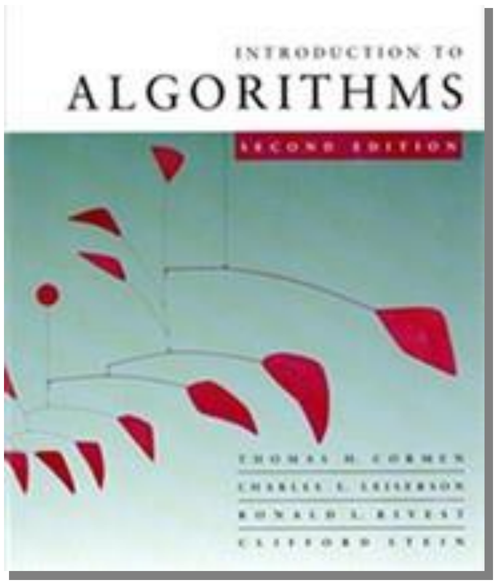
Algorithms



LECTURE 14

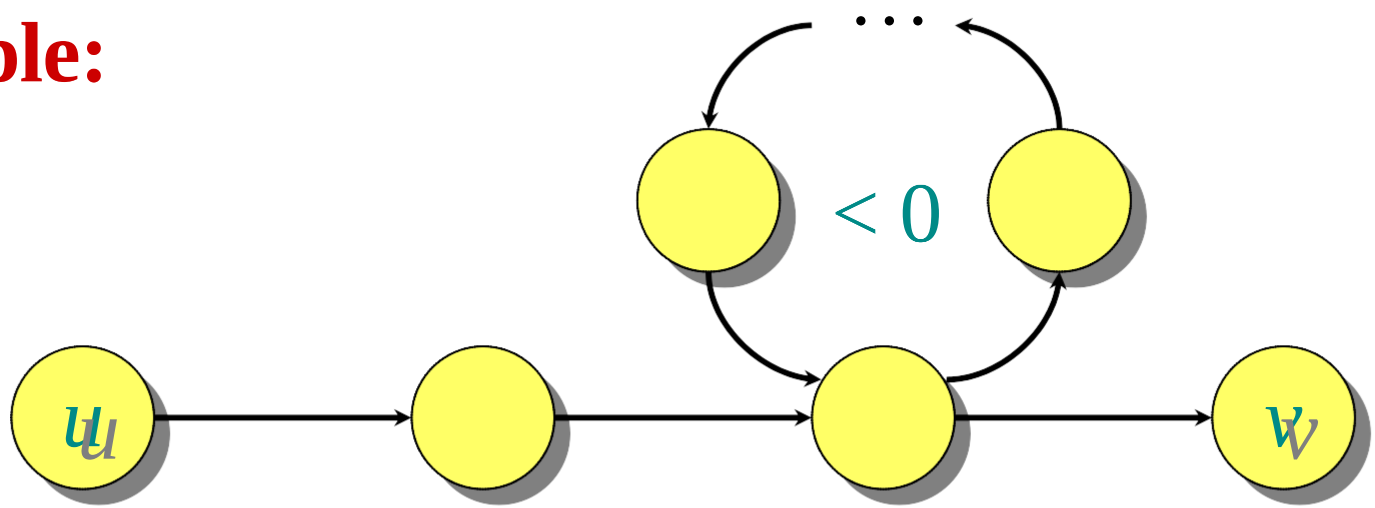
Shortest Paths II

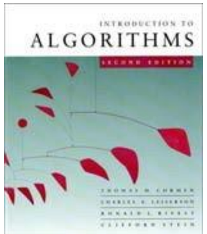
- Bellman-Ford algorithm
- Floyd-Warshal algorithm

Negative-weight cycles

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

Example:

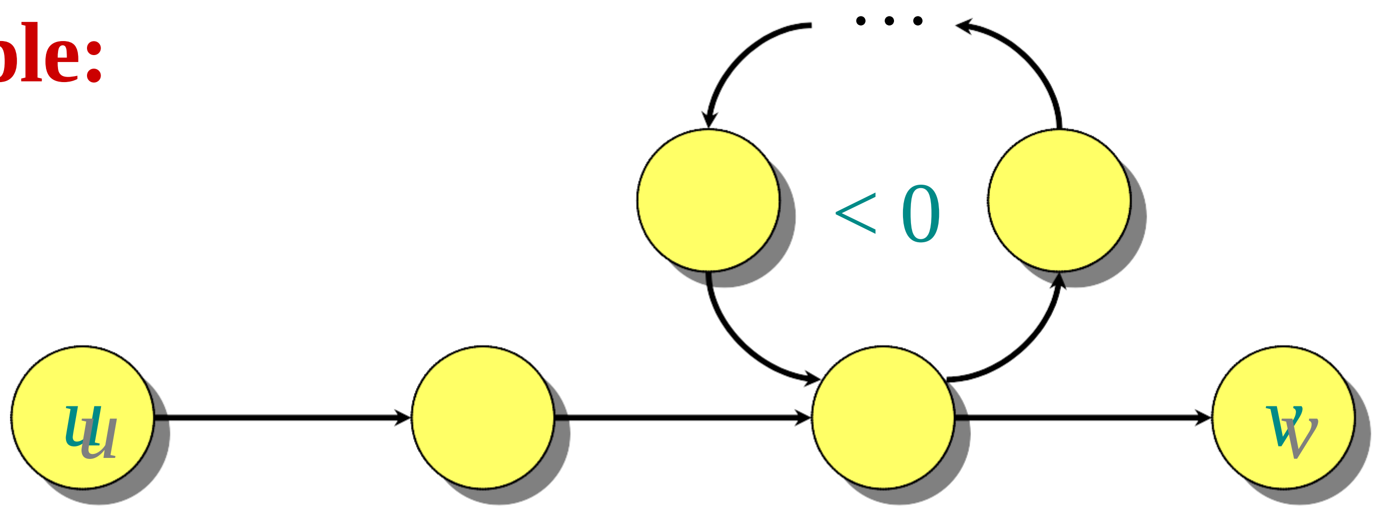




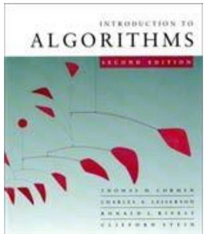
Negative-weight cycles

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.



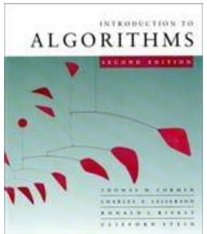
Bellman-Ford algorithm

for each $v \in V$

  do $d[v] \leftarrow \infty$

$d[s] \leftarrow 0$

} initialization



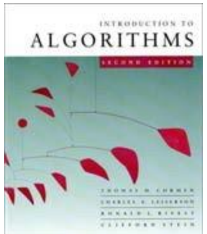
Bellman-Ford algorithm

for each $v \in V$
    do $d[v] \leftarrow \infty$
$d[s] \leftarrow 0$

} initialization

for $i \leftarrow 1$ **to ????**
    do for each edge $(u, v) \in E$
        do if $d[v] > d[u] + w(u, v)$
            then $d[v] \leftarrow d[u] + w(u, v)$

} ***relaxation step***



Bellman-Ford algorithm

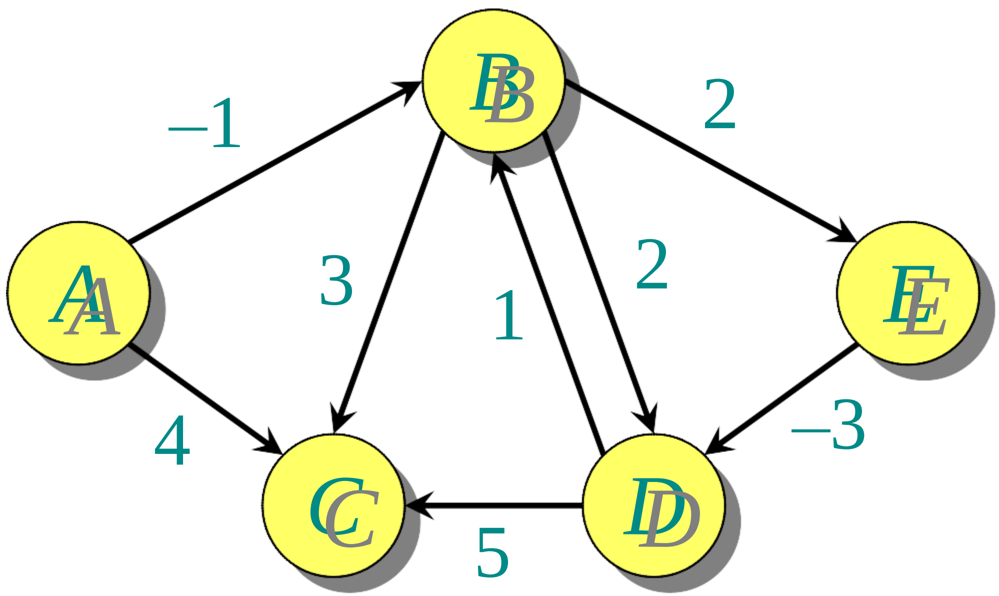
for each $v \in V$
    do $d[v] \leftarrow \infty$
$d[s] \leftarrow 0$

} initialization

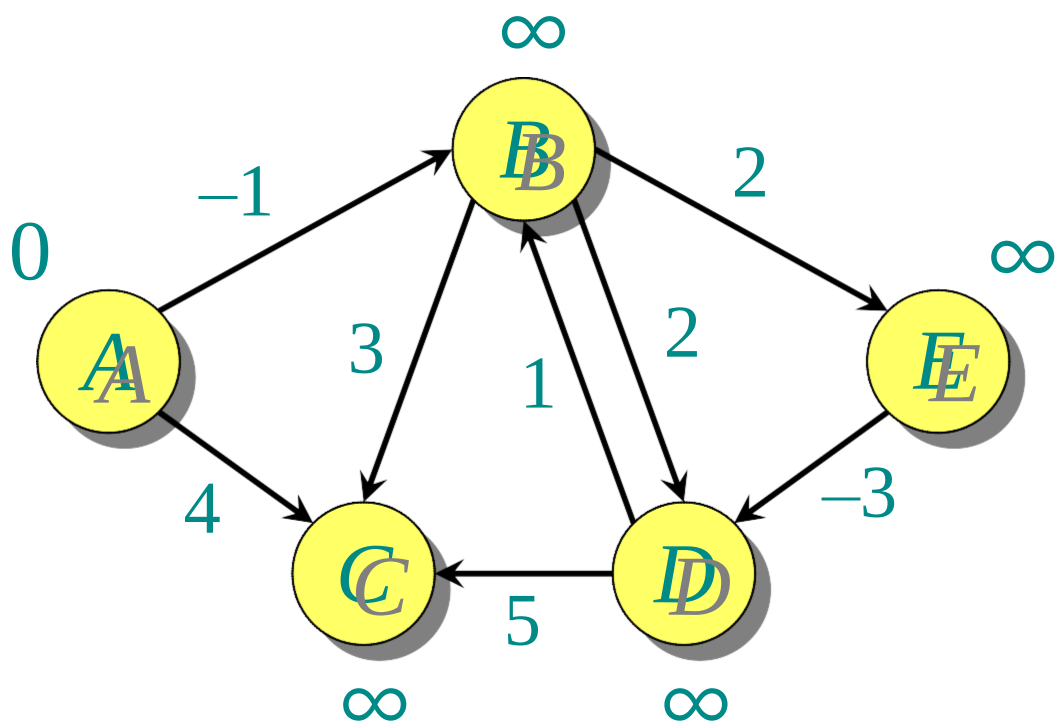
for $i \leftarrow 1$ **to** $|V| - 1$
    do for each edge $(u, v) \in E$
        do if $d[v] > d[u] + w(u, v)$
            then $d[v] \leftarrow d[u] + w(u, v)$

} ***relaxation step***

Example of Bellman-Ford

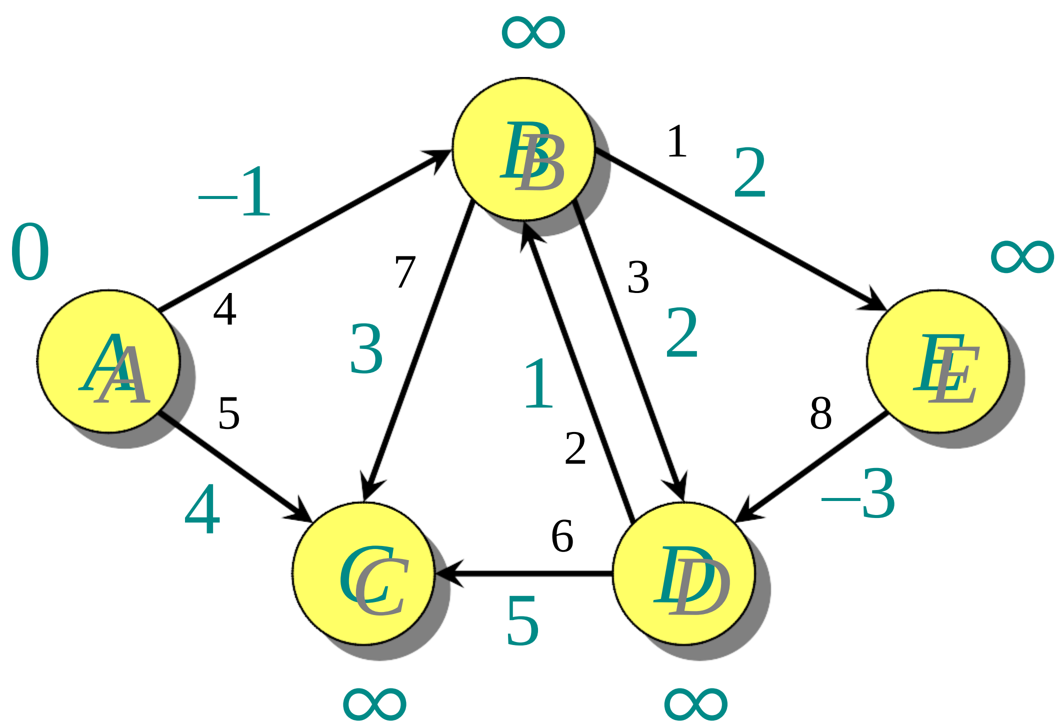


Example of Bellman-Ford



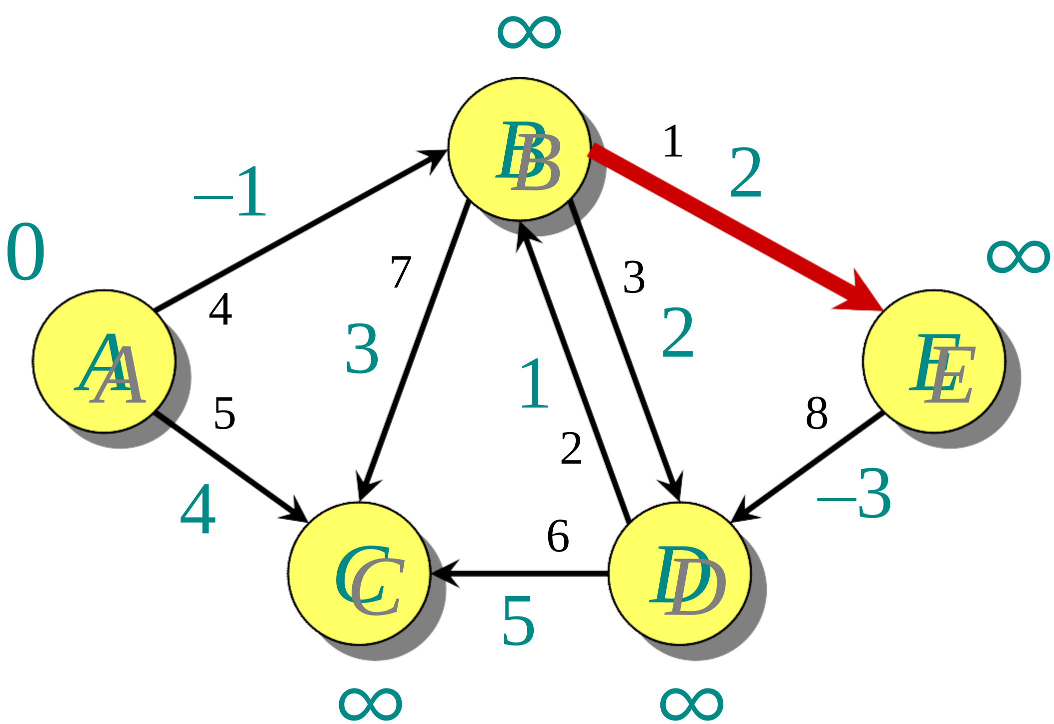
Initialization.

Example of Bellman-Ford

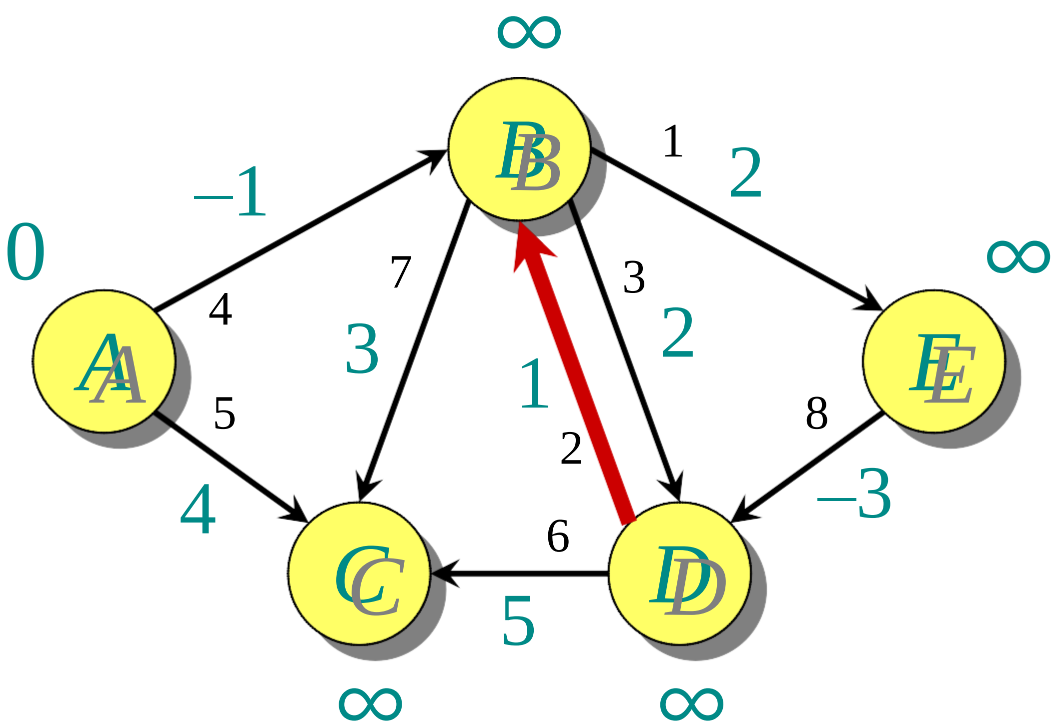


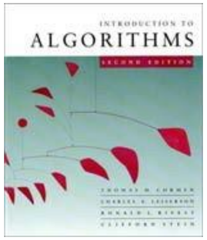
Order of edge relaxation.

Example of Bellman-Ford

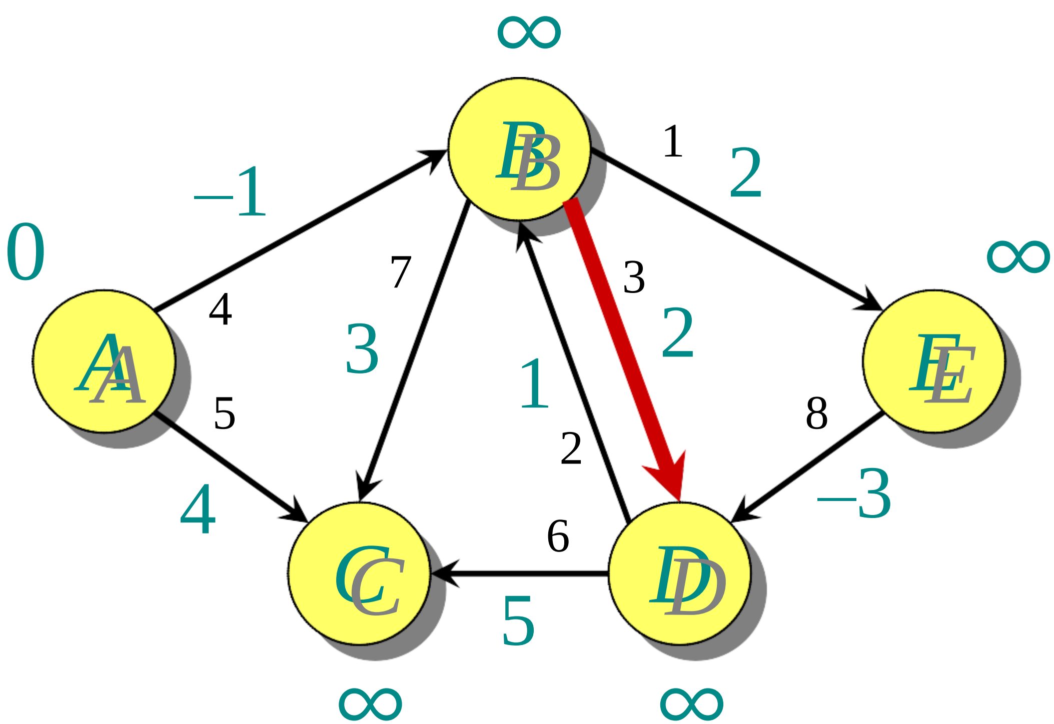


Example of Bellman-Ford

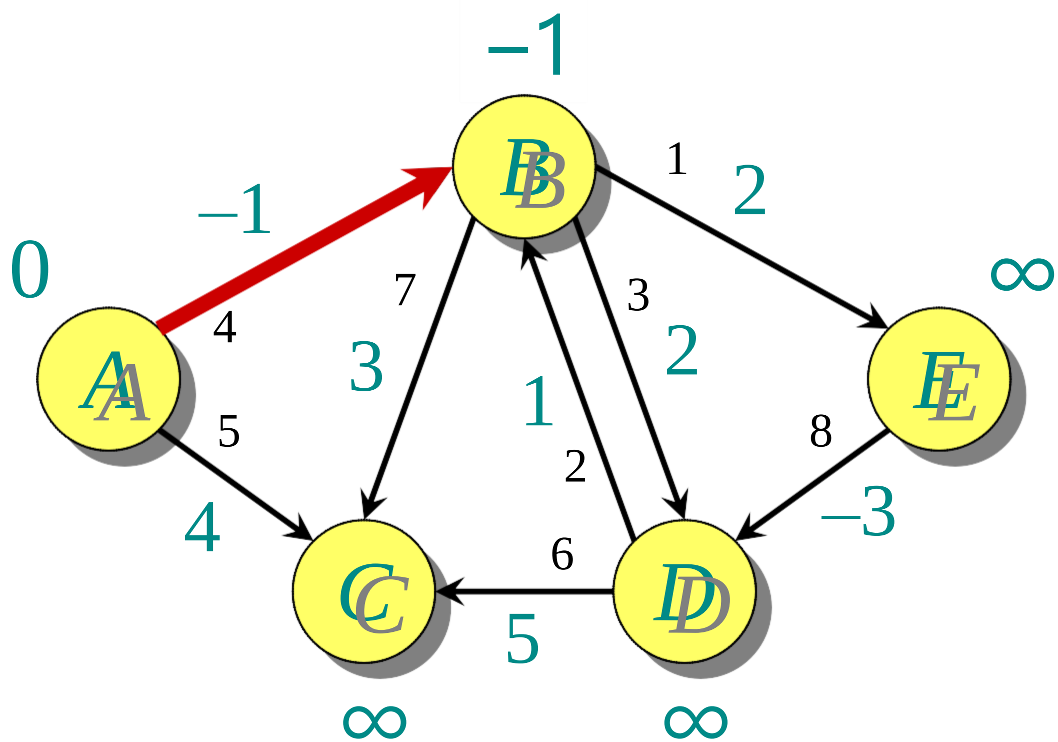




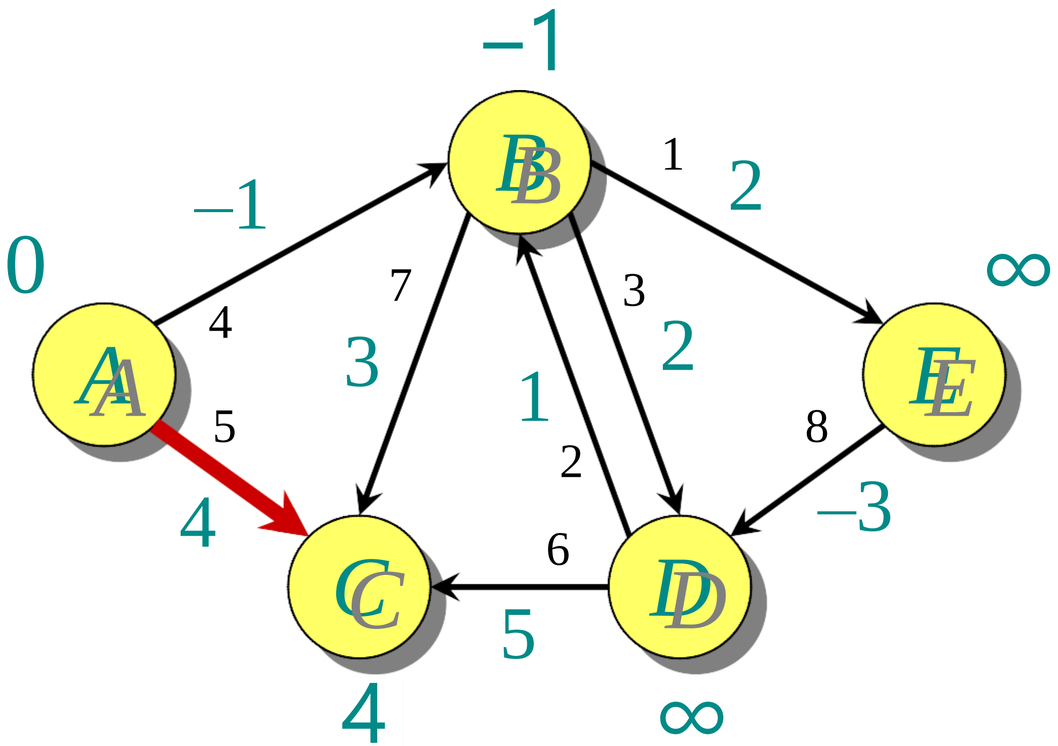
Example of Bellman-Ford

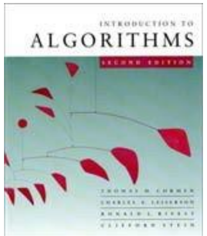


Example of Bellman-Ford

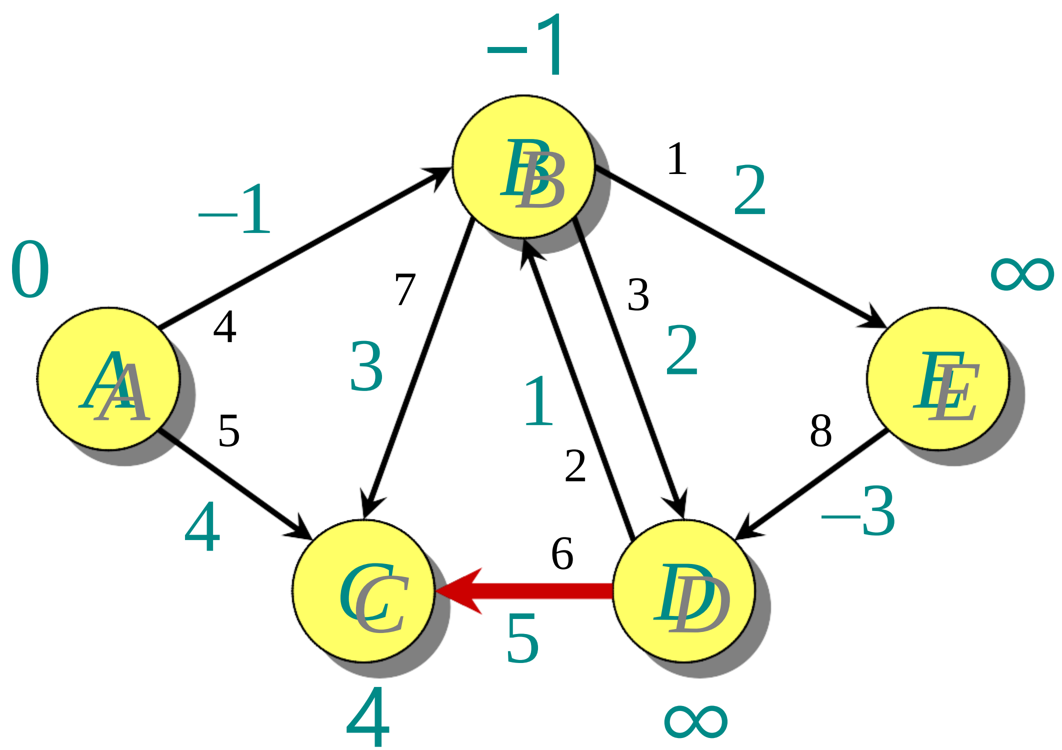


Example of Bellman-Ford

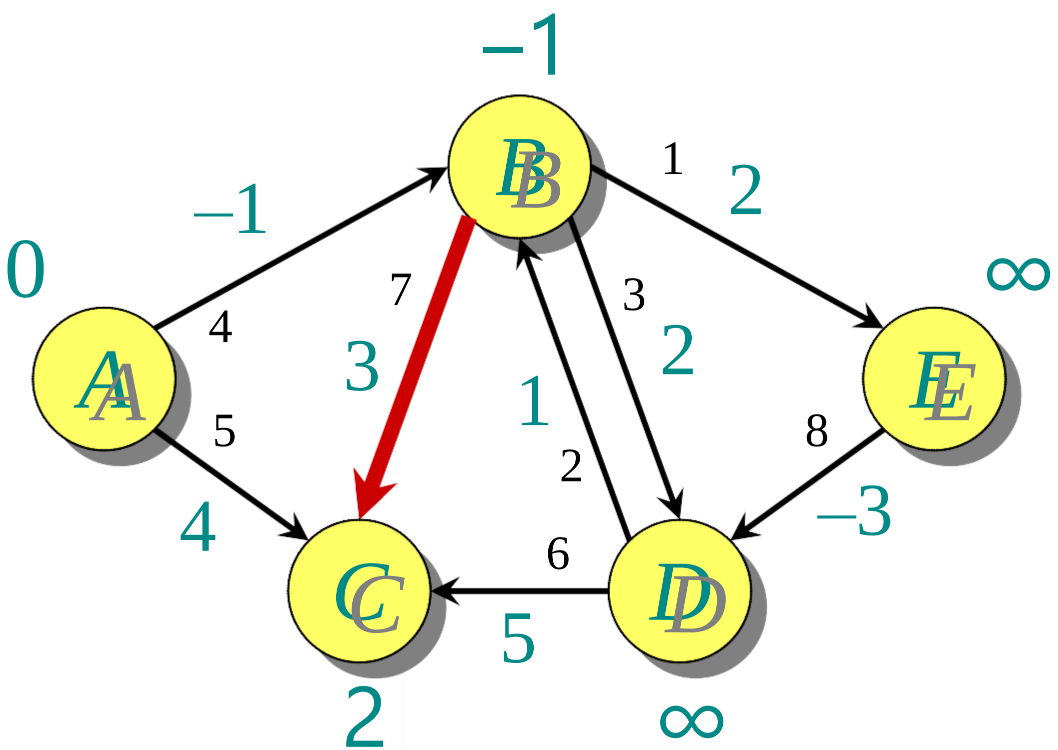




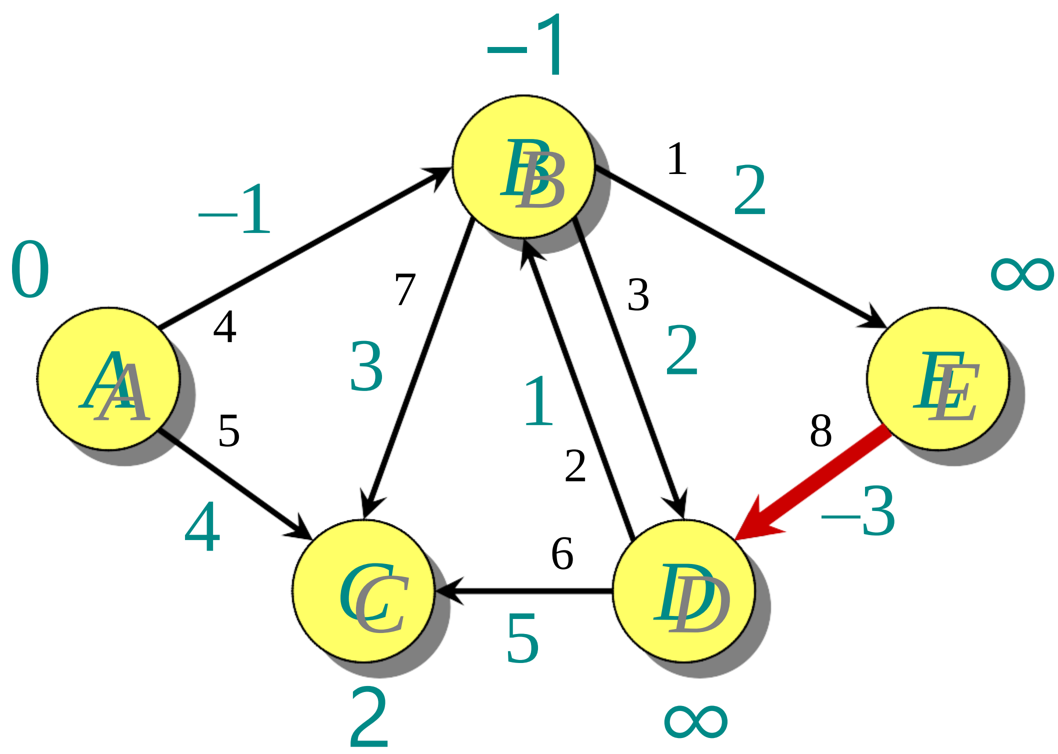
Example of Bellman-Ford



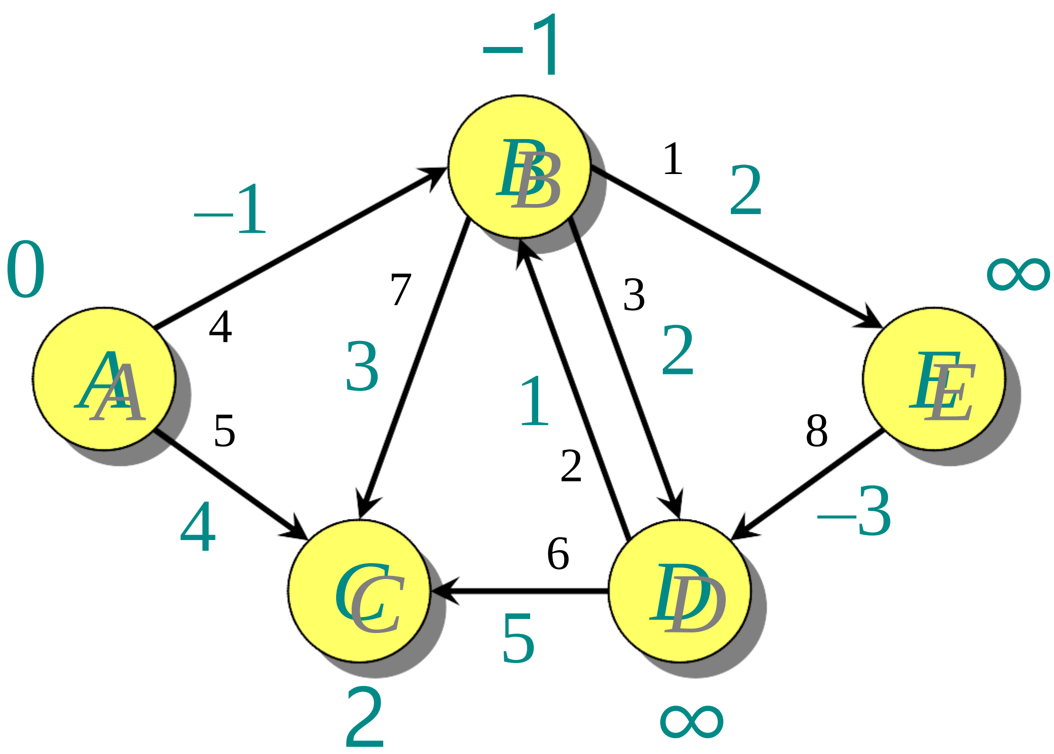
Example of Bellman-Ford



Example of Bellman-Ford

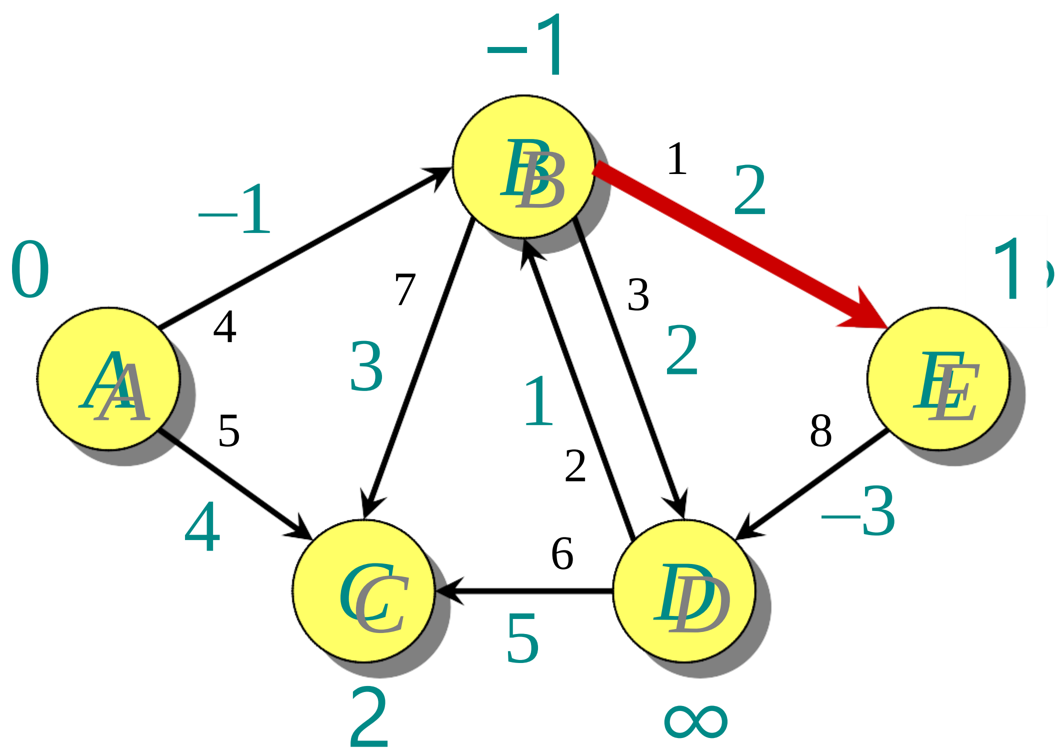


Example of Bellman-Ford

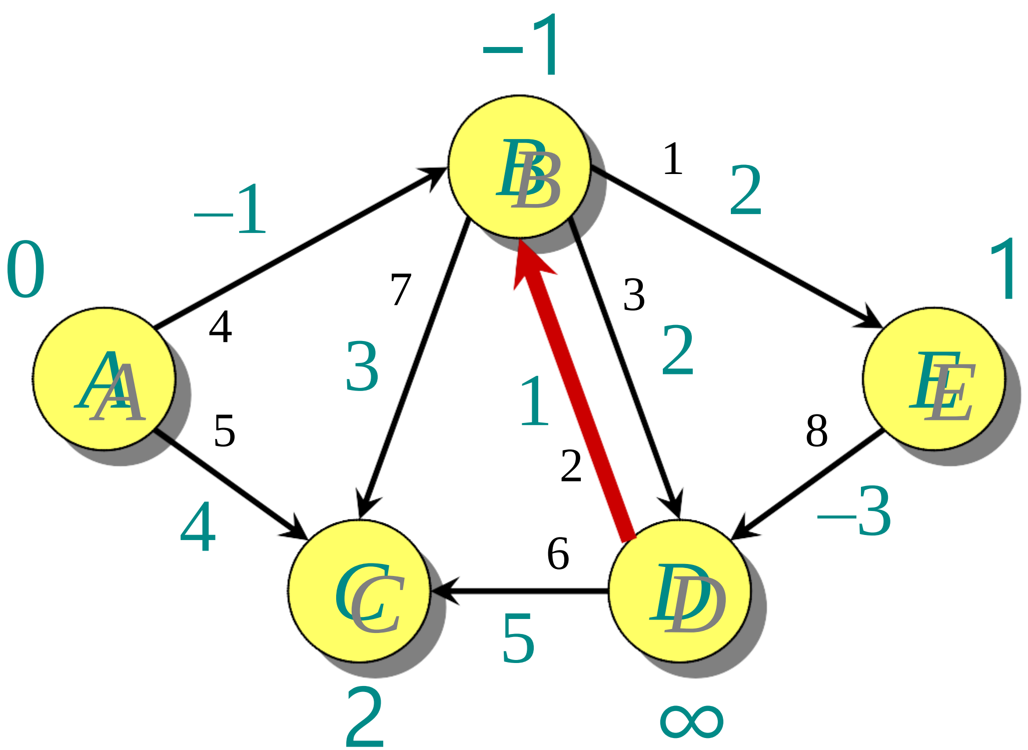


End of pass 1.

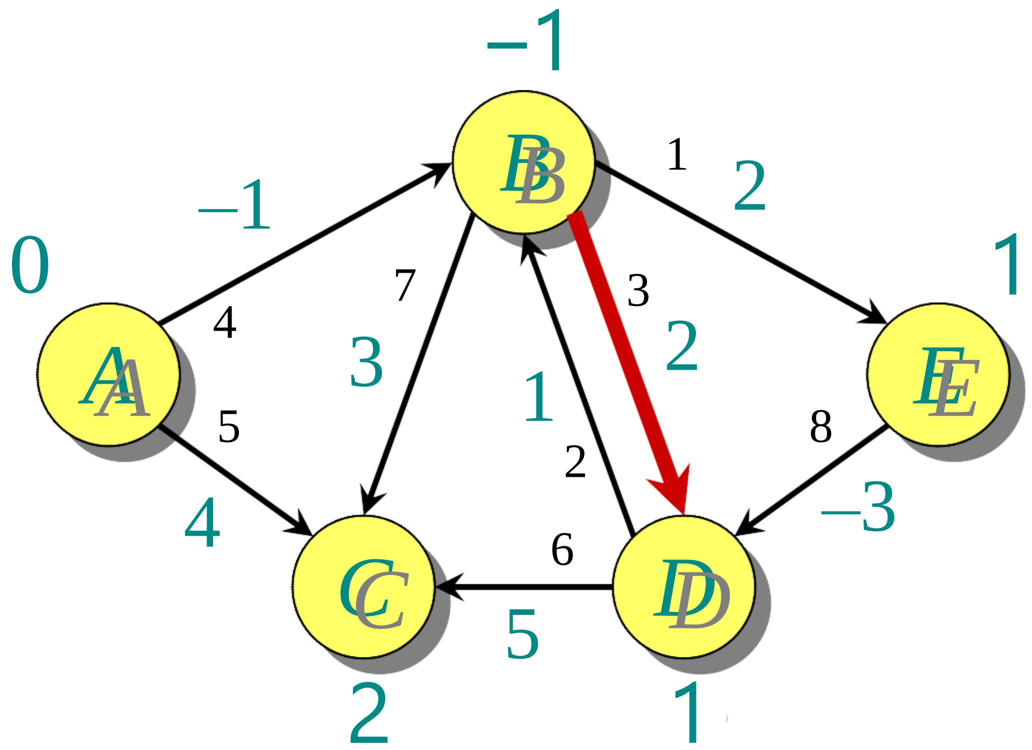
Example of Bellman-Ford



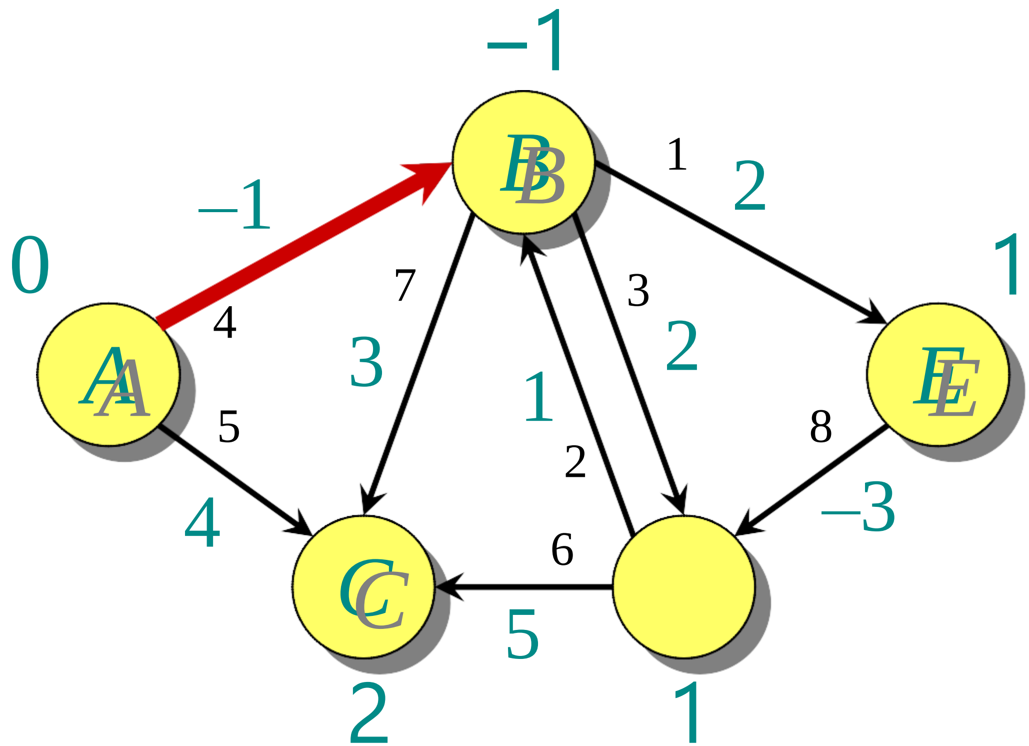
Example of Bellman-Ford



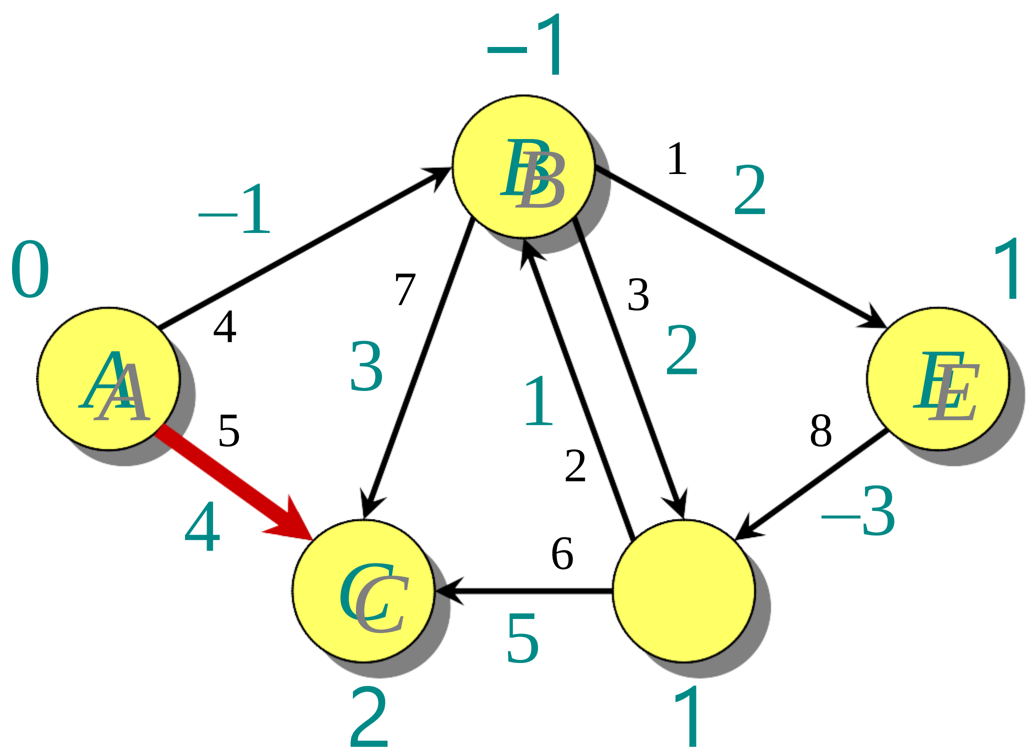
Example of Bellman-Ford



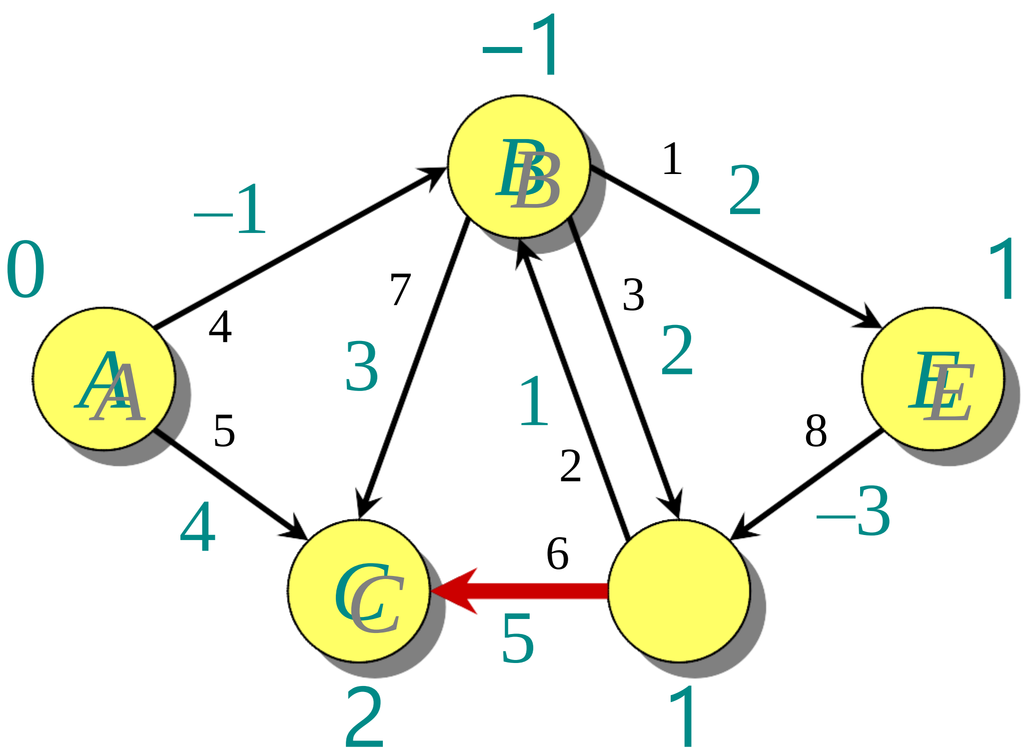
Example of Bellman-Ford



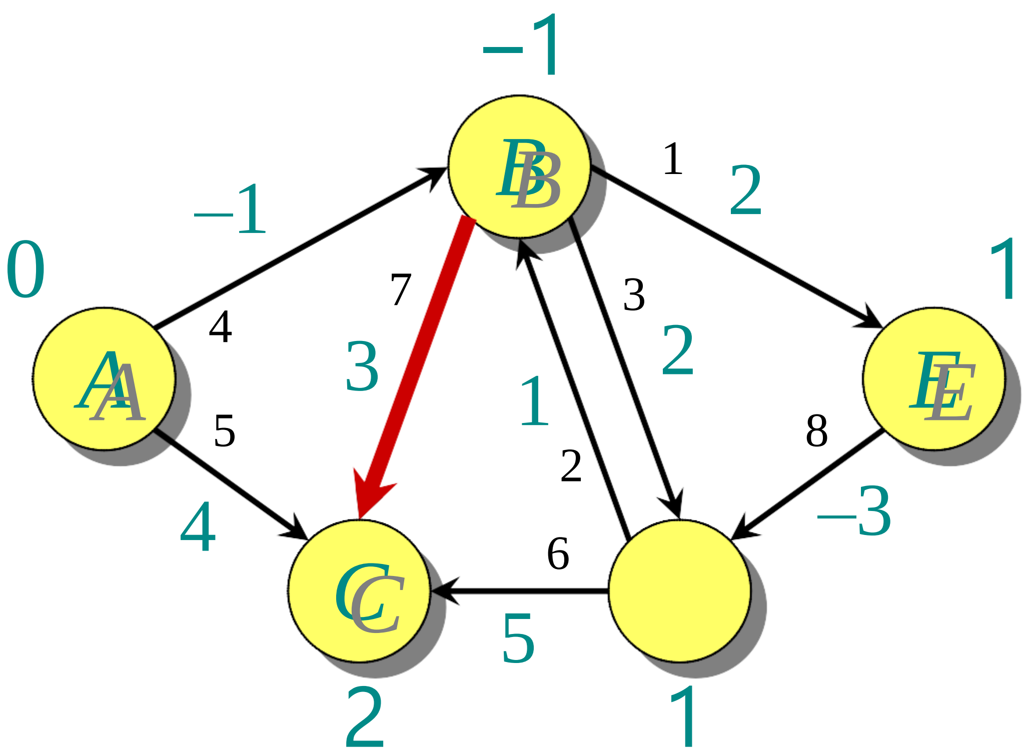
Example of Bellman-Ford



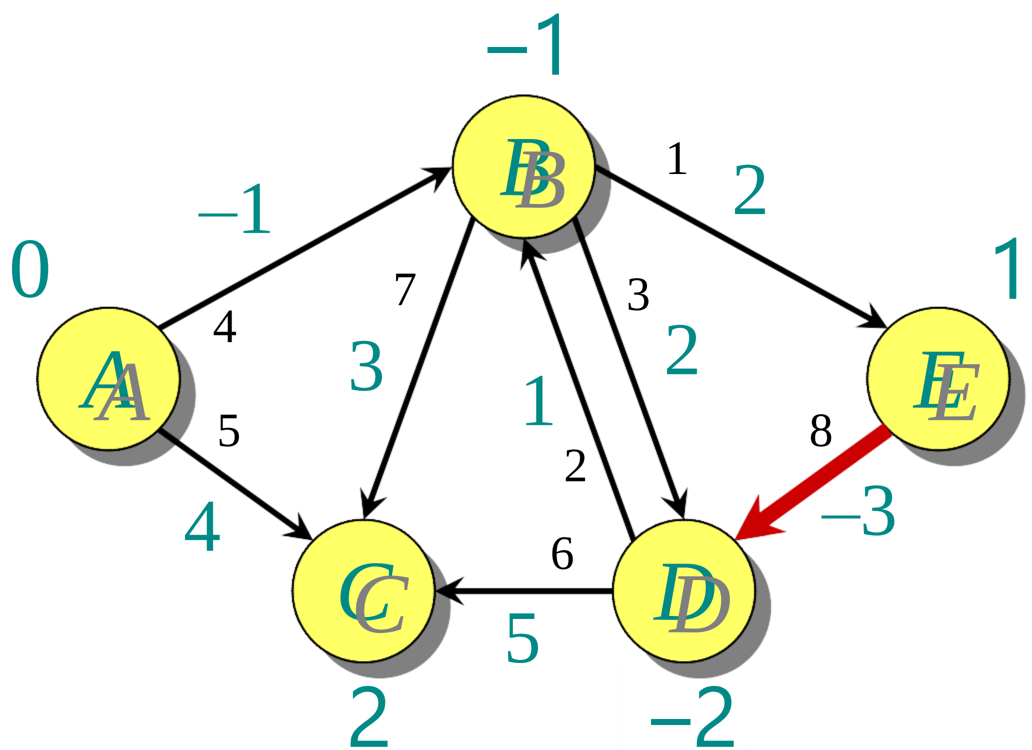
Example of Bellman-Ford



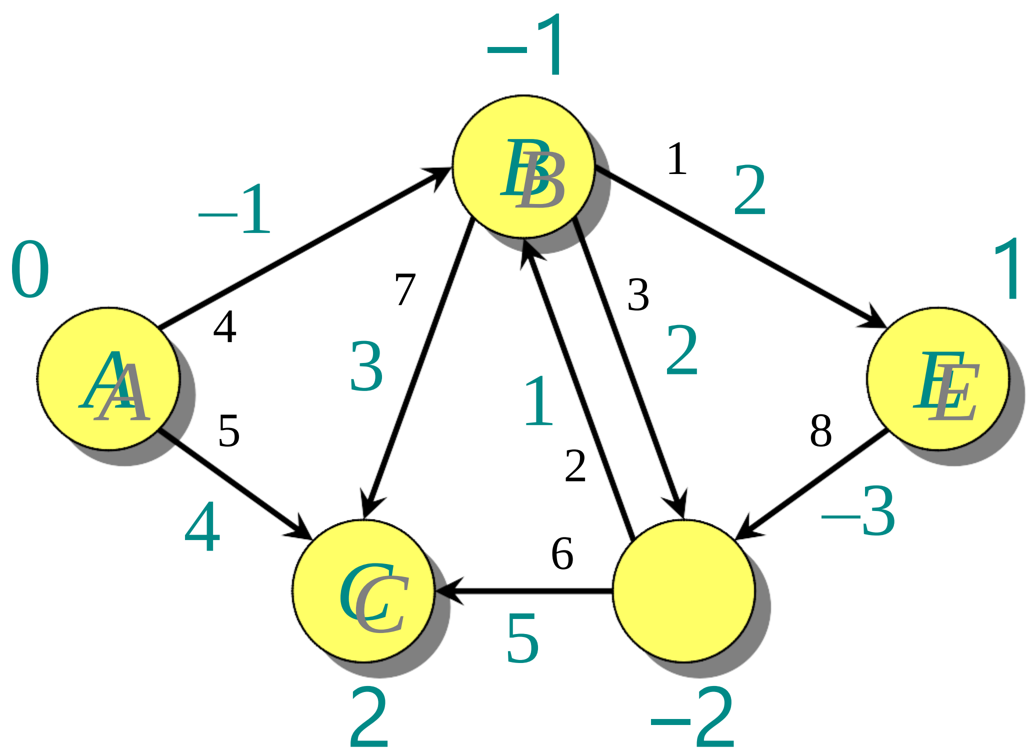
Example of Bellman-Ford



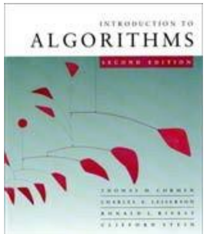
Example of Bellman-Ford



Example of Bellman-Ford



End of pass 2 (and 3 and 4).



Bellman-Ford algorithm

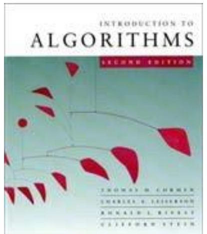
for each $v \in V$
    do $d[v] \leftarrow \infty$
$d[s] \leftarrow 0$

} initialization

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

} ***relaxation step***

???????
    then report that a negative-weight cycle exists



Bellman-Ford algorithm

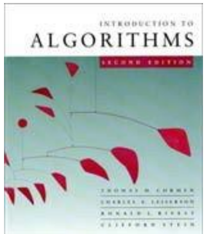
for each $v \in V$
    do $d[v] \leftarrow \infty$
$d[s] \leftarrow 0$

} initialization

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

} ***relaxation 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



Bellman-Ford algorithm

for each $v \in V$
    do $d[v] \leftarrow \infty$
$d[s] \leftarrow 0$

} initialization

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

} ***relaxation 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 $= O(VE)$.

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 $v \in V$.

Theorem. If $G = (V, E)$ contains no negative-weight cycles, then after k iterations of Bellman-Ford algorithm every node knows the shortest path from v that uses at most k edges and the distance of this path.

All-pairs shortest paths

Input: Digraph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, with edge-weight function $w : E \to \mathbb{R}$.

Output: $n \times n$ matrix of shortest-path lengths $\delta(i, j)$ for all $i, j \in V$.

All-pairs shortest paths

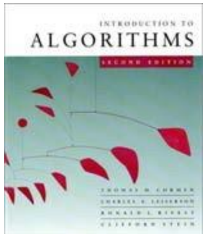
Input: Digraph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, with edge-weight function $w : E \to \mathbb{R}$.

Output: $n \times n$ matrix of shortest-path lengths $\delta(i, j)$ for all $i, j \in V$.

IDEA:

- Run Bellman-Ford once from each vertex.
- Time $= O(V^2E)$.
- Dense graph (V^2 edges) $\Rightarrow \Theta(V^4)$ time in the worst case.

Good first try!



Pseudocode for Floyd-Warshall

for $k \leftarrow 1$ **to** n
    do for $i \leftarrow 1$ **to** n
        do for $j \leftarrow 1$ **to** n
            do if $d_{ij} > d_{ik} + d_{kj}$
                then $d_{ij} \leftarrow d_{ik} + d_{kj}$

} ***relaxation***

Notes:
- Runs in $\Theta(V^3)$ time.
- Simple to code.
- Efficient in practice.