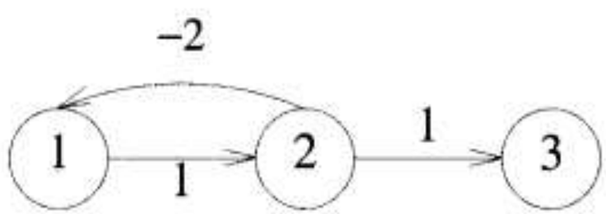
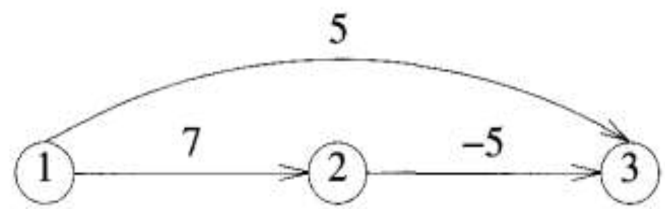
Bellman-Ford algorithm - Single-source shortest paths

Bellman-Ford algorithm

- Single-source shortest paths
- Handling negative-weight cycles and negative edges



Graph with negative cycle



Directed graph with a negative-length edge

Bellman & Ford

Richard E. Bellman
(1920–1984)
IEEE Medal of Honor, 1979

http://www.amazon.com/Bellman-Continuum-Collection-Works-Richard/dp/9971500906

Lester R. Ford, Jr.
(1927–)
president of MAA, 1947–48

http://www.maa.org/aboutmaa/maaapresidents.html

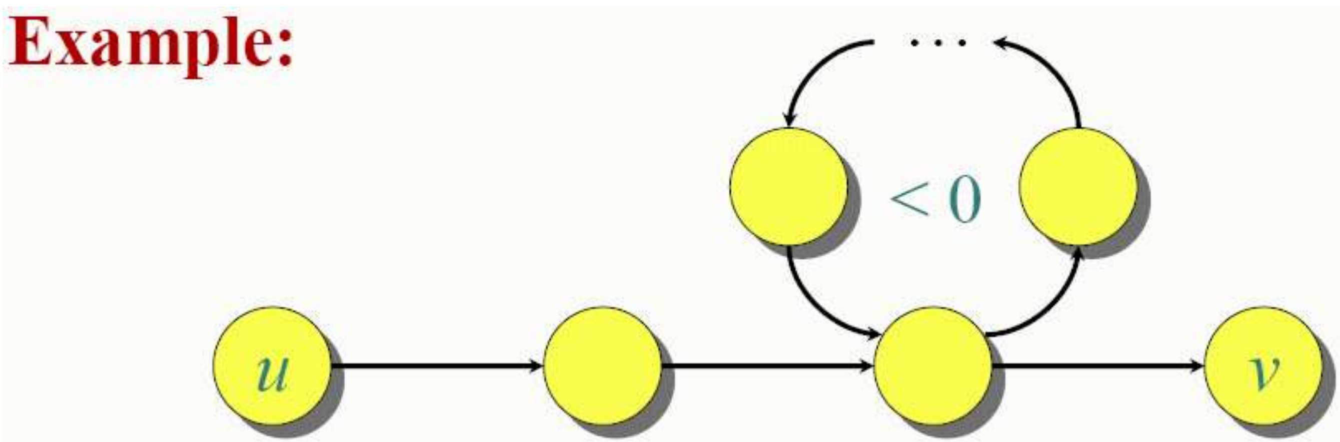
Bellman-Ford in Practice

- Distance-vector routing protocol
  - Repeatedly relax edges until convergence
  - Relaxation is local!
- On the Internet:
  - Routing Information Protocol (RIP)
  - Interior Gateway Routing Protocol (IGRP)

photo by Ross Imlach, 2011
http://www.flickr.com/photos/rossimlach/5446205998/

Negative-weight cycles

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

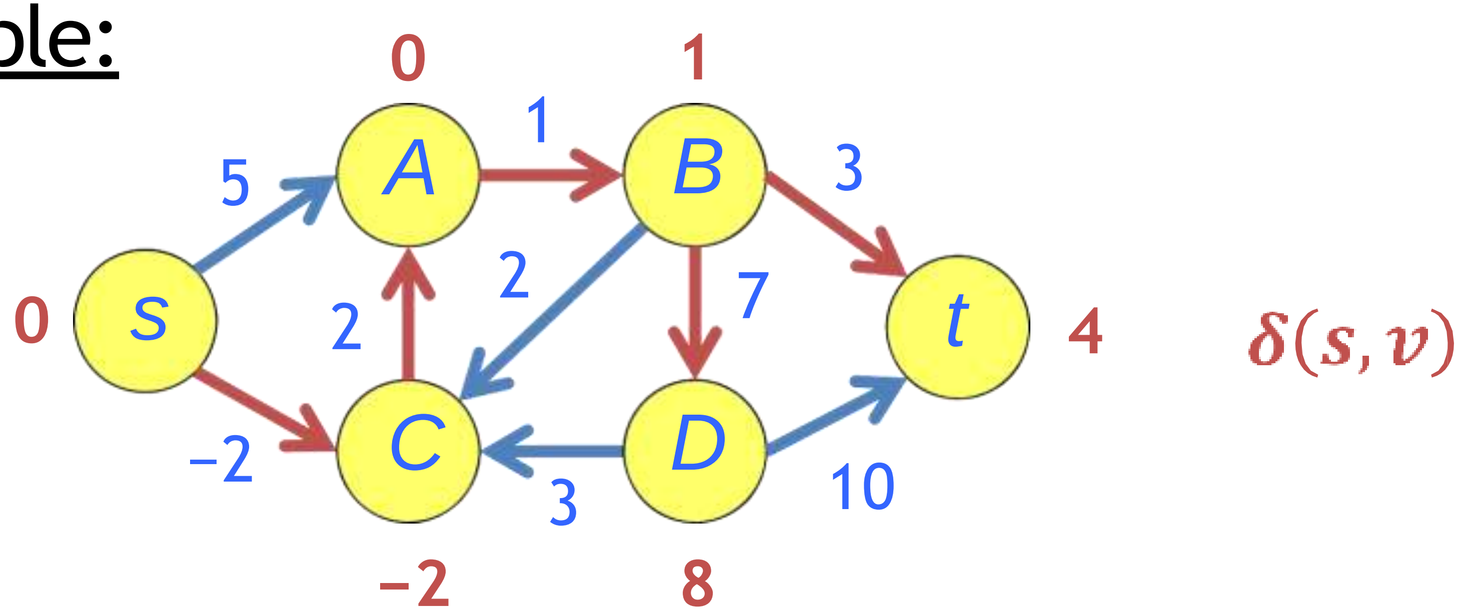


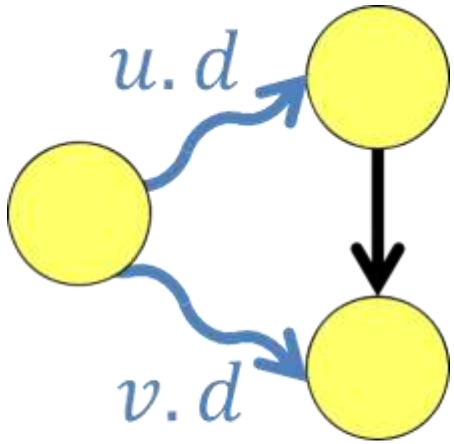
Bellman-Ford algorithm: Finds all shortest-path lengths from a ***source*** $s \in V$ to all $v \in V$ or determines that a negative-weight cycle exists.

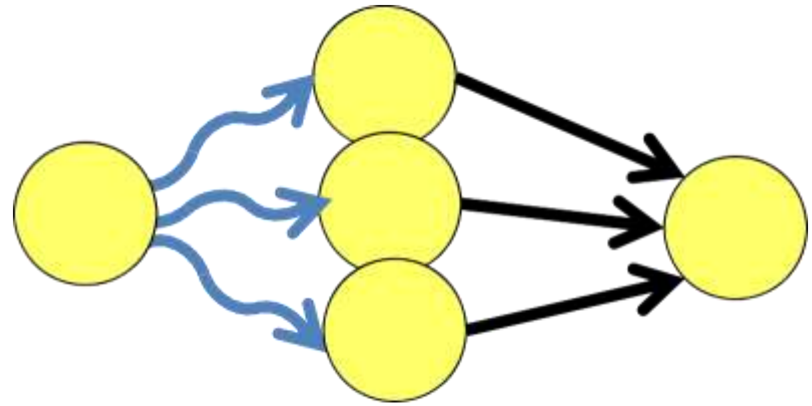
Recall: Single-Source Shortest Paths

- Problem: Given a directed graph $G = (V, E)$ with edge-weight function $w : E \to \mathbb{R}$, and a ***source*** vertex s, compute $\delta(s, v)$ for all $v \in V$
  - Also want shortest-path tree represented by $v.\pi$

Example:







When there are no cycles of negative length, there is a shortest path between any two vertices of an n-vertex graph that has at most n -1 edges on it

a path that has more than n – 1 edges must repeat at least one vertex and hence must contain a cycle.

Let $dist^{\ell}[u]$ be the length of a shortest path from the source vertex v to vertex u under the constraint that the shortest path contains at most ℓ edges. Then, $dist^1[u] = cost[v, u]$, $1 \leq u \leq n$. As noted earlier, when there are no cycles of negative length, we can limit our search for shortest paths to paths with at most $n - 1$ edges. Hence, $dist^{n-1}[u]$ is the length of an unrestricted shortest path from v to u.

Our goal then is to compute $dist^{n-1}[u]$ for all u. This can be done using the dynamic programming methodology. First, we make the following observations:

1. If the shortest path from v to u with at most k, $k > 1$, edges has no more than $k-1$ edges, then $dist^k[u] = dist^{k-1}[u]$.

2. If the shortest path from v to u with at most k, $k > 1$, edges has exactly k edges, then it is made up of a shortest path from v to some vertex j followed by the edge $\langle j, u \rangle$. The path from v to j has $k-1$ edges, and its length is $dist^{k-1}[j]$. All vertices i such that the edge $\langle i, u \rangle$ is in the graph are candidates for j. Since we are interested in a shortest path, the i that minimizes $dist^{k-1}[i] + cost[i, u]$ is the correct value for j.

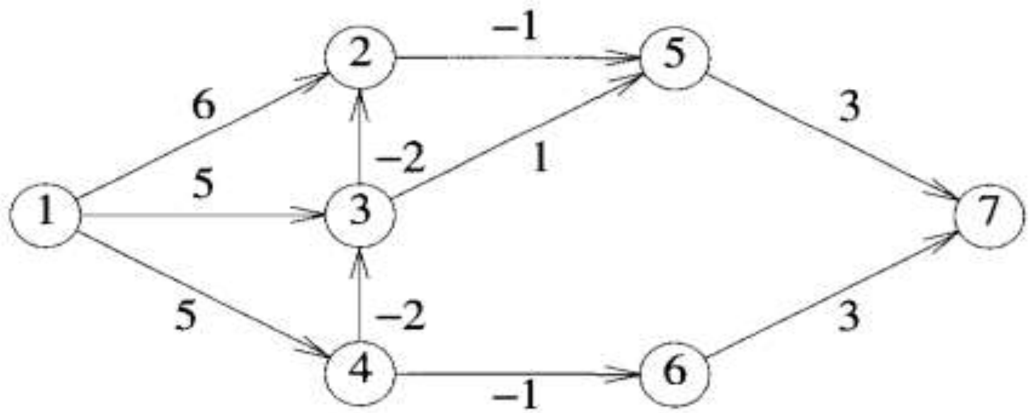
These observations result in the following recurrence for $dist$:

$$dist^k[u] = \min \{dist^{k-1}[u], \min_i \{dist^{k-1}[i] + cost[i, u]\}\}$$

This recurrence can be used to compute $dist^k$ from $dist^{k-1}$, for $k = 2, 3, \ldots, n-1$.

```
Algorithm BellmanFord(v, cost, dist, n)
// Single-source/all-destinations shortest
// paths with negative edge costs
{
    for i := 1 to n do // Initialize dist.
        dist[i] := cost[v, i];
    for k := 2 to n - 1 do
        for each u such that u ≠ v and u has
                at least one incoming edge do
            for each ⟨i, u⟩ in the graph do
                if dist[u] > dist[i] + cost[i, u] then
                    dist[u] := dist[i] + cost[i, u];
}
```

Bellman and Ford algorithm to compute shortest paths

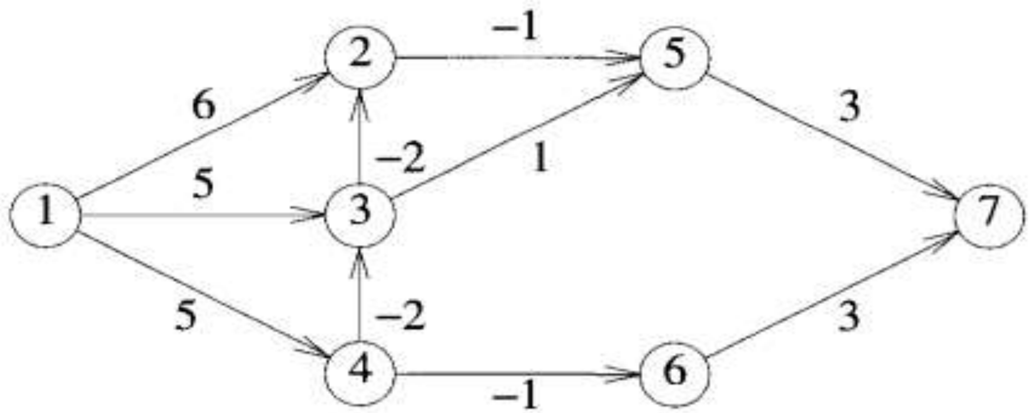


(a) A directed graph

k	$dist^k[1..7]$						
	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	0	3	3	5	5	4	∞
3	0	1	3	5	2	4	7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3

(b) $dist^k$

$$dist^k[u] \; = \; \min \; \{dist^{k-1}[u], \; \min_i \; \{dist^{k-1}[i] \; + \; cost[i,u]\}\} \quad k = 2, 3, \ldots, \; n-1.$$

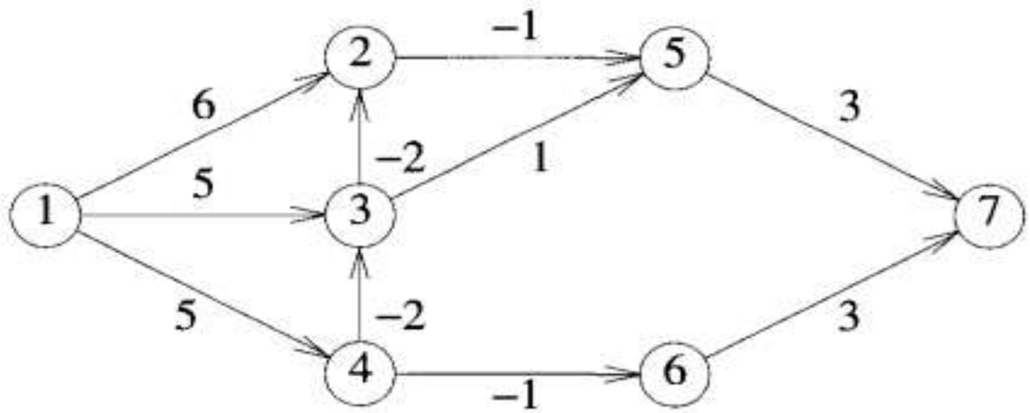


(a) A directed graph

k	$dist^k[1..7]$						
	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	0	3	3	5	5	4	∞
3	0	1	3	5	2	4	7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3

(b) $dist^k$

$$dist^k[u] \;=\; \min\ \{dist^{k-1}[u],\ \min_i\ \{dist^{k-1}[i]\ +\ cost[i,u]\}\}\quad k = 2, 3, \ldots,\ n-1.$$

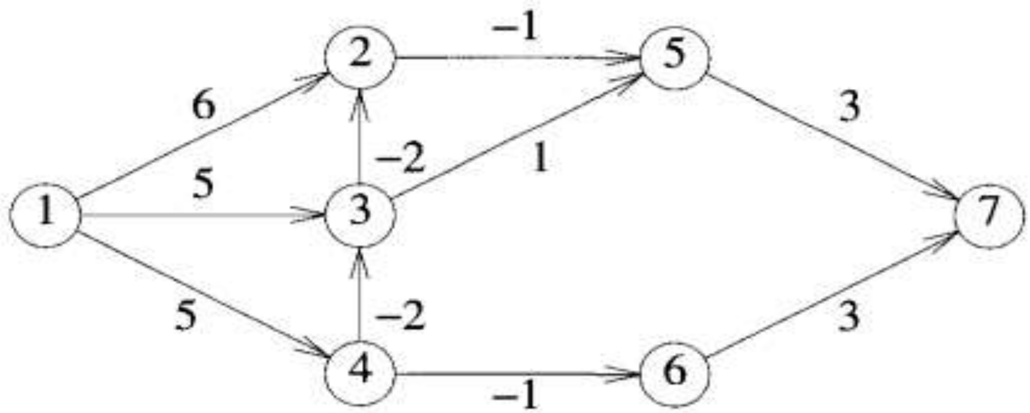


(a) A directed graph

k	$dist^k[1..7]$						
	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	0	3	3	5	5	4	∞
3	0	1	3	5	2	4	7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3

(b) $dist^k$

$$dist^k[u] = \min \{dist^{k-1}[u], \min_i \{dist^{k-1}[i] + cost[i,u]\}\} \quad k = 2, 3, \ldots, n-1.$$

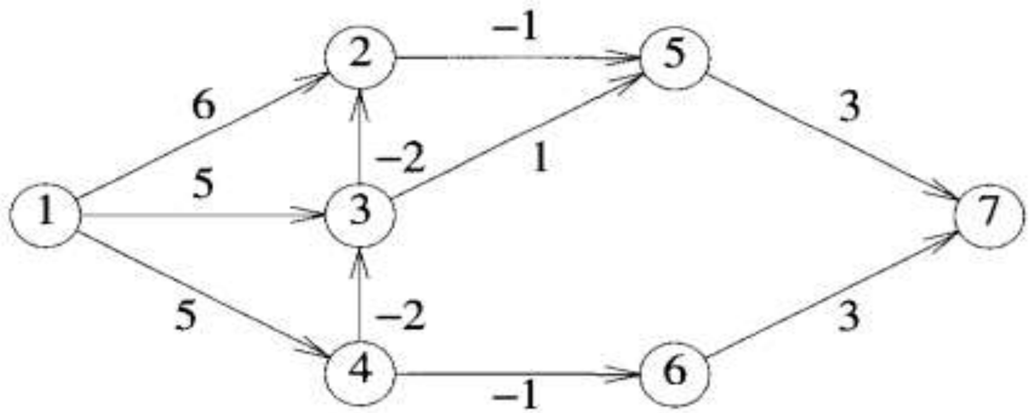


(a) A directed graph

k	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	0	3	3	5	5	4	∞
3	0	1	3	5	2	4	7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3

(b) $dist^k$

$$dist^k[u] = \min \{dist^{k-1}[u], \min_i \{dist^{k-1}[i] + cost[i,u]\}\} \quad k = 2, 3, \ldots, n-1.$$



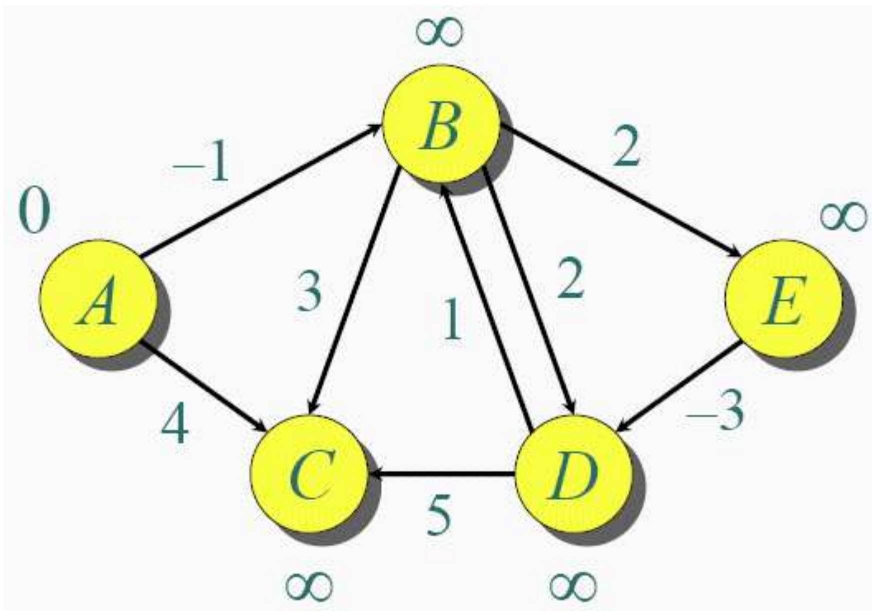
(a) A directed graph

k	$dist^k[1..7]$						
	1	2	3	4	5	6	7
1	0	6	5	5	∞	∞	∞
2	0	3	3	5	5	4	∞
3	0	1	3	5	2	4	7
4	0	1	3	5	0	4	5
5	0	1	3	5	0	4	3
6	0	1	3	5	0	4	3

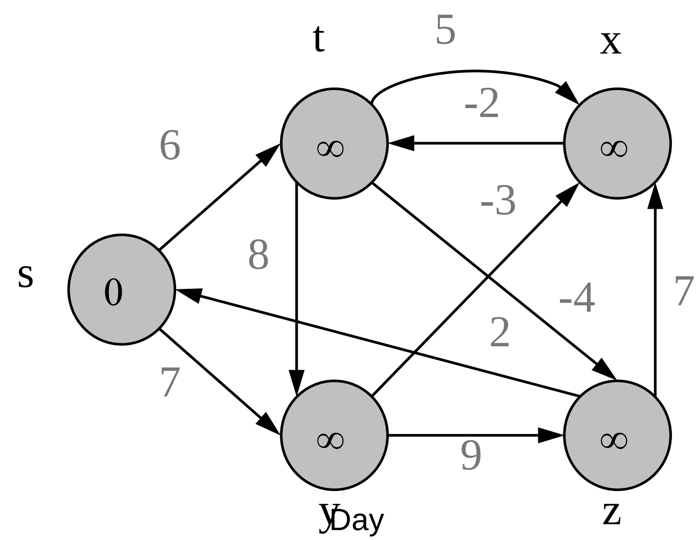
(b) $dist^k$

$$dist^k[u] \; = \; \min \; \{dist^{k-1}[u], \; \min_i \; \{dist^{k-1}[i] \; + \; cost[i,u]\}\} \quad k = 2, 3, \ldots, \; n-1.$$

Example of Bellman-Ford



A	B	C	D	E
0	∞	∞	∞	∞



$$dist^k[u] = \min \{dist^{k-1}[u], \min_i \{dist^{k-1}[i] + cost[i,u]\}\} \quad k = 2, 3, \ldots, n-1.$$

	A	B	C	D	E
A	0	-1	4	∞	∞
B	∞	0	3	2	2
C	∞	∞	0	∞	∞
D	∞	1	5	0	∞
E	∞	∞	∞	-3	0

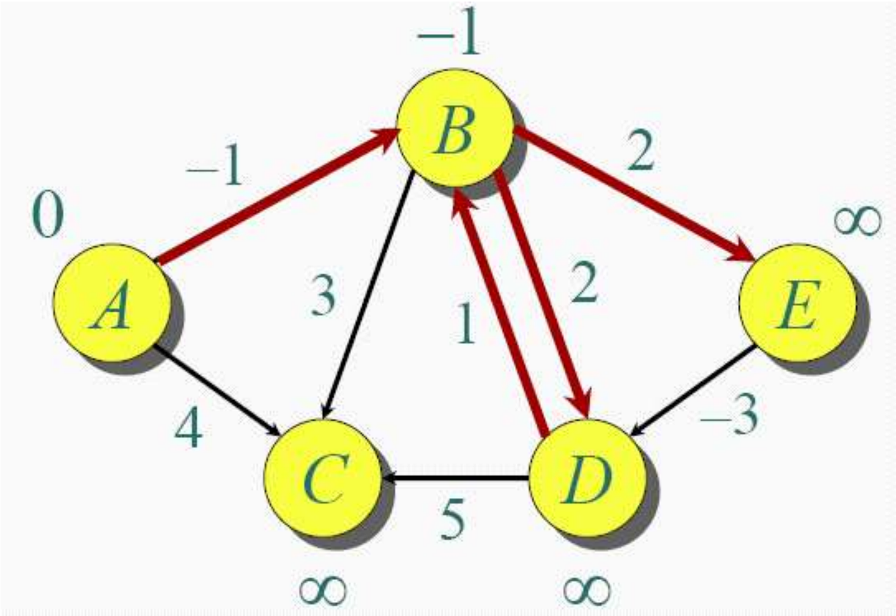
	A	B	C	D	E
A	0	-1	4	∞	∞
B	∞	0	3	2	2
C	∞	∞	0	∞	∞
D	∞	1	5	0	∞
E	∞	∞	∞	-3	0

	A	B	C	D	E
A	0	-1	4	∞	∞
B	∞	0	3	2	2
C	∞	∞	0	∞	∞
D	∞	1	5	0	∞
E	∞	∞	∞	-3	0

	A	B	C	D	E
A	0	-1	4	∞	∞
B	∞	0	3	2	2
C	∞	∞	0	∞	∞
D	∞	1	5	0	∞
E	∞	∞	∞	-1	0

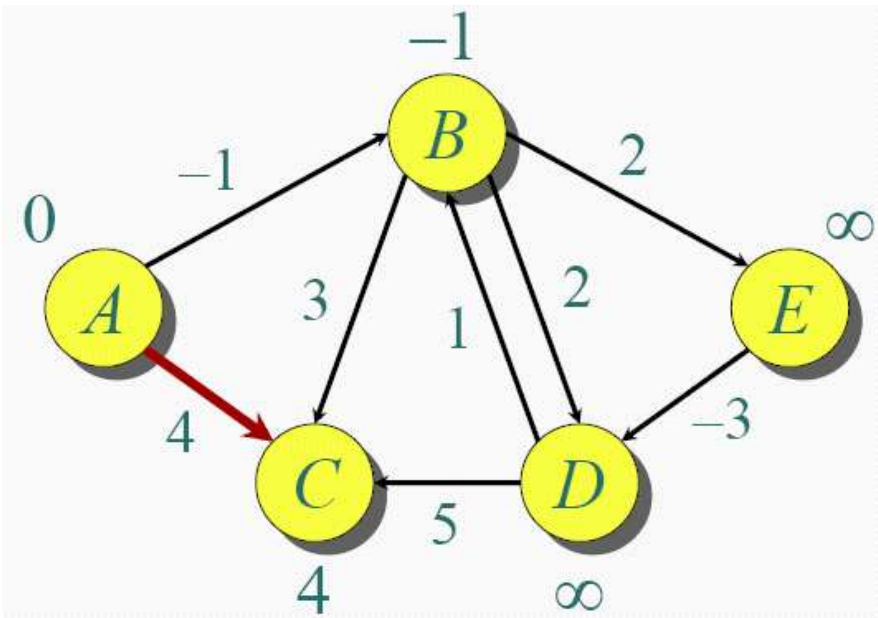
	A	B	C	D	E
A	0	-1	4	∞	∞
B	∞	0	3	2	2
C	∞	∞	0	∞	∞
D	∞	1	5	0	∞
E	∞	∞	∞	-3	0

Example of Bellman-Ford



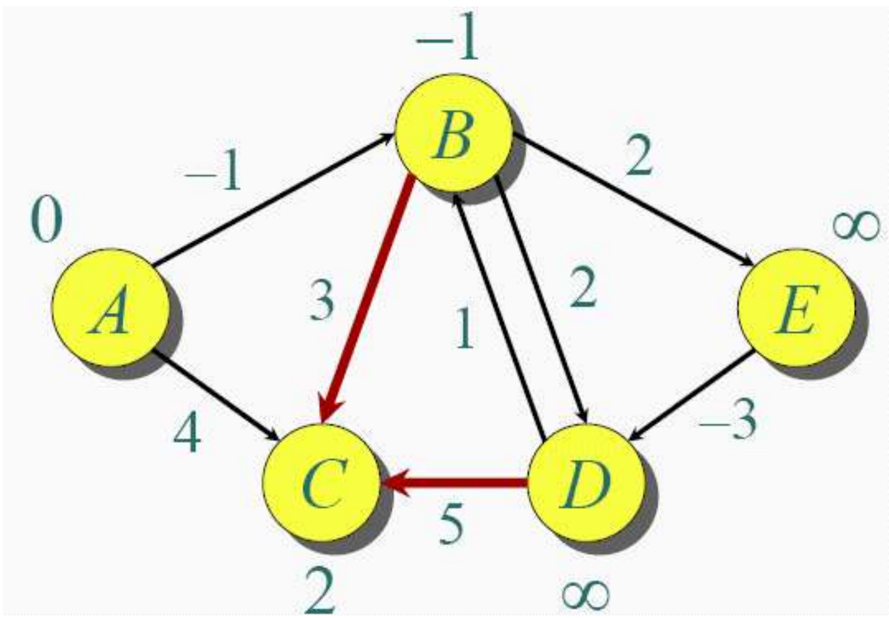
A	B	C	D	E
0	∞	∞	∞	∞
0	-1	∞	∞	∞

Example of Bellman-Ford



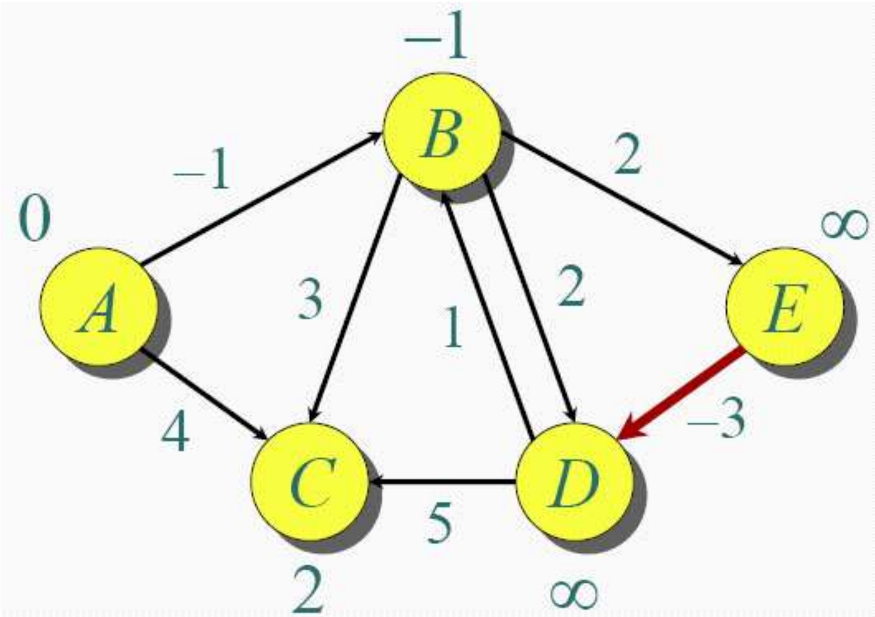
A	B	C	D	E
0	∞	∞	∞	∞
0	−1	∞	∞	∞
0	−1	4	∞	∞

Example of Bellman-Ford



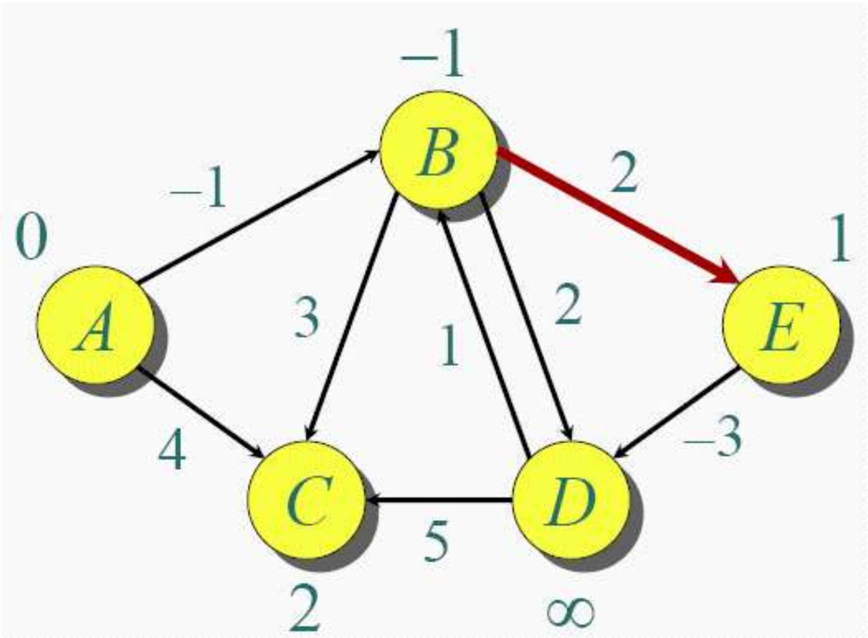
A	B	C	D	E
0	∞	∞	∞	∞
0	–1	∞	∞	∞
0	–1	4	∞	∞
0	–1	2	∞	∞

Example of Bellman-Ford



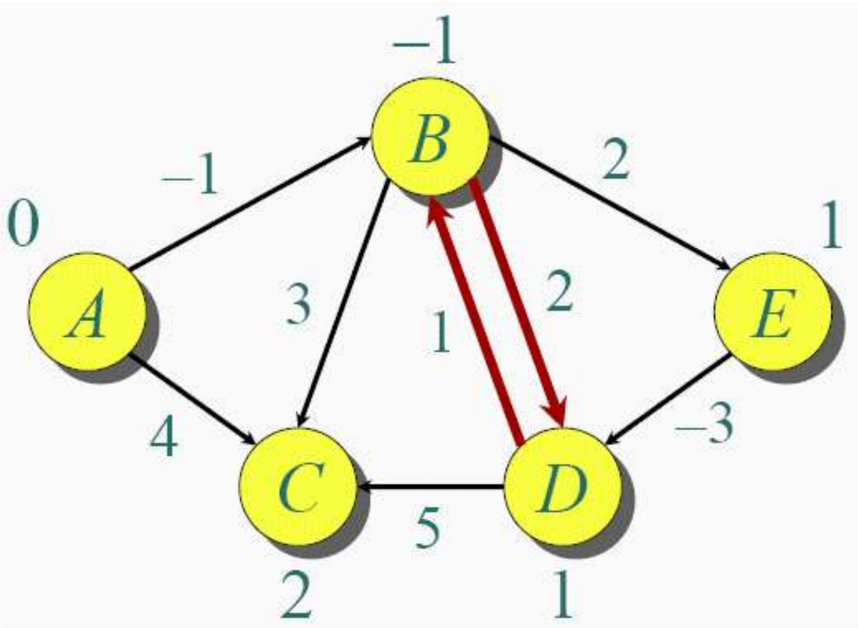
A	B	C	D	E
0	∞	∞	∞	∞
0	–1	∞	∞	∞
0	–1	4	∞	∞
0	–1	2	∞	∞

Example of Bellman-Ford



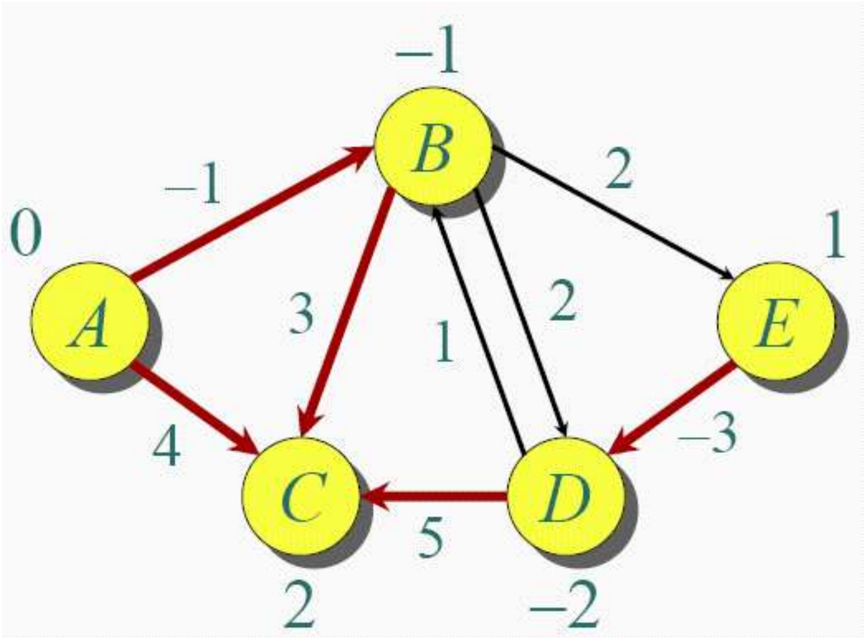
A	B	C	D	E
0	∞	∞	∞	∞
0	−1	∞	∞	∞
0	−1	4	∞	∞
0	−1	2	∞	∞
0	−1	2	∞	1

Example of Bellman-Ford



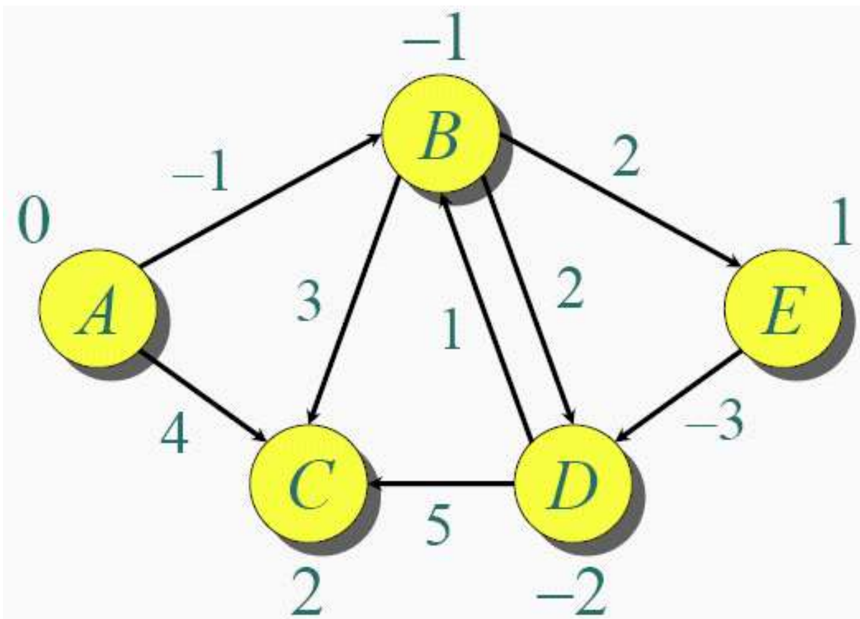
A	B	C	D	E
0	∞	∞	∞	∞
0	−1	∞	∞	∞
0	−1	4	∞	∞
0	−1	2	∞	∞
0	−1	2	∞	1
0	−1	2	1	1

Example of Bellman-Ford



A	B	C	D	E
0	∞	∞	∞	∞
0	–1	∞	∞	∞
0	–1	4	∞	∞
0	–1	2	∞	∞
0	–1	2	∞	1
0	–1	2	1	1
0	–1	2	–2	1

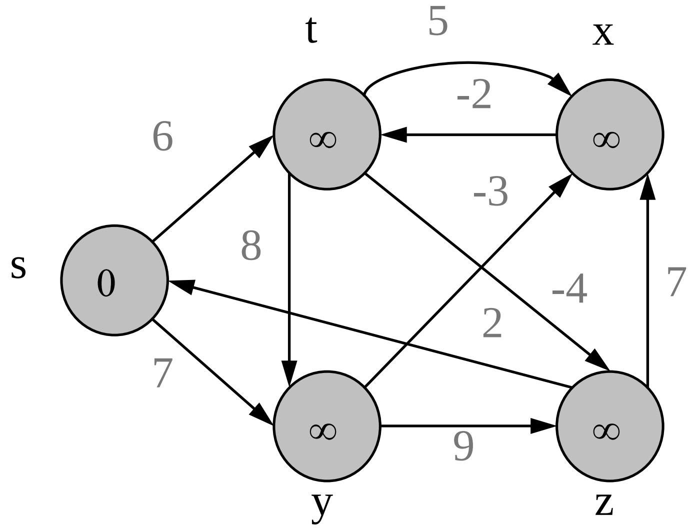
Example of Bellman-Ford



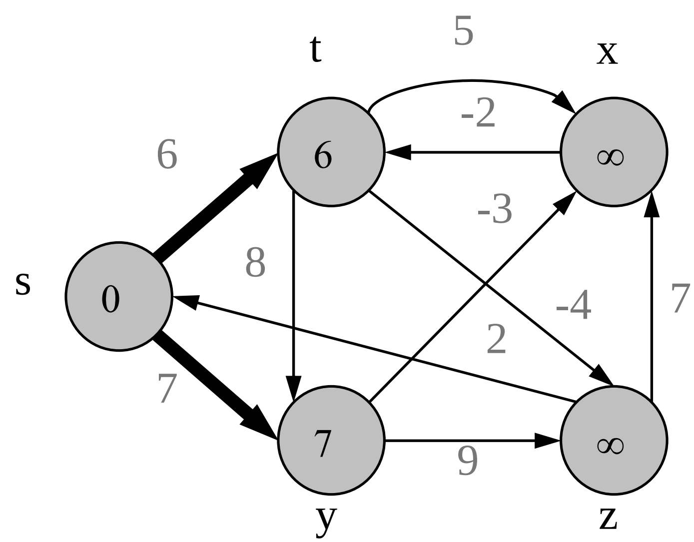
A	B	C	D	E
0	∞	∞	∞	∞
0	–1	∞	∞	∞
0	–1	4	∞	∞
0	–1	2	∞	∞
0	–1	2	∞	1
0	–1	2	1	1
0	–1	2	–2	1

Note: Values decrease monotonically.

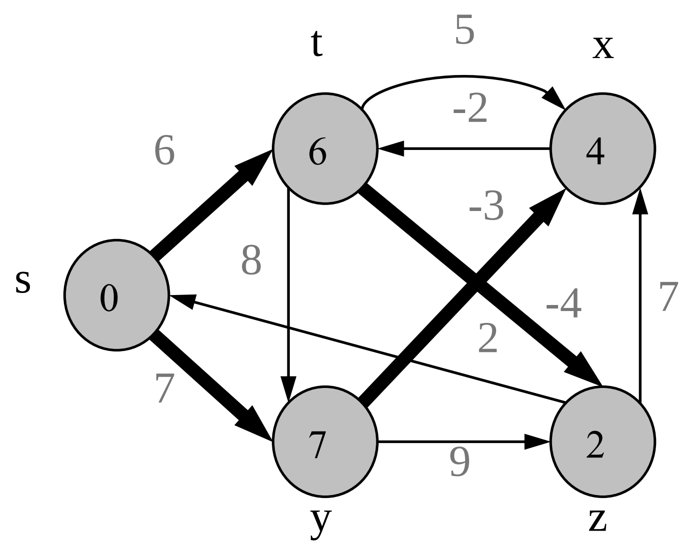
Bellman-Ford Algorithm Example



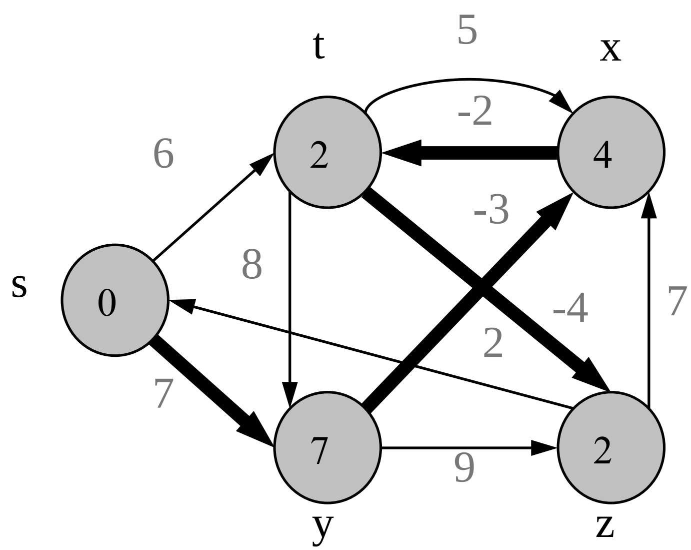
Bellman-Ford Algorithm Example



Bellman-Ford Algorithm Example



Bellman-Ford Algorithm Example



Bellman-Ford Algorithm Example