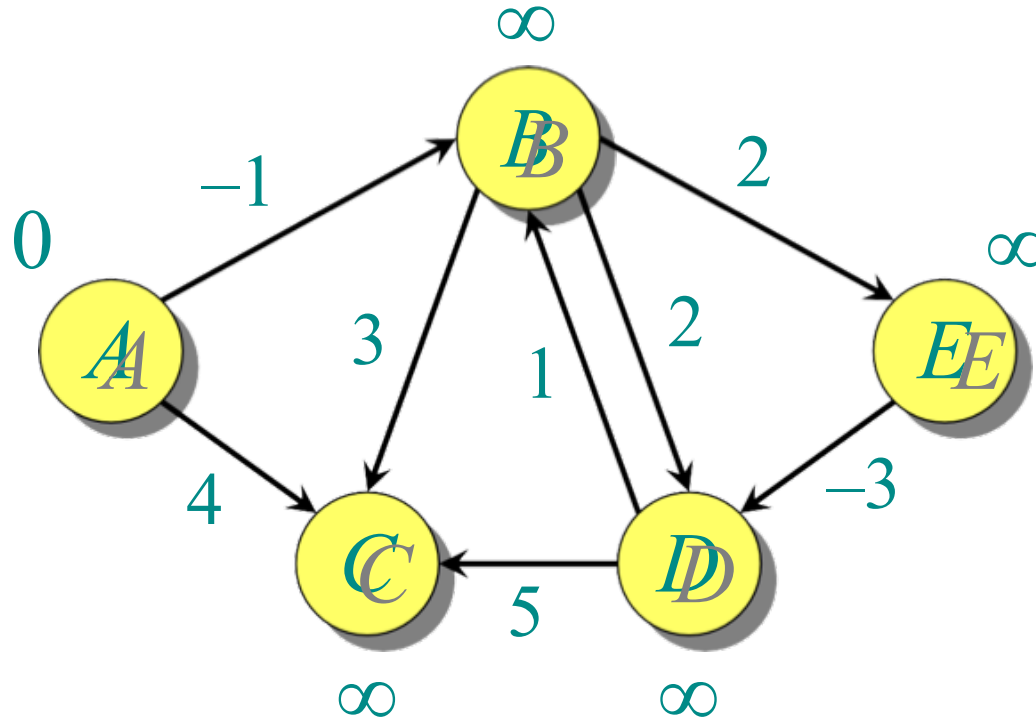
Initialization.
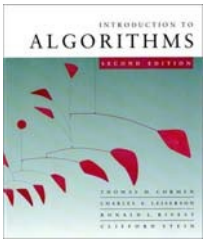
Order of edge relaxation.

Example of Bellman-Ford
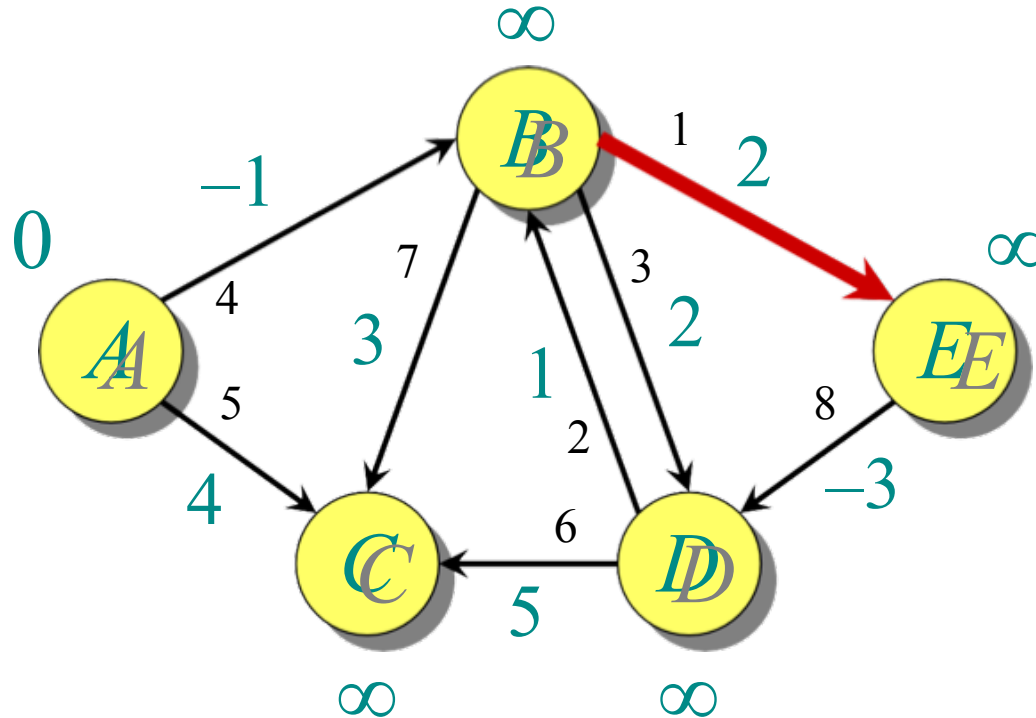
Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson
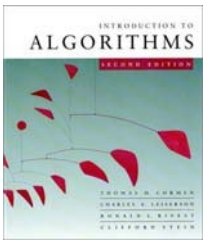
Example of Bellman-Ford

Example of Bellman-Ford

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

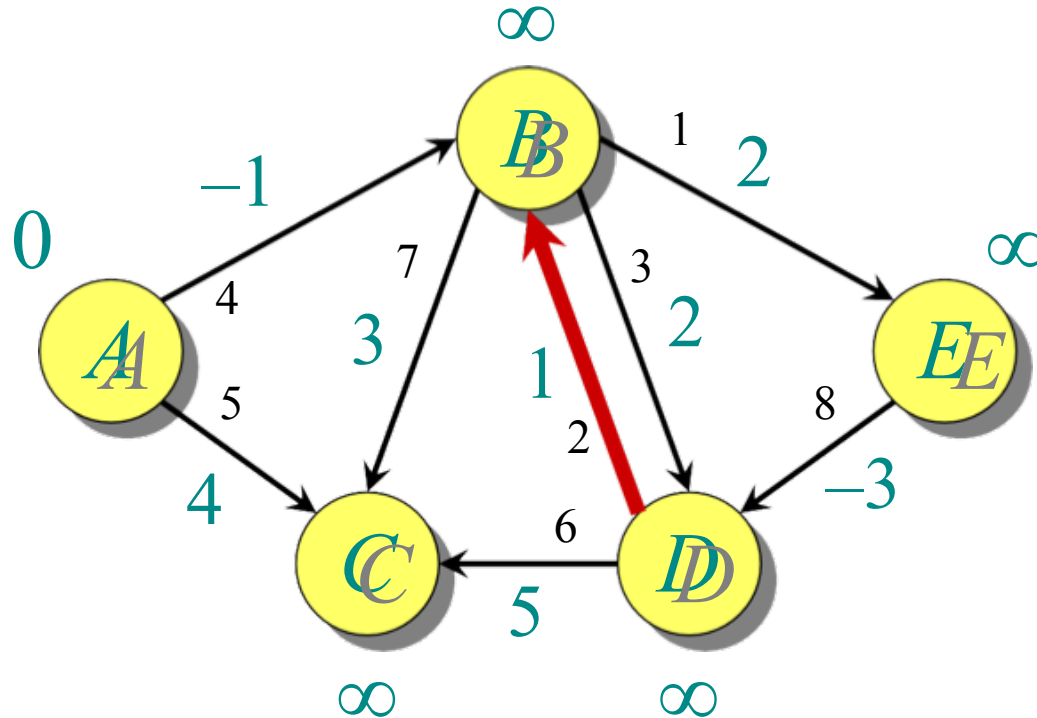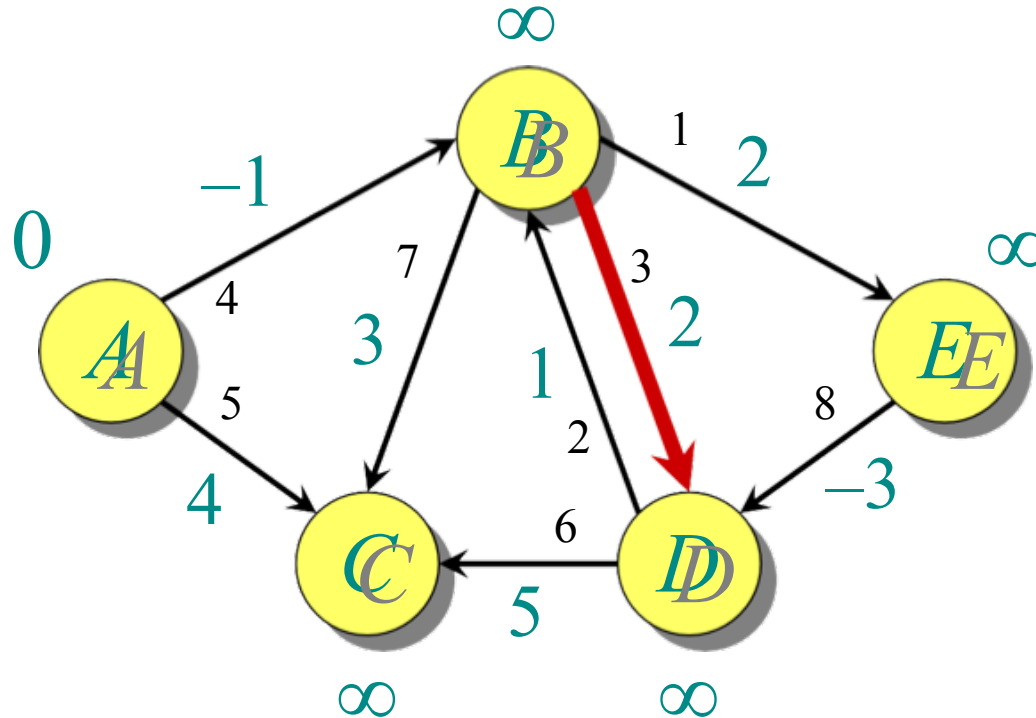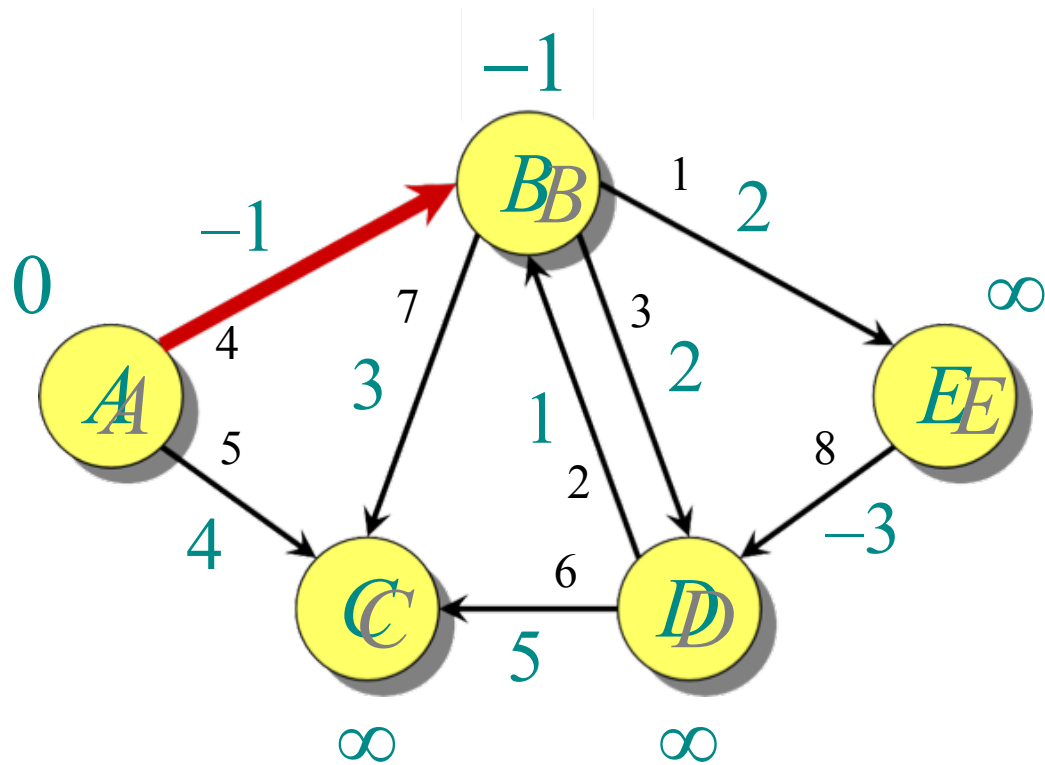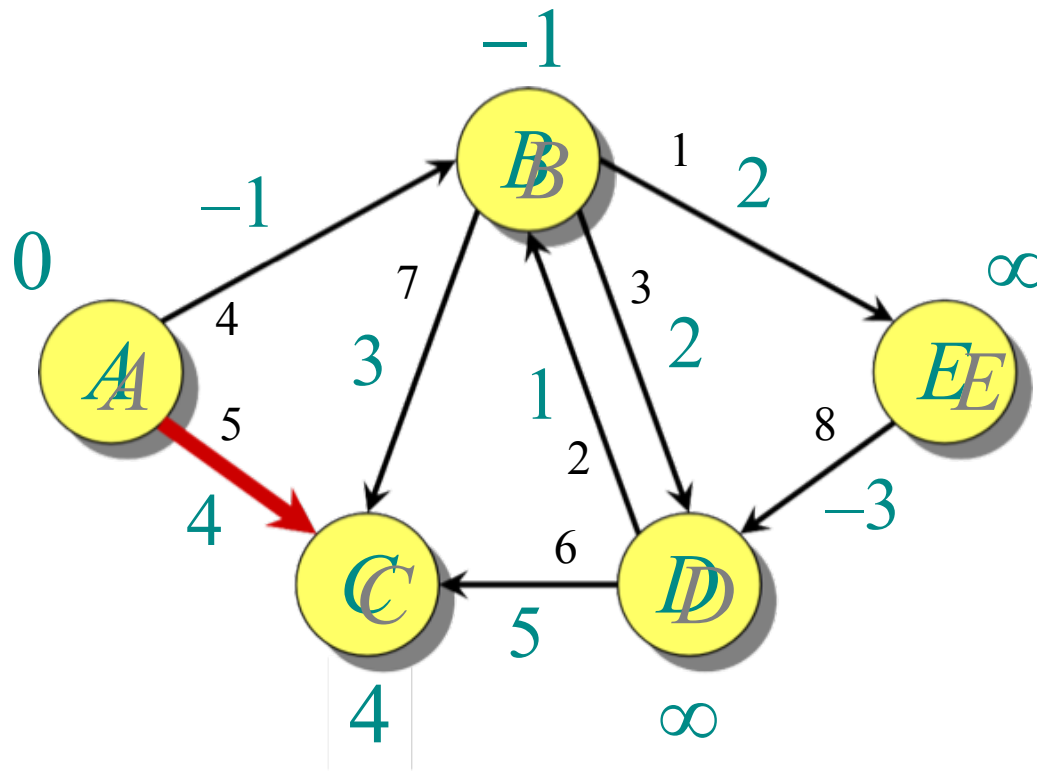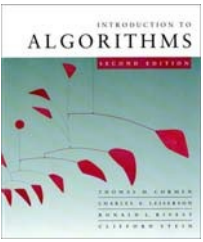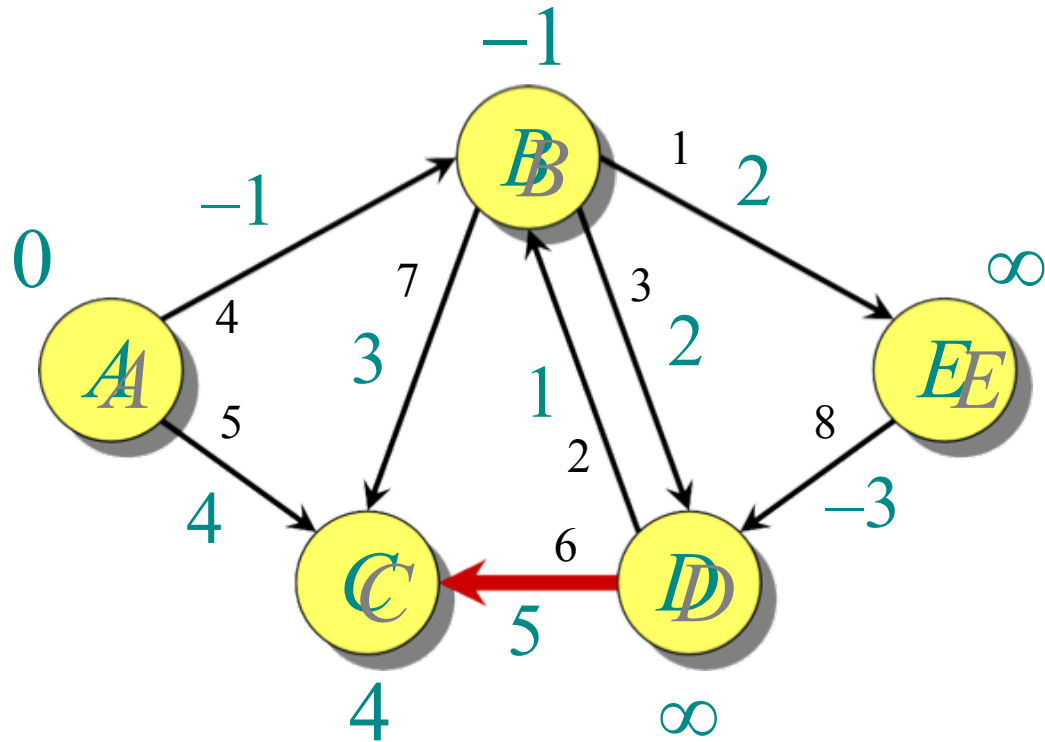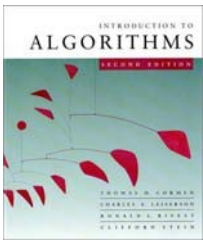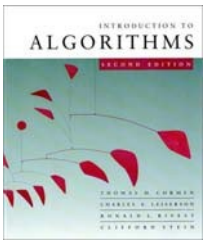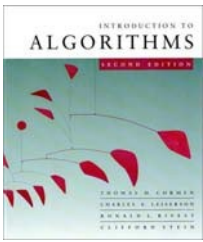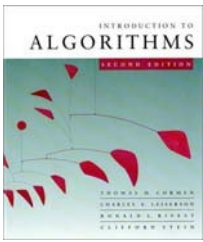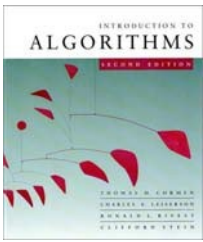# Example of Bellman-Ford
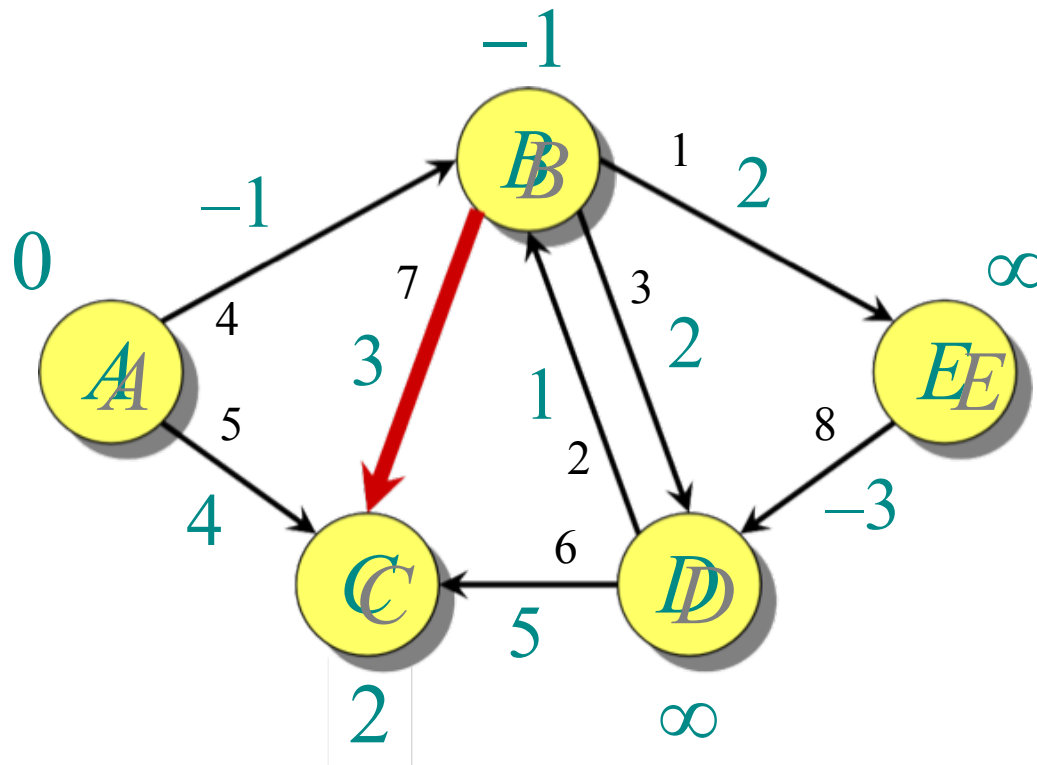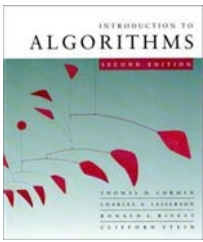
Example of Bellman-Ford

Example of Bellman-Ford

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

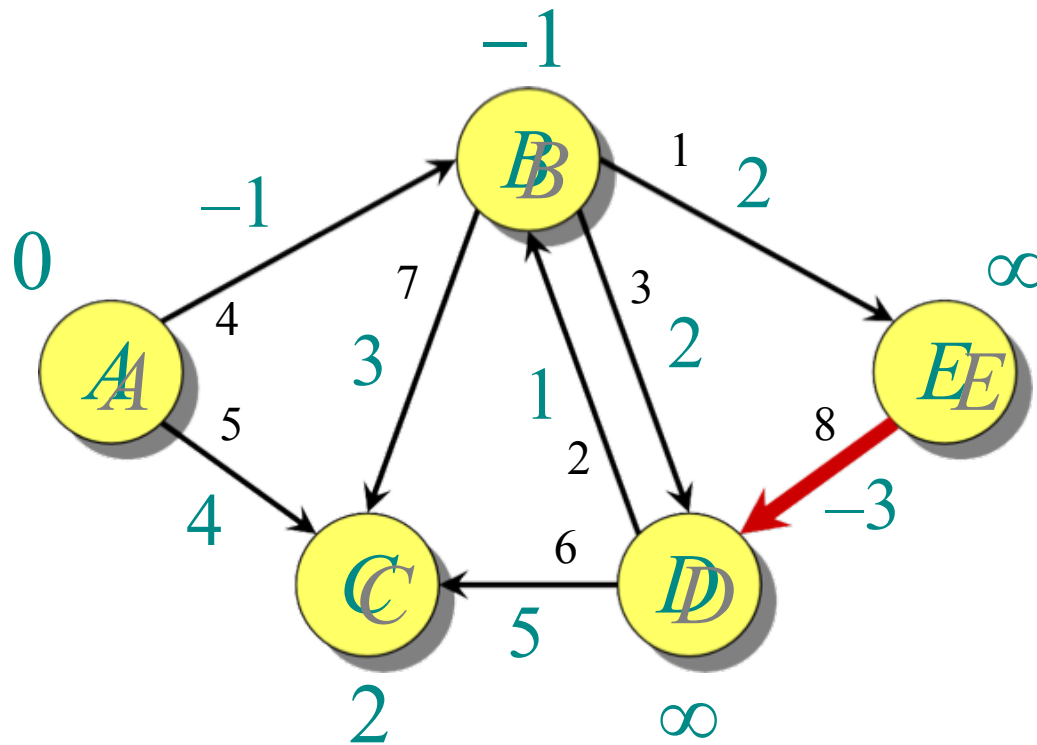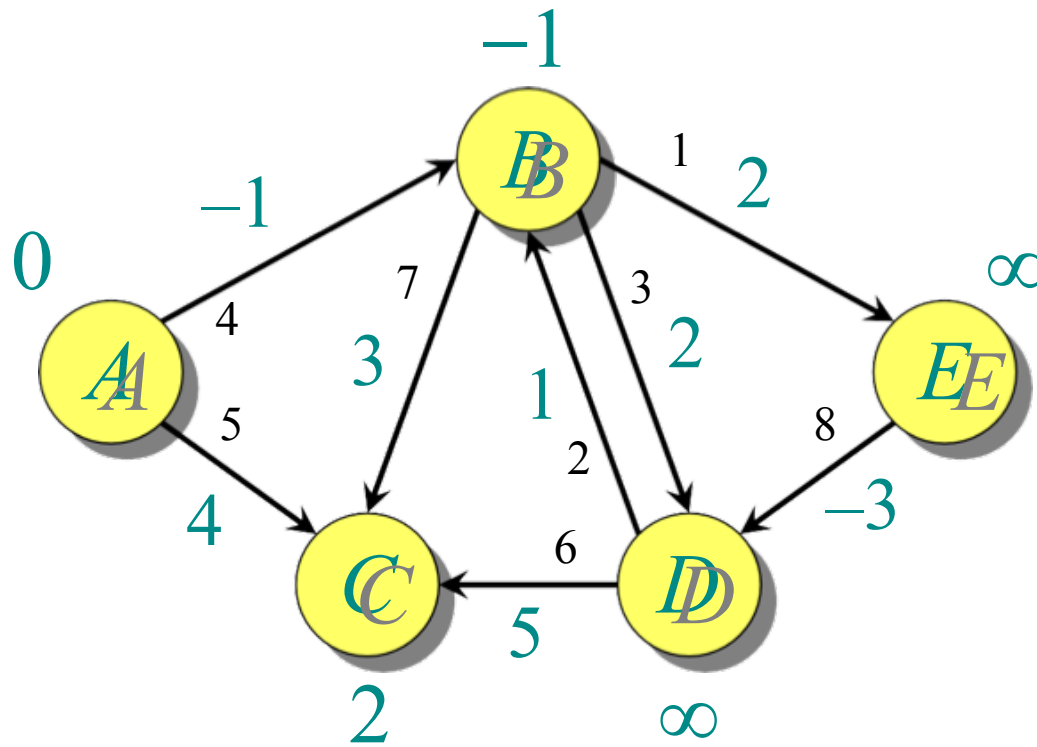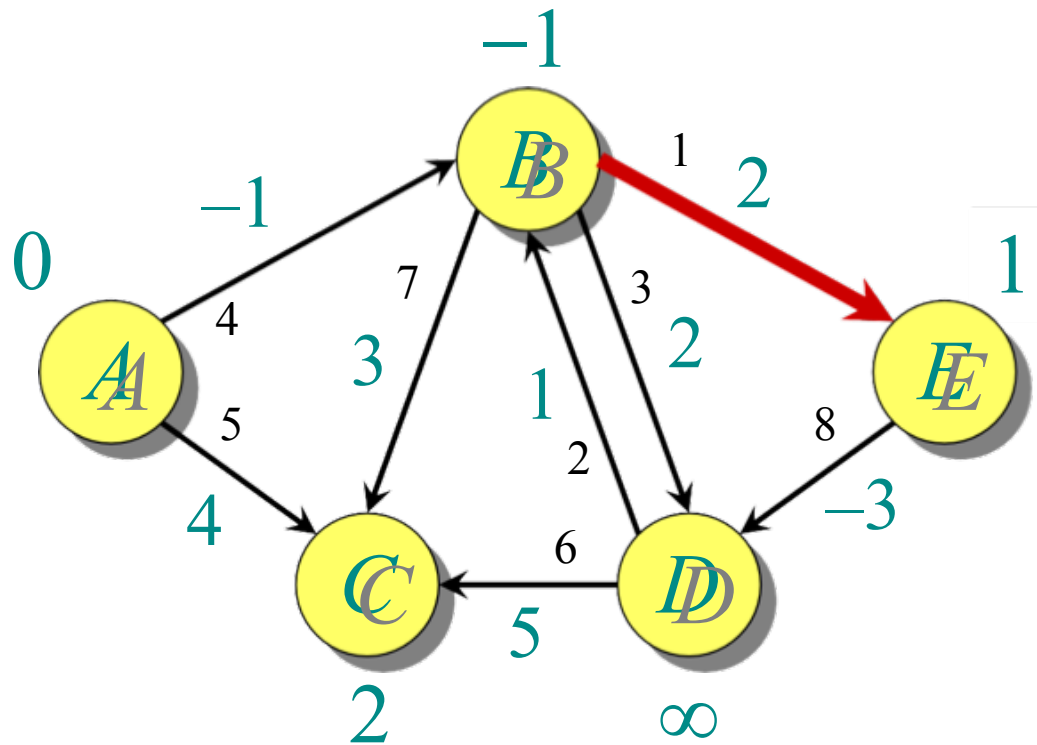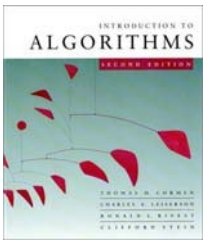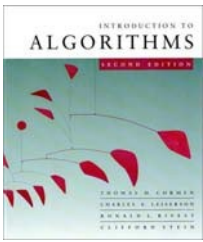# Example of Bellman-Ford

Example of Bellman-Ford

Example of Bellman-Ford

End of pass 1.

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

Example of Bellman-Ford

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

Example of Bellman-Ford
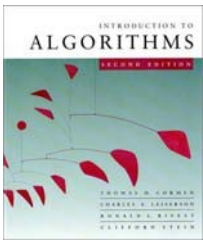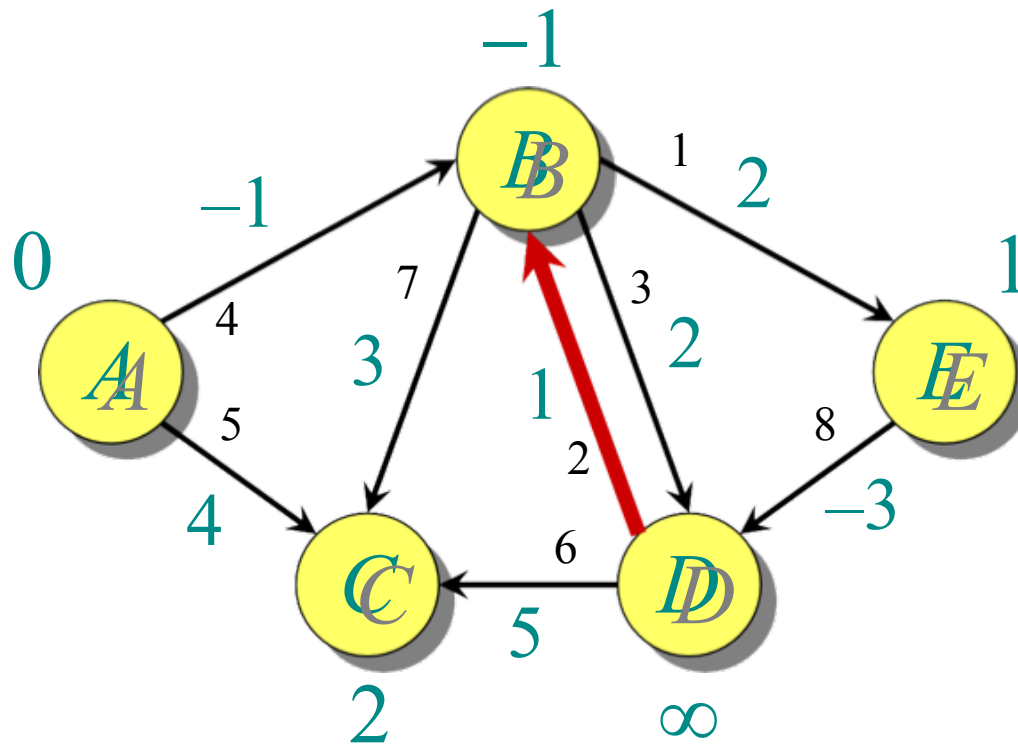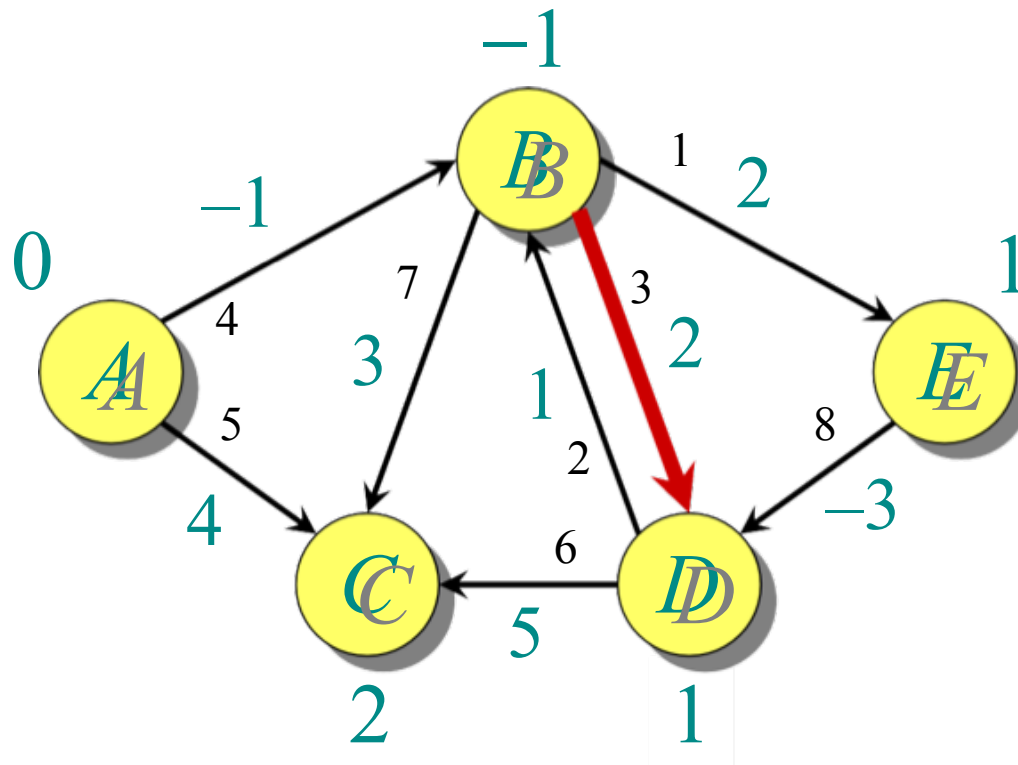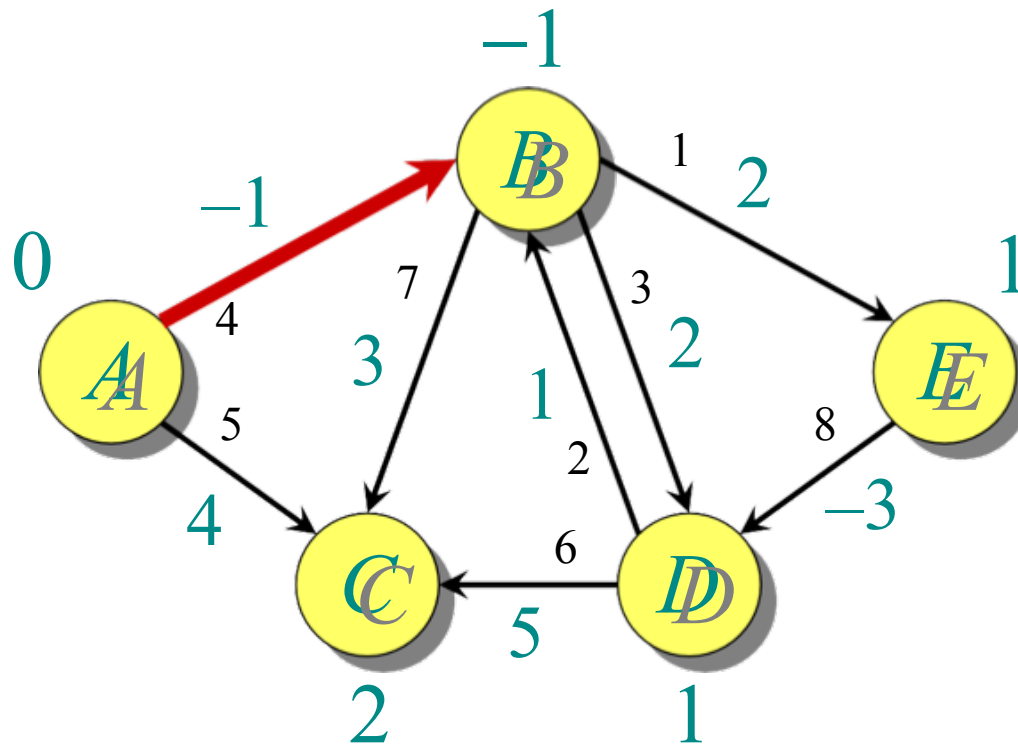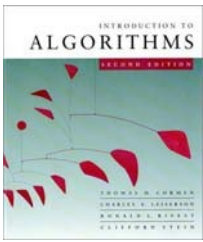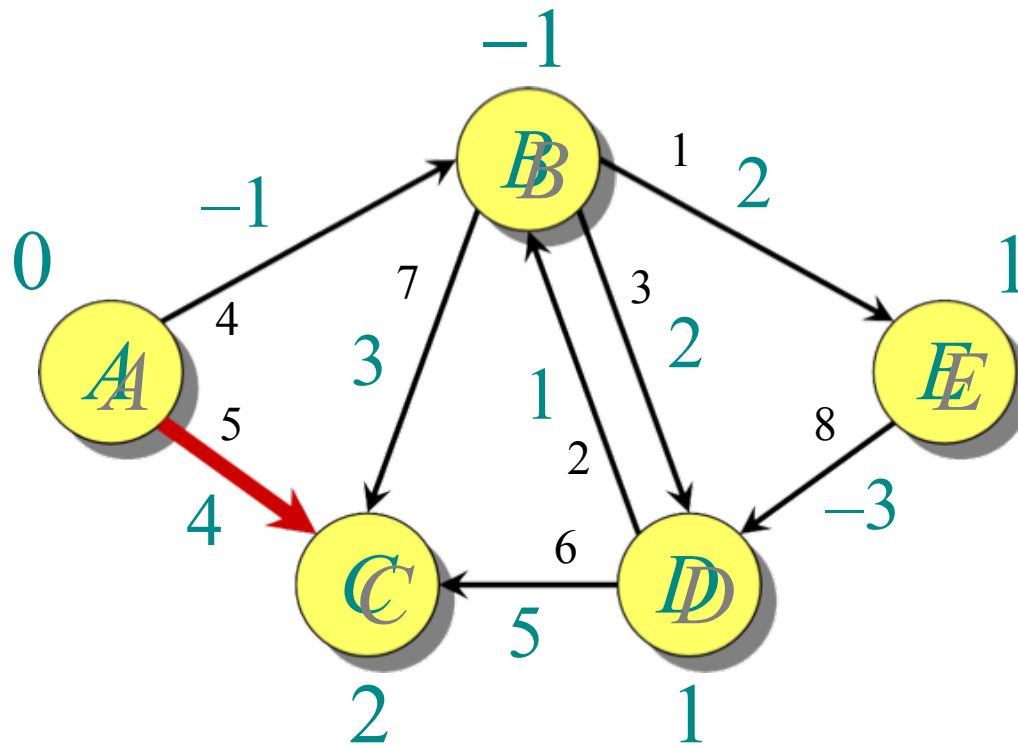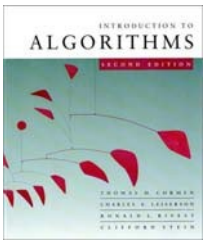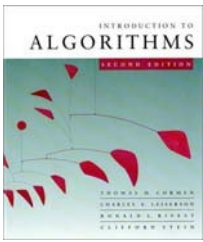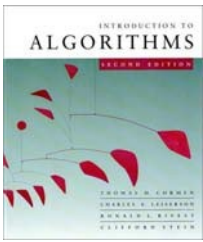
Example of Bellman-Ford

Example of Bellman-Ford

Example of Bellman-Ford

Example of Bellman-Ford

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

Example of Bellman-Ford

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

Example of Bellman-Ford

Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson
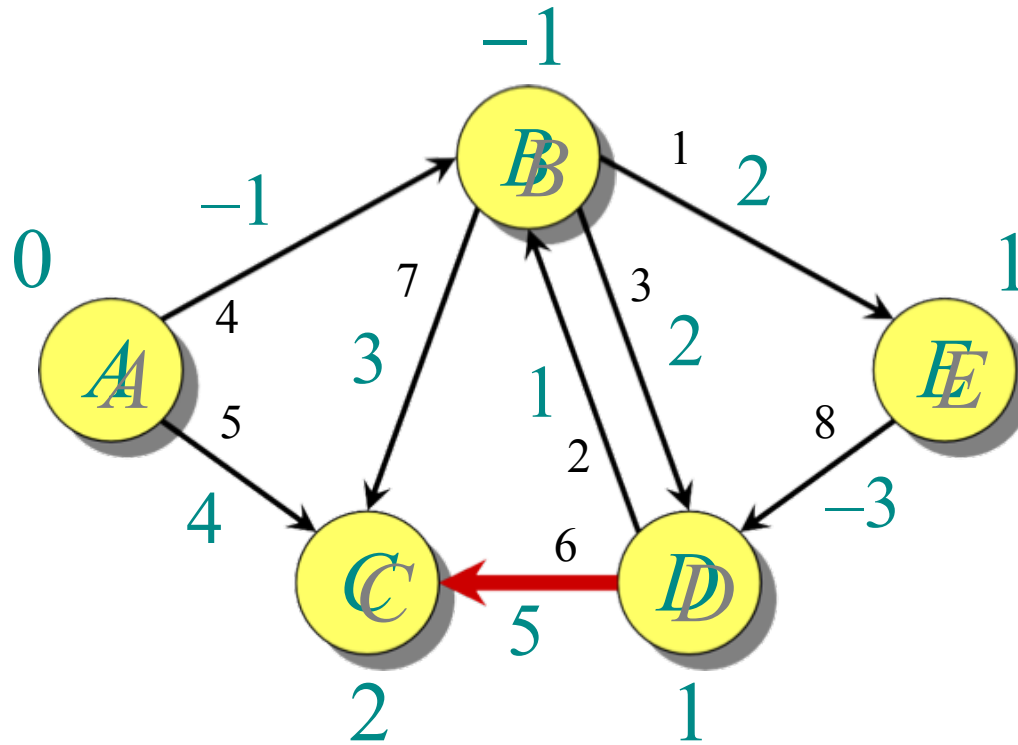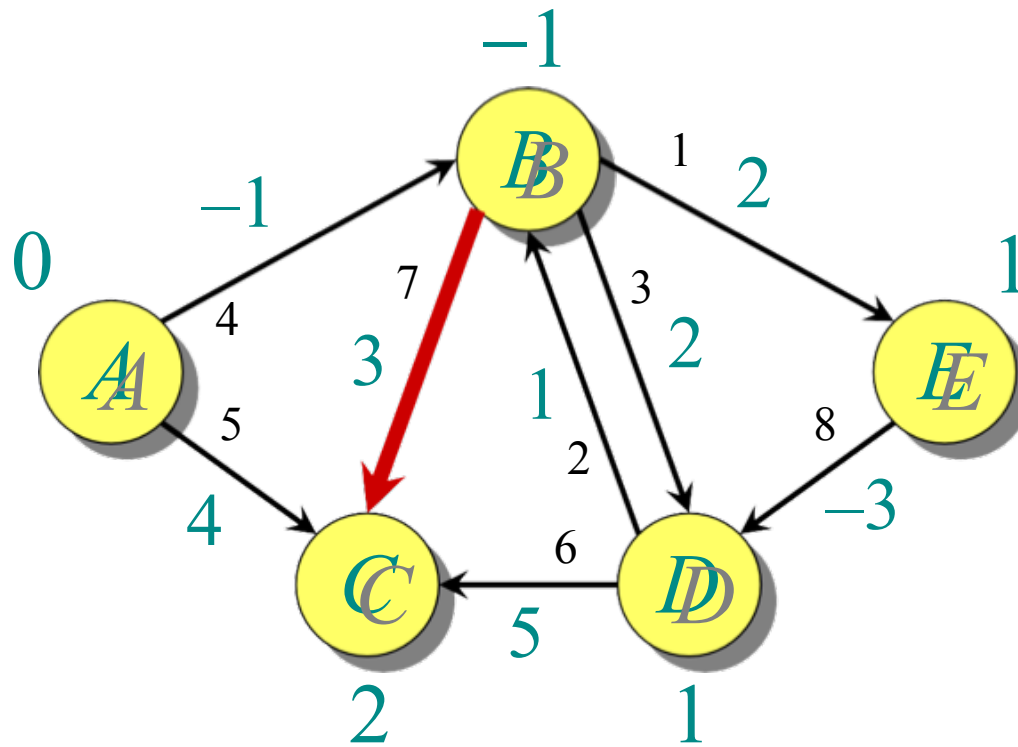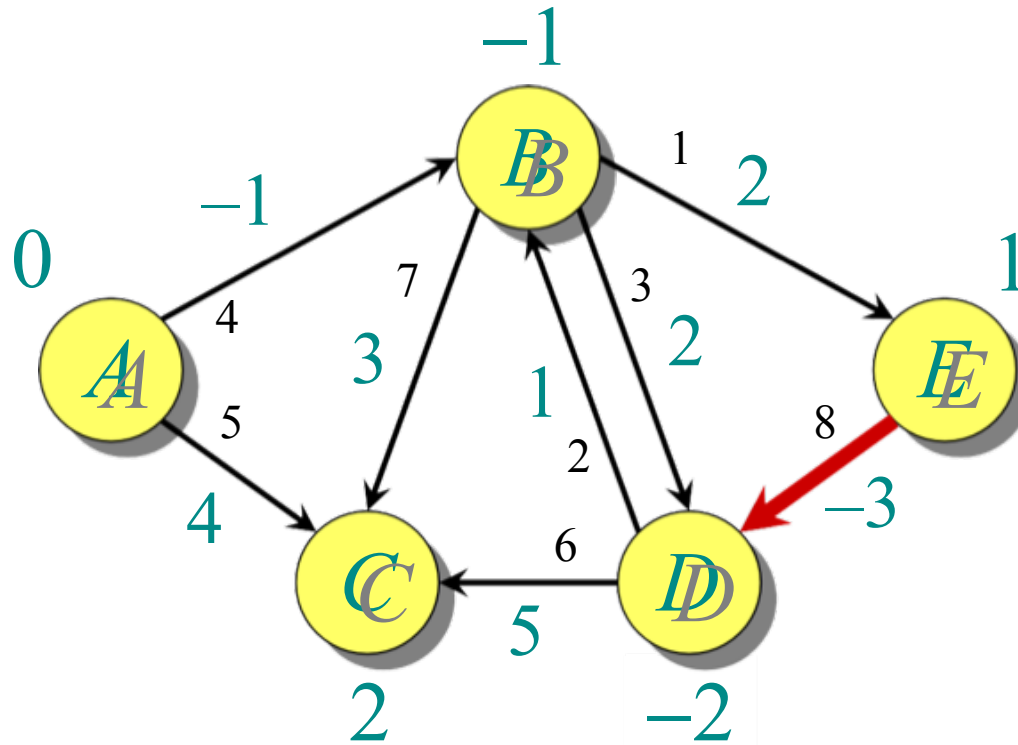
Example of Bellman-Ford

End of pass 2 (and 3 and 4).
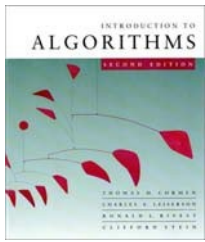
Copyright © 2001-5 by Erik D. Demaine and Charles E. Leiserson

Correctness

Theorem. If $G = (V, E)$ contains no negative-weight cycles, then after the Bellman-Ford algorithm executes, $d[v] = \delta(s, v)$ for all $v \in V$.

Correctness

Theorem. If $G = (V, E)$ contains no negative-weight cycles, then after the Bellman-Ford algorithm executes, $d[v] = \delta(s, v)$ for all $v \in V$.
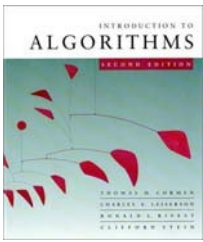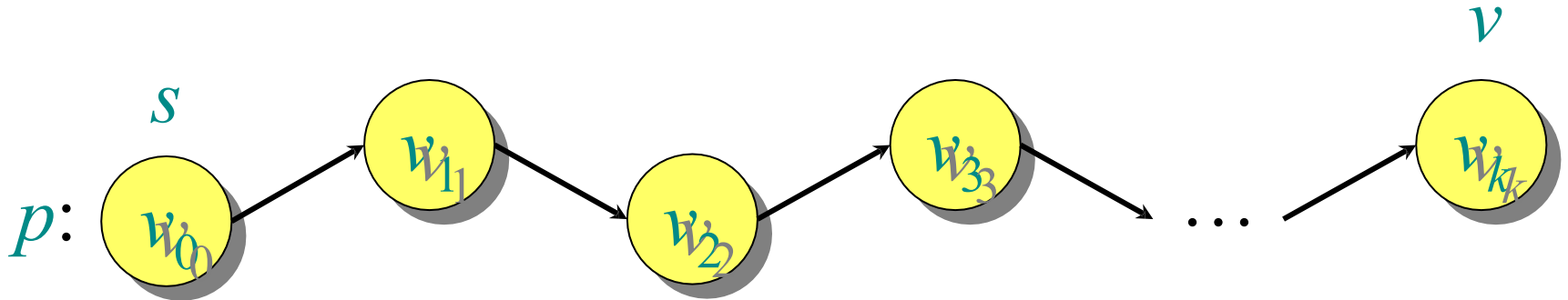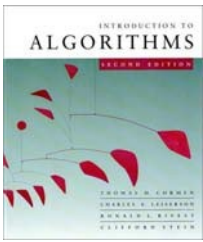
Proof. Let $v \in V$ be any vertex, and consider a shortest path p from s to v with the minimum number of edges.
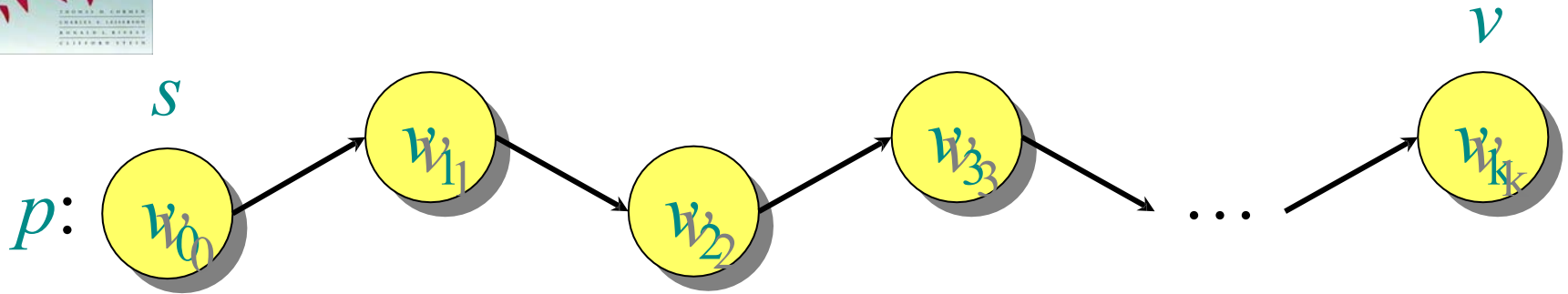

p:

s ... v

Since p is a shortest path, we have

$$\delta(s, v_i) = \delta(s, v_{i-1}) + w(v_{i-1}, v_i) .$$

Correctness (continued)

Initially, $d[v_0] = 0 = \delta(s, v_0)$, and $d[v_0]$ is unchanged by subsequent relaxations (because of the lemma from Lecture 14 that $d[v] \geq \delta(s, v)$).

- After 1 pass through E, we have $d[v_1] = \delta(s, v_1)$.
- After 2 passes through E, we have $d[v_2] = \delta(s, v_2)$.
  M
- After k passes through E, we have $d[v_k] = \delta(s, v_k)$.

Since G contains no negative-weight cycles, p is simple. Longest simple path has $\leq |V| - 1$ edges.