THE SHORTEST PATH PROBLEM

CMP-410-3: Data Structures and Algorithms

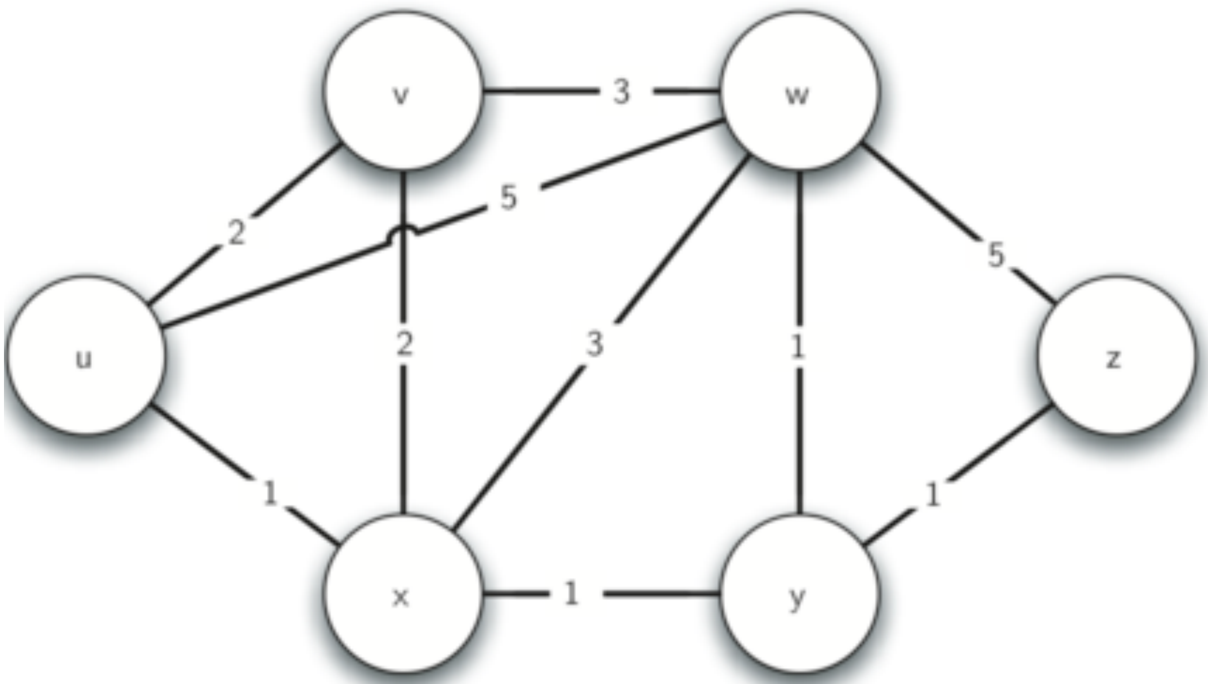
Waheed Iqbal

Punjab University College of Information Technology (PUCIT)
University of the Punjab, Lahore, Pakistan.

Introduction

- Consider the problem of finding the shortest path between nodes s and t in a graph (directed or undirected).

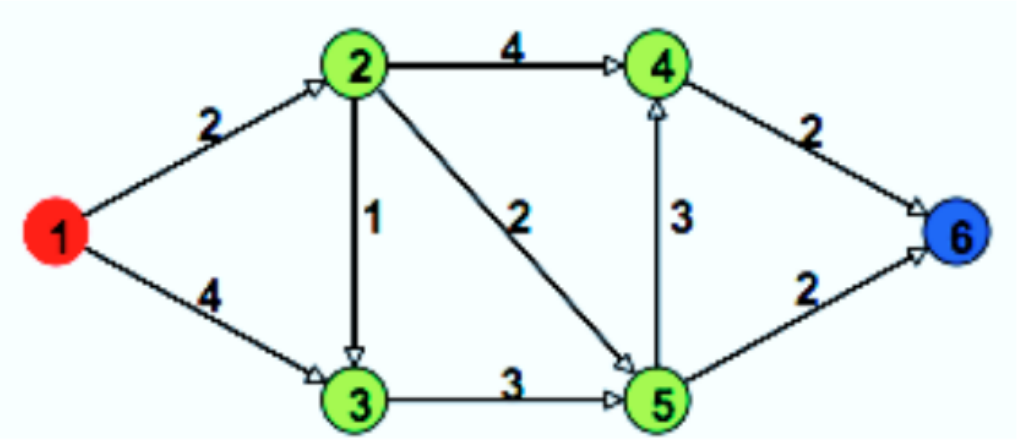
- We already know BFS but how about if edges have weights. Consider the following:



DIJKSTRA'S ALGORITHM

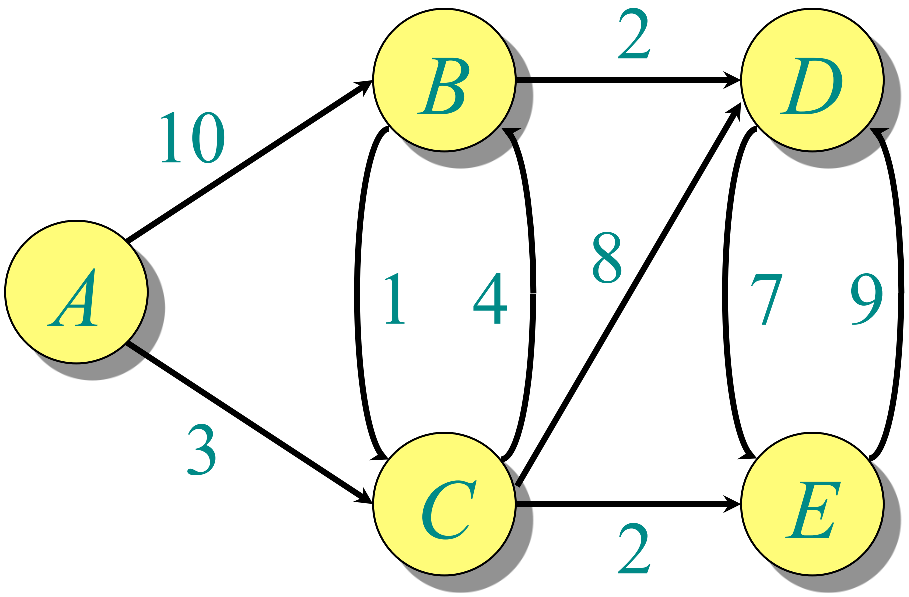
Dijkstra’s Algorithm

- One of a solution to the single-source shortest path problem in graph theory
  - Both directed and undirected graphs
  - All edges must have nonnegative weights
  - Graph must be connected



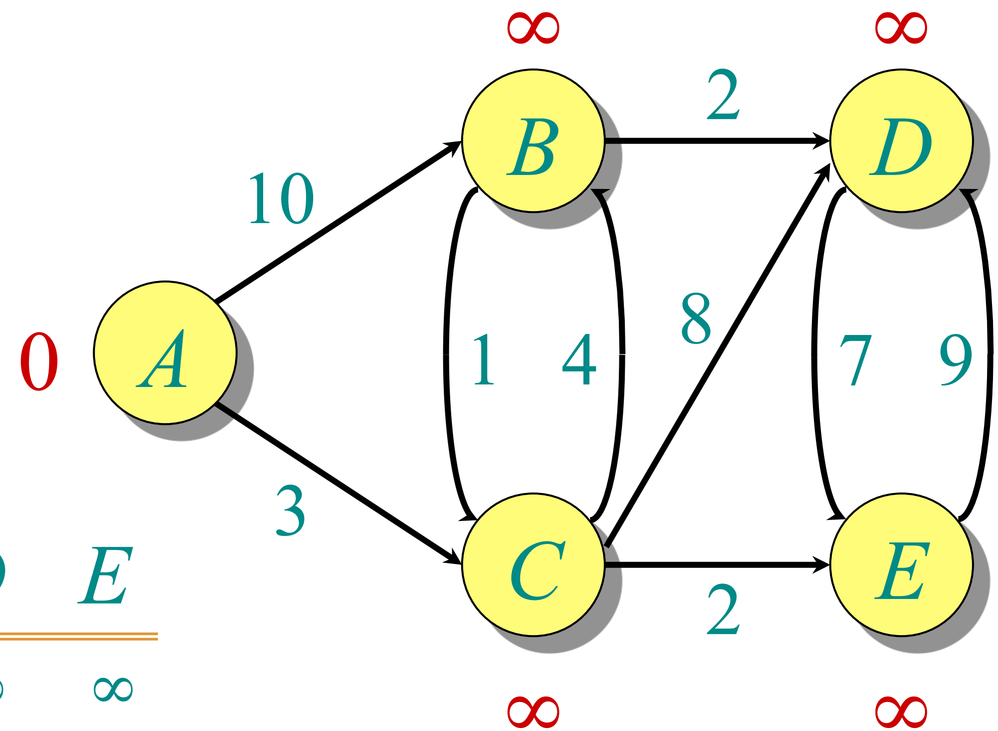
Example of Dijkstra’s algorithm

Graph with nonnegative edge weights:



Example of Dijkstra's algorithm

Initialize:

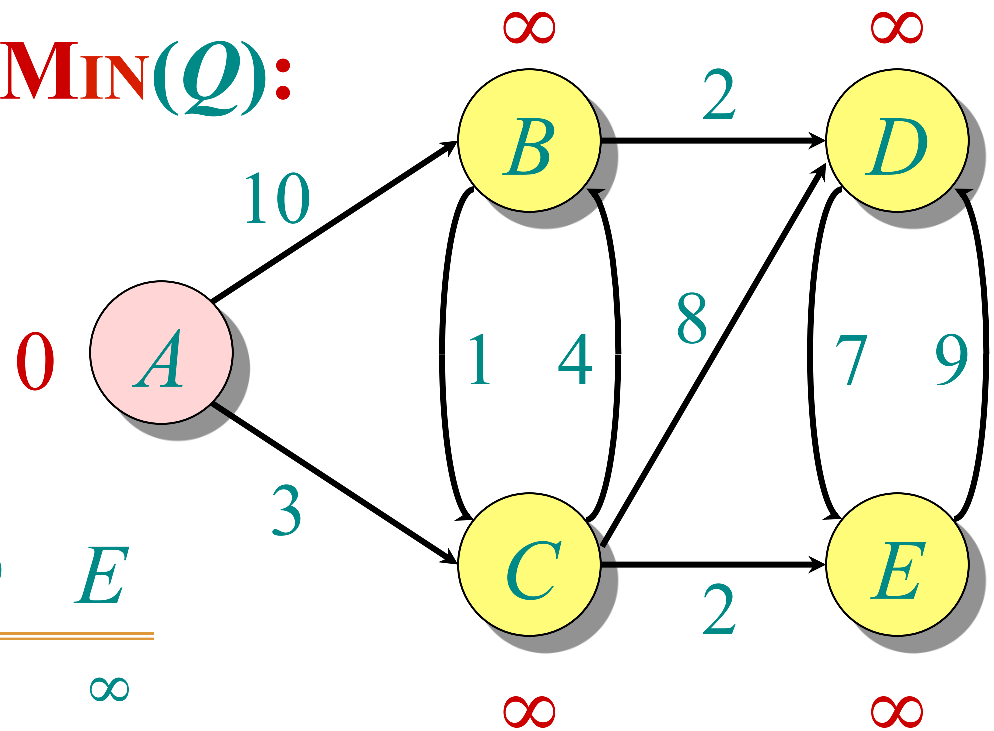


Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: { }

Example of Dijkstra's algorithm

"A" ← **EXTRACT-MIN**(Q):

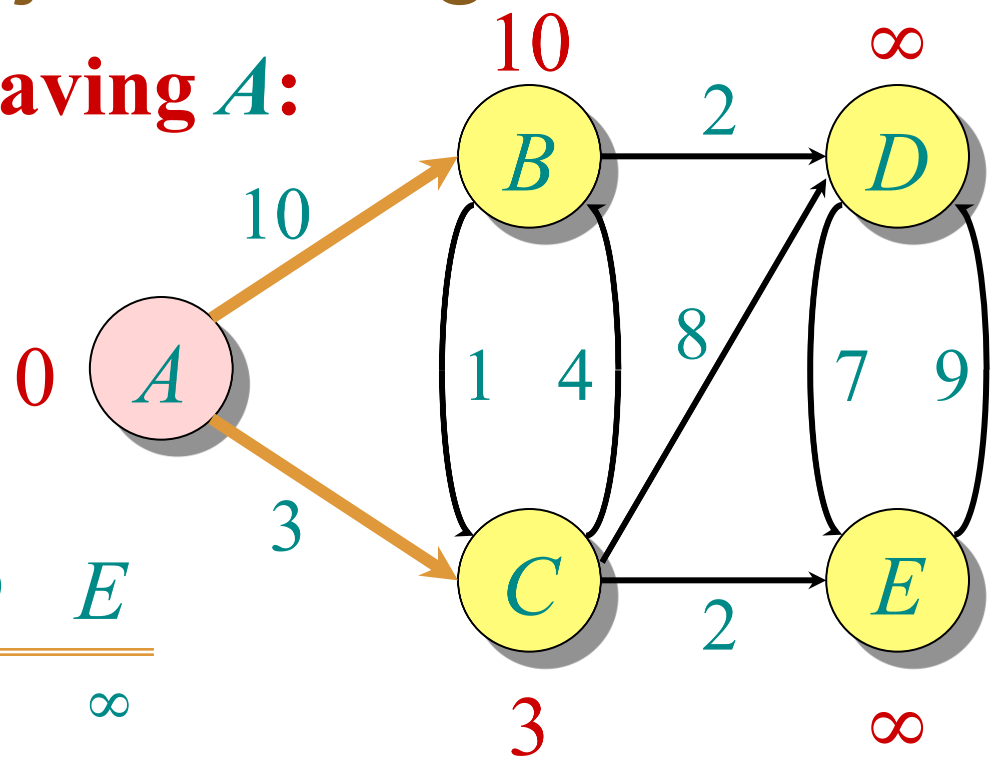


Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: { A }

Example of Dijkstra's algorithm

Relax all edges leaving *A*:

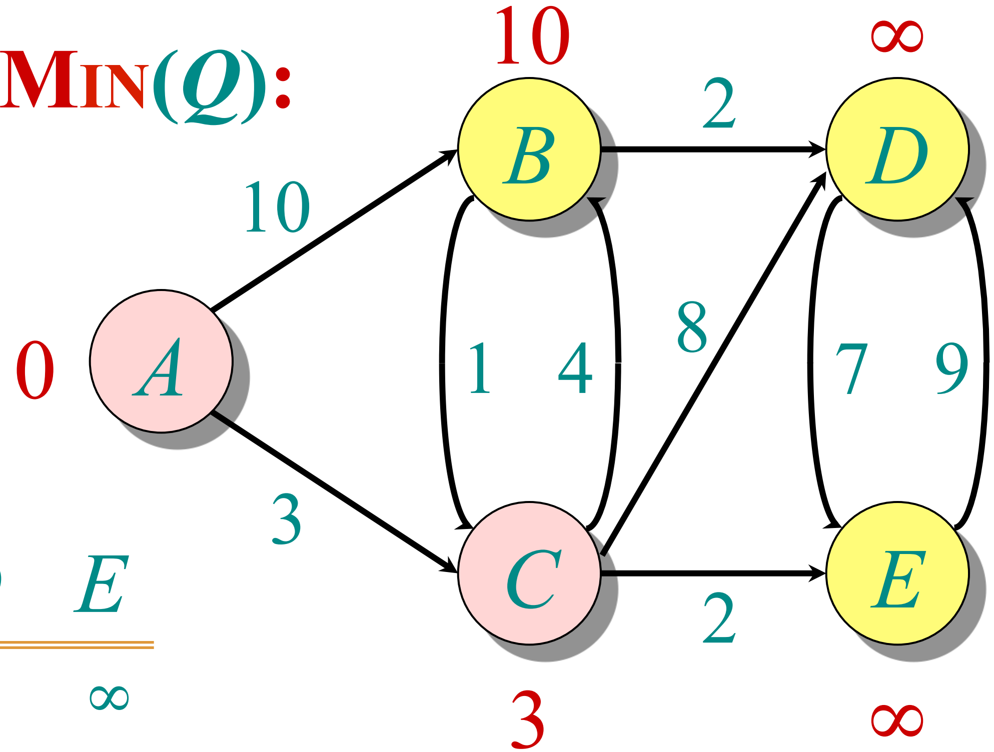


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A }

Example of Dijkstra's algorithm

"C" ← EXTRACT-MIN(Q):

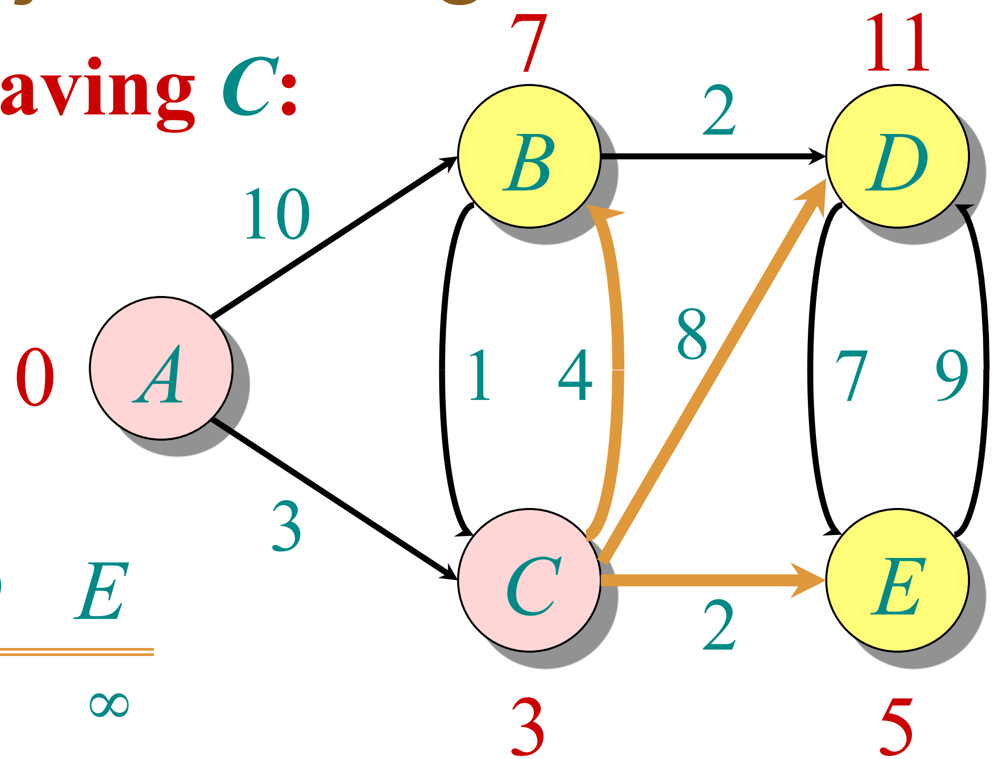


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A, C }

Example of Dijkstra's algorithm

Relax all edges leaving *C*:

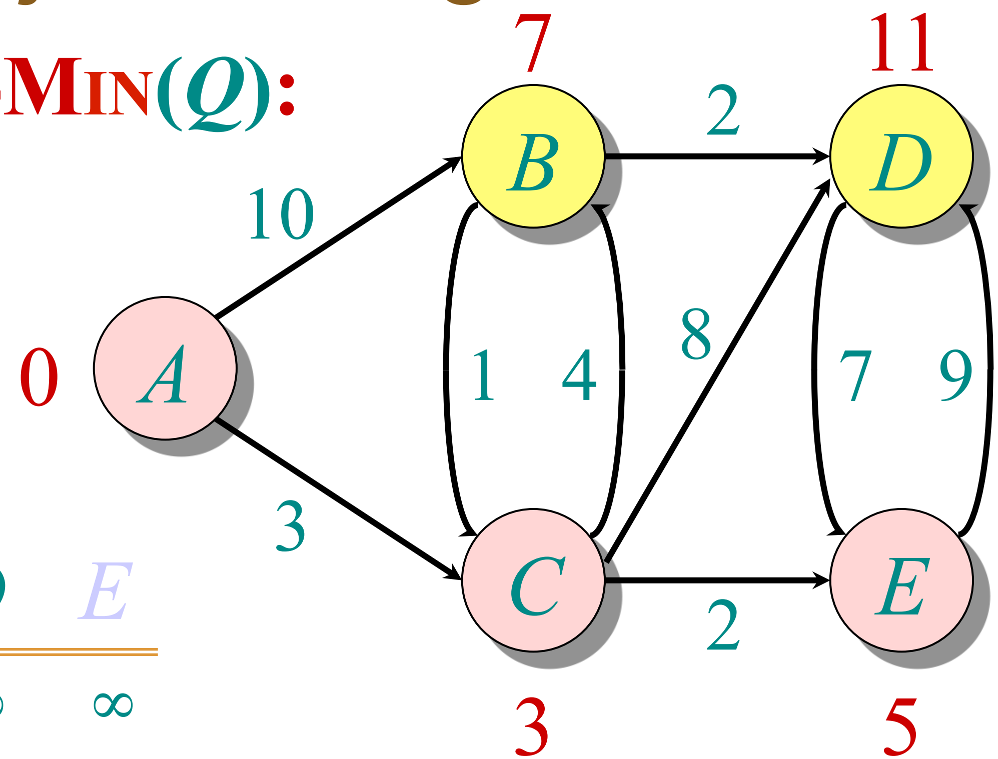


Q:	*A*	*B*	*C*	*D*	*E*
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { *A*, *C* }

Example of Dijkstra's algorithm

"*E*" ← EXTRACT-MIN(*Q*):

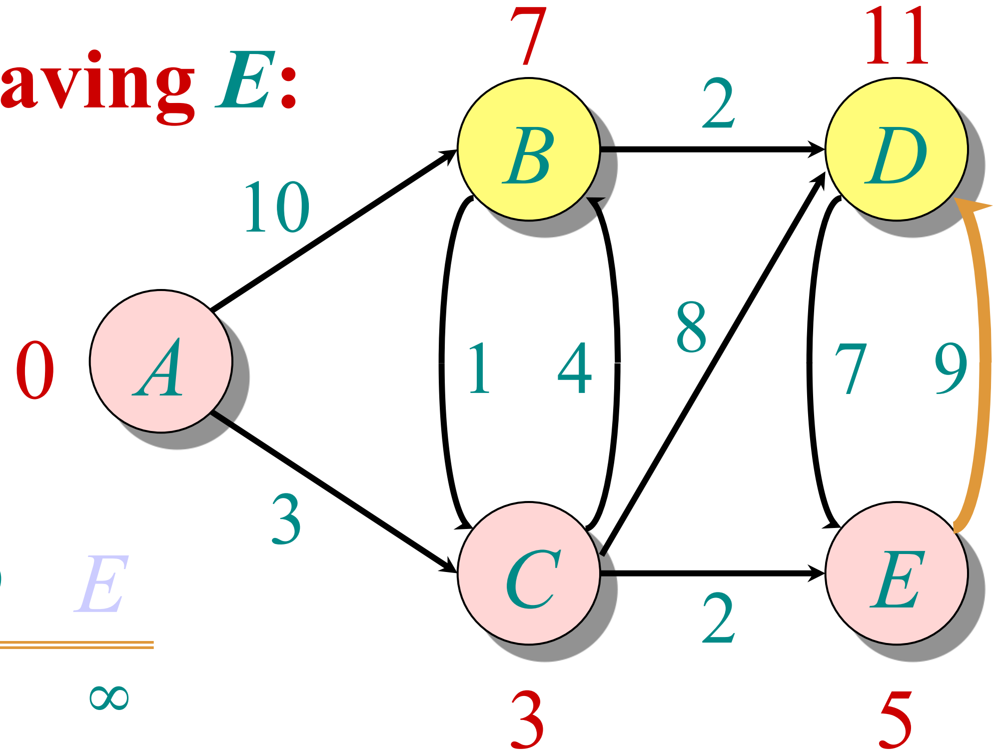


Q:	*A*	*B*	*C*	*D*	*E*
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { *A*, *C*, *E* }

Example of Dijkstra's algorithm

Relax all edges leaving *E*:

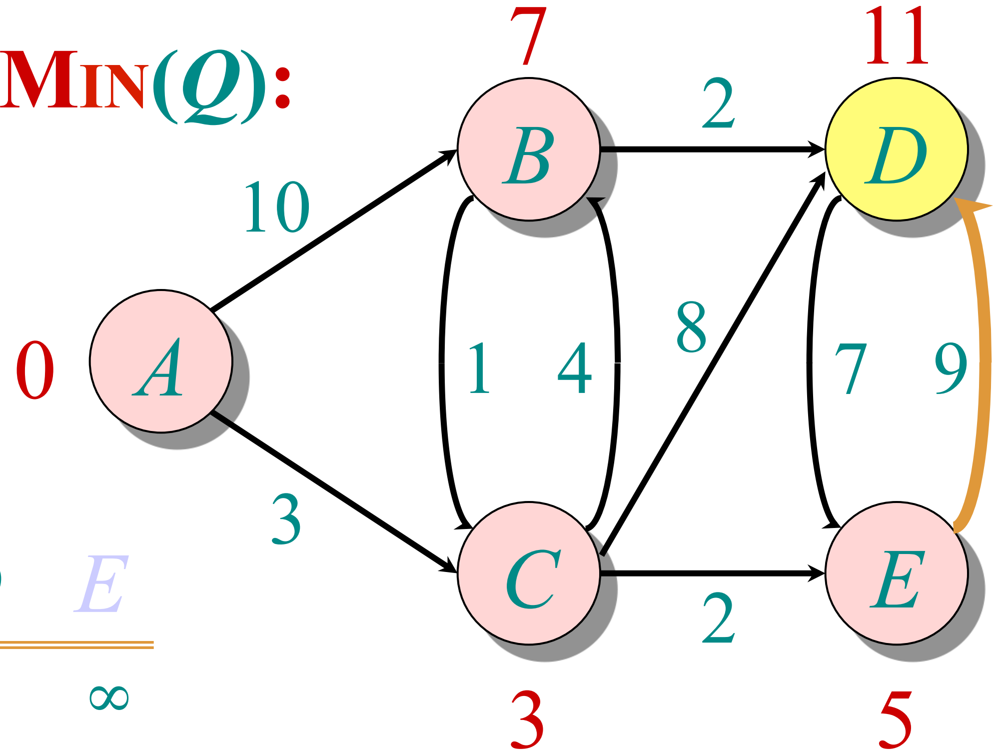


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E }

Example of Dijkstra's algorithm

"B" ← **EXTRACT-MIN**(Q):

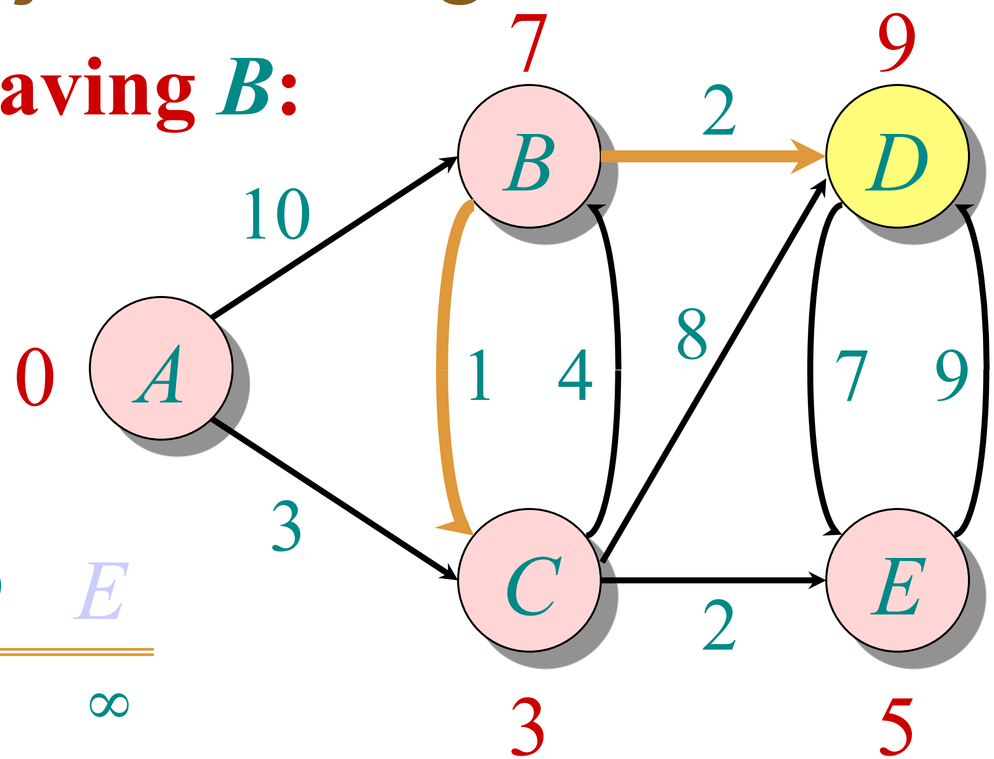


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E, B }

Example of Dijkstra's algorithm

Relax all edges leaving *B*:

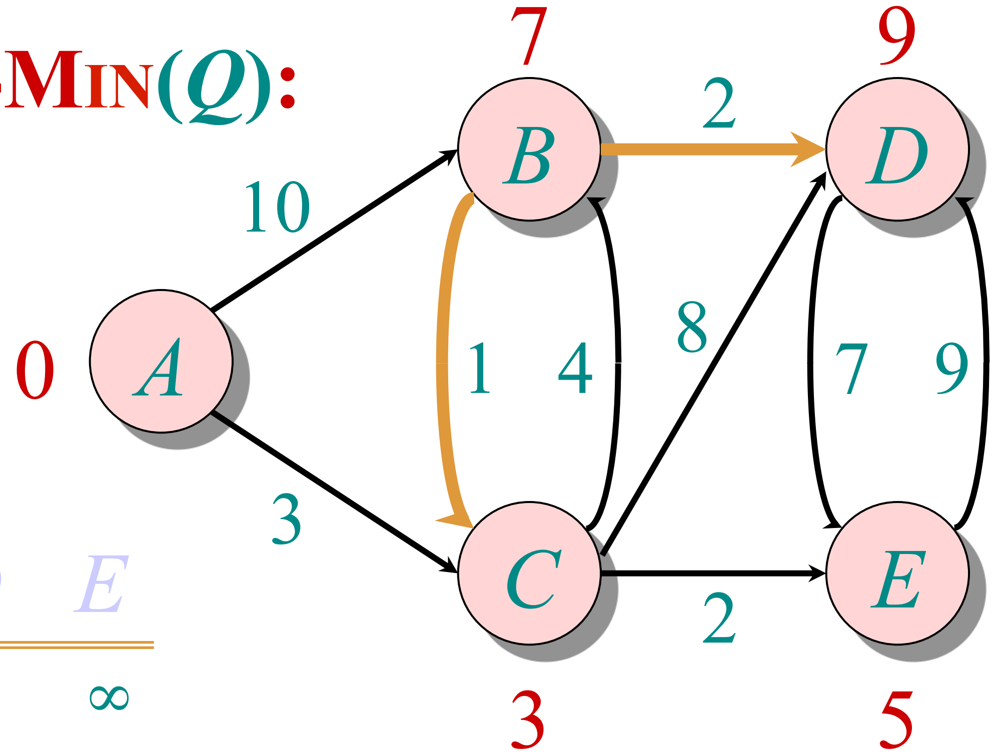


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B }

Example of Dijkstra's algorithm

"D" ← EXTRACT-MIN(Q):

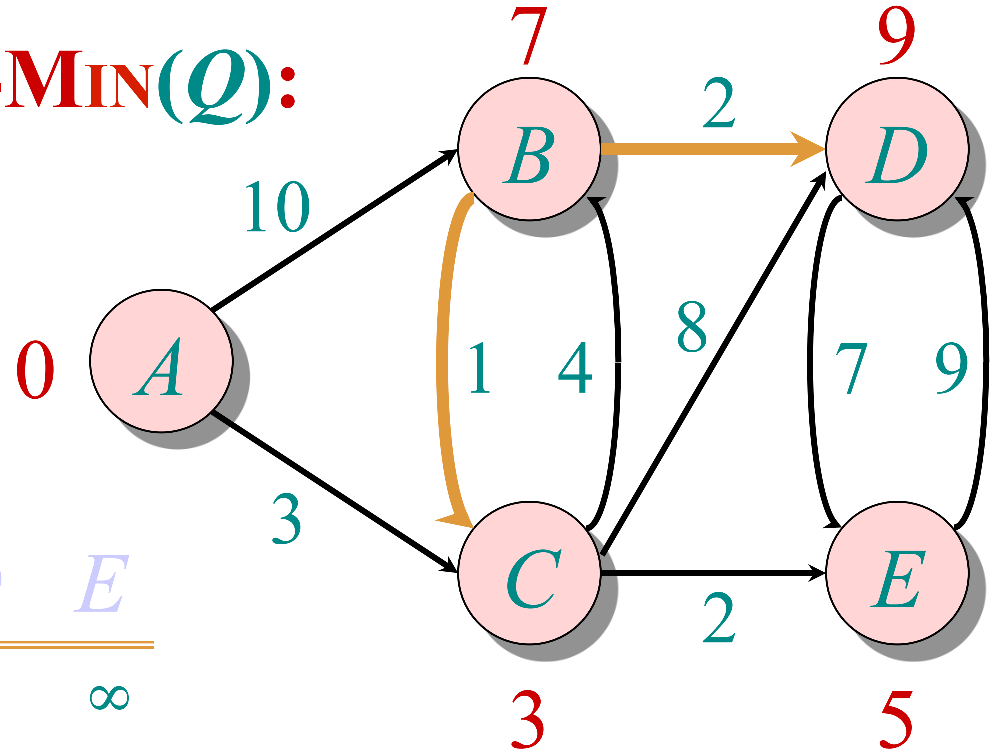


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B, D }

Example of Dijkstra's algorithm

"D" $\leftarrow$ **EXTRACT-MIN(Q):**



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B, D }

How do we now find out shortest path from source node A to any destination node?

Dijkstra's Algorithm

$d[s] \leftarrow 0$
for each $v \in V - \{s\}$
    do $d[v] \leftarrow \infty$
$S \leftarrow \varnothing$
$Q \leftarrow V$ ▷ Q is a priority queue maintaining $V - S$
while $Q \neq \varnothing$
    do $u \leftarrow$ EXTRACT-MIN(Q)
        $S \leftarrow S \cup \{u\}$
        for each $v \in Adj[u]$
            do if $d[v] > d[u] + w(u, v)$
                then $d[v] \leftarrow d[u] + w(u, v)$
                $p[v] \leftarrow u$

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 $u \leftarrow$ EXTRACT-MIN(Q)
        $S \leftarrow S \cup \{u\}$
        for each $v \in Adj[u]$
            do if $d[v] > d[u] + w(u, v)$
                then $d[v] \leftarrow d[u] + w(u, v)$
                $p[v] \leftarrow u$

Implicit DECREASE-KEY

BELLMAN-FORD ALGORITHM

Bellman-Ford Algorithm

- Bellman-Ford algorithm solves the single-source shortest-path problem in the general case in which
  - edges of a given digraph can have ***negative weight*** as long as G contains ***no negative cycles***.
- This algorithm, like Dijkstra's algorithm uses the notion of edge relaxation but does not use with greedy method.

Bellman-Ford Algorithm (Pseudo Code)

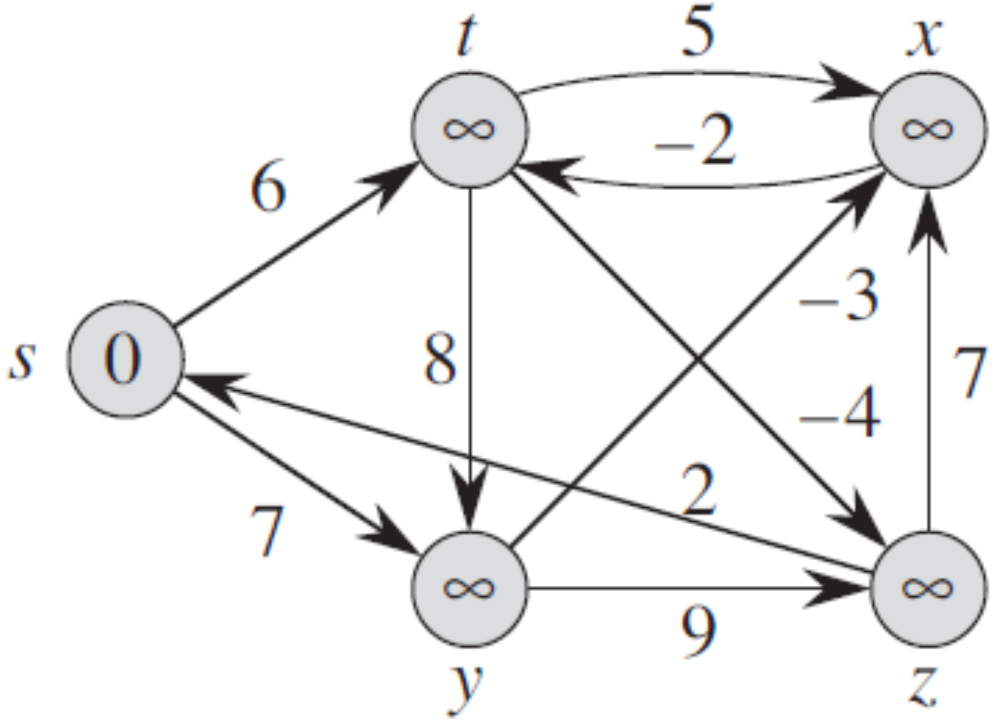
$$\left.\begin{array}{l} d[s] \leftarrow 0 \\ \textbf{for each } v \in V - \{s\} \\ \quad \textbf{do } d[v] \leftarrow \infty \end{array}\right\} \text{ initialization}$$

for $i \leftarrow 1$ **to** $|V| - 1$ **do**
  for each edge $(u, v) \in E$ **do**

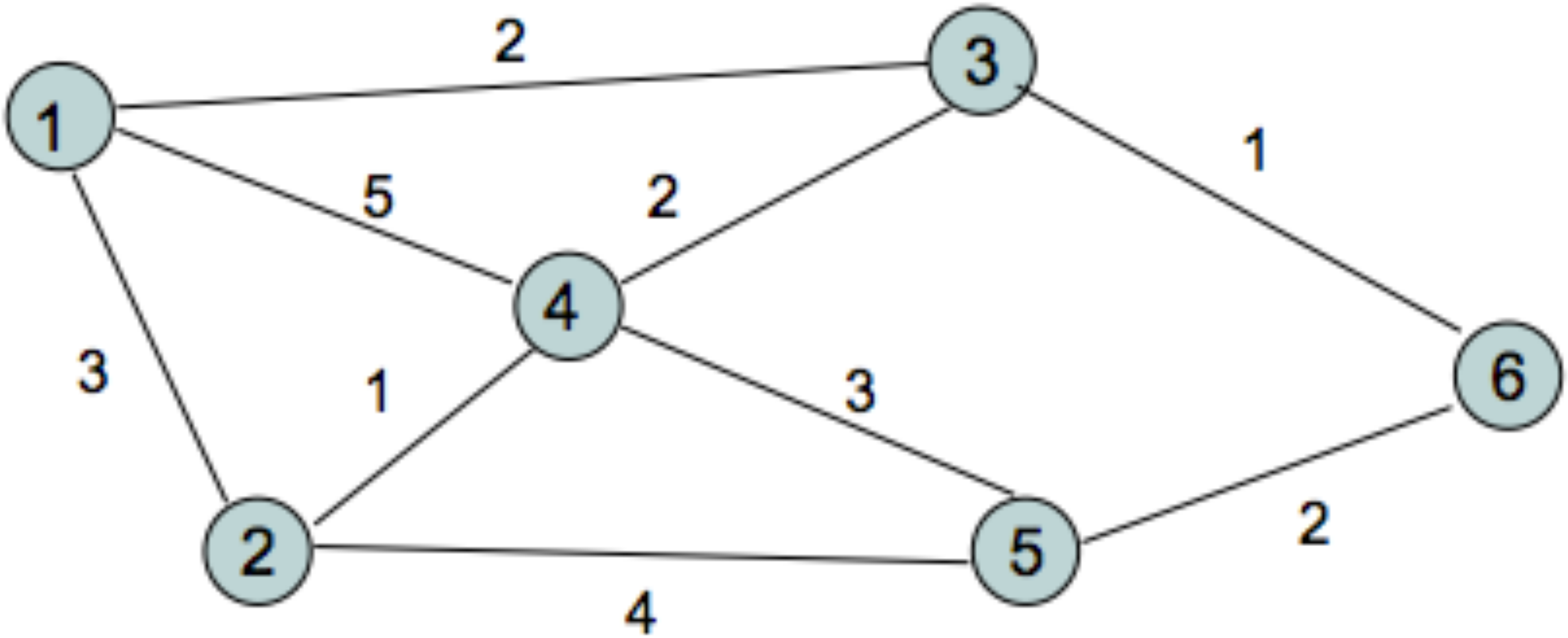
$$\left.\begin{array}{l} \textbf{if } d[v] > d[u] + w(u, v) \textbf{ then} \\ \qquad d[v] \leftarrow d[u] + w(u, v) \\ \qquad \pi[v] \leftarrow u \end{array}\right\} \textit{relaxation step}$$

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 Example 1



Bellman-Ford Example 2



Credits

- http://www.personal.kent.edu/~rmuhamma/Algorithms/MyAlgorithms/GraphAlgor/bellFordAlgor.htm

- http://www.cs.arizona.edu/classes/cs545/fall09/ShortestPath2.prn.pdf