

The background of the slide features a 3D perspective grid of nodes and edges. The nodes are represented as small, light blue spheres, and the edges are thin, light blue lines connecting them. The grid recedes into the distance, creating a sense of depth. The overall color scheme is a gradient of dark blue.

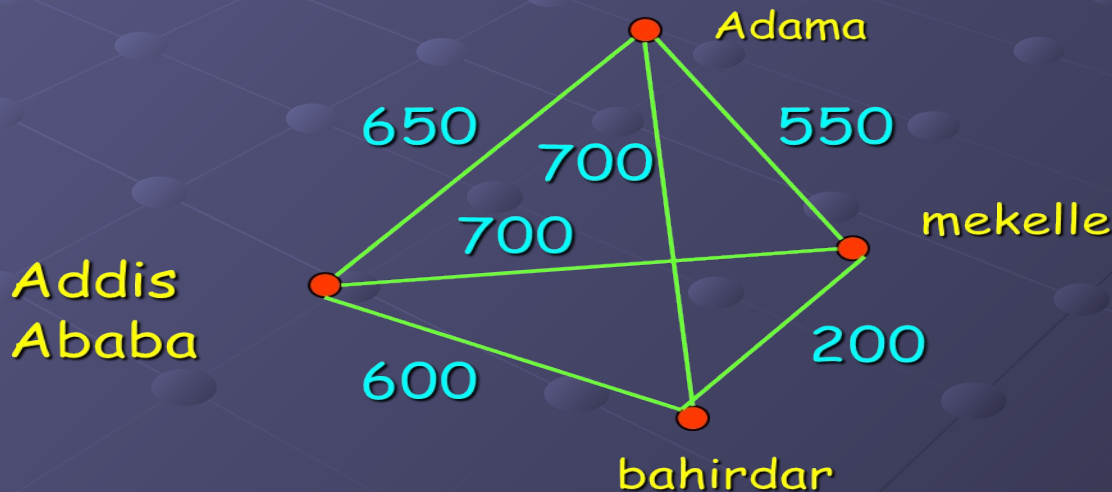
Minimum Spanning Trees

Outline

- Spanning trees and minimum spanning trees.
- The minimum spanning tree (MST) problem.
- Prim's and Kruskal's algorithm for the MST problem.

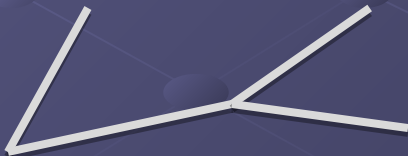
Weighted graph

- In many applications, each edge of a graph has an associated numerical value, called a weight. Usually, the edge weights are nonnegative integers. Weighted graphs may be either directed or undirected.



Spanning tree

- A subgraph T of a undirected graph $G = (V, E)$ is a spanning tree of G if it is a tree and contains every vertex of G .
- Example:



MST

- A minimum spanning tree connects all nodes in a given graph
- A MST must be a connected and undirected graph
- A MST can have weighted edges
- Multiple MSTs can exist within a given undirected graph

More about Multiple MSTs

- Multiple MSTs can be generated depending on which algorithm is used
- If you wish to have a MST start at a specific node
- However, if there are weighted edges and all weighted edges are unique, only one MST will exist

Real Life Application of a MST

A cable TV company is laying cable in a new neighborhood. If it is constrained to bury the cable only along certain paths, then there would be a graph representing which points are connected by those paths. Some of those paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights. *A minimum spanning tree* would be the network with the lowest total cost.

Borůvka's Algorithm

- The first MST Algorithm was created by Otakar Borůvka in 1926
- The algorithm was used to create efficient connections between the electricity network in the Czech Republic
- No longer used since Prim's and Kruskal's algorithms were discovered

Prim's Algorithm

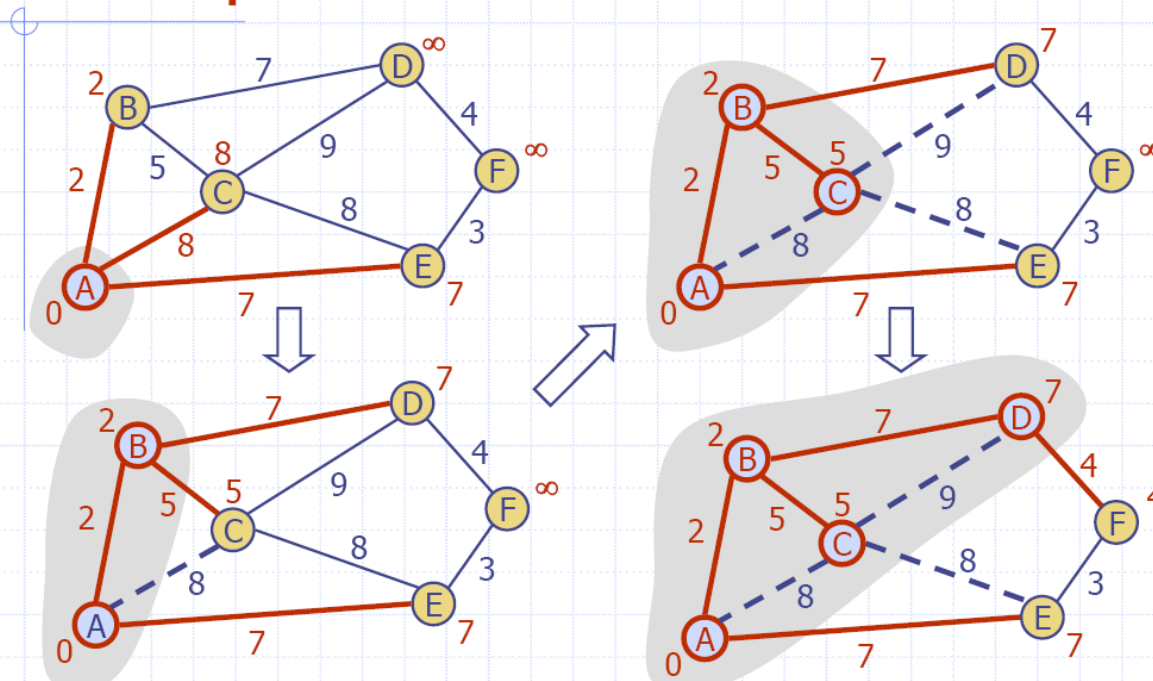
- Initially discovered in 1930 by Vojtěch Jarník, then rediscovered in 1957 by Robert C. Prim
- Similar to Dijkstra's Algorithm regarding a connected graph
- Starts off by picking any node within the graph and growing from there

Prim's Algorithm Cont.

- Label the starting node, A, with a 0 and all others with infinite
- Starting from A, update all the connected nodes' labels to A with their weighted edges if it less than the labeled value
- Find the next smallest label and update the corresponding connecting nodes
- Repeat until all the nodes have been visited

Prim's Algorithm Example

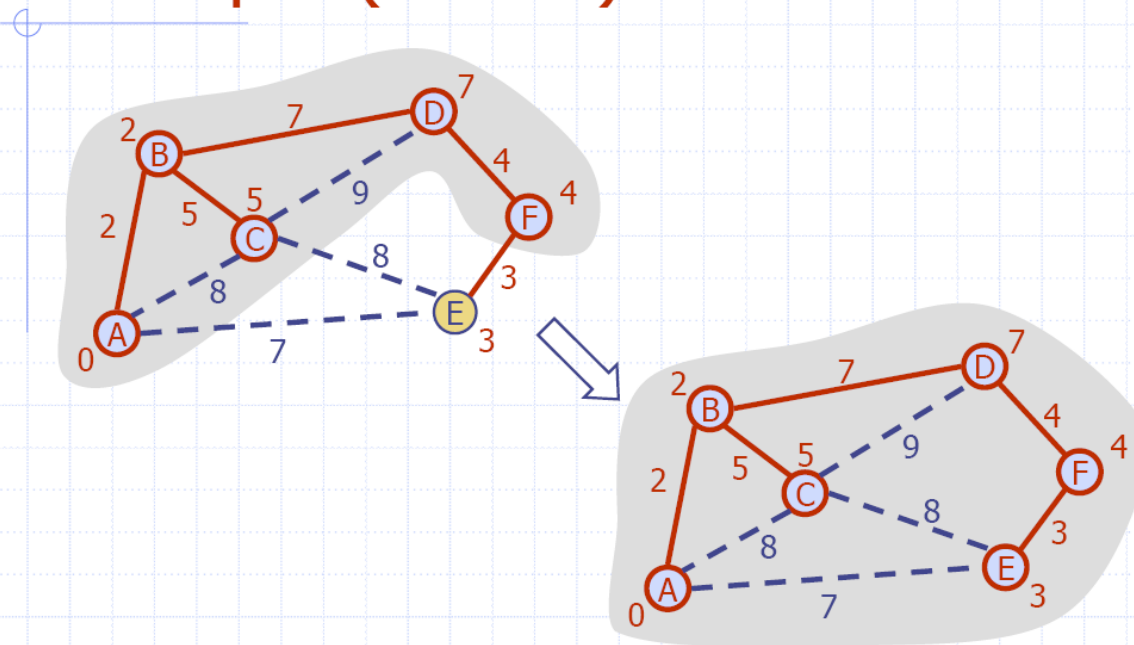
Example



Minimum Spanning Trees

Prim's Algorithm Example

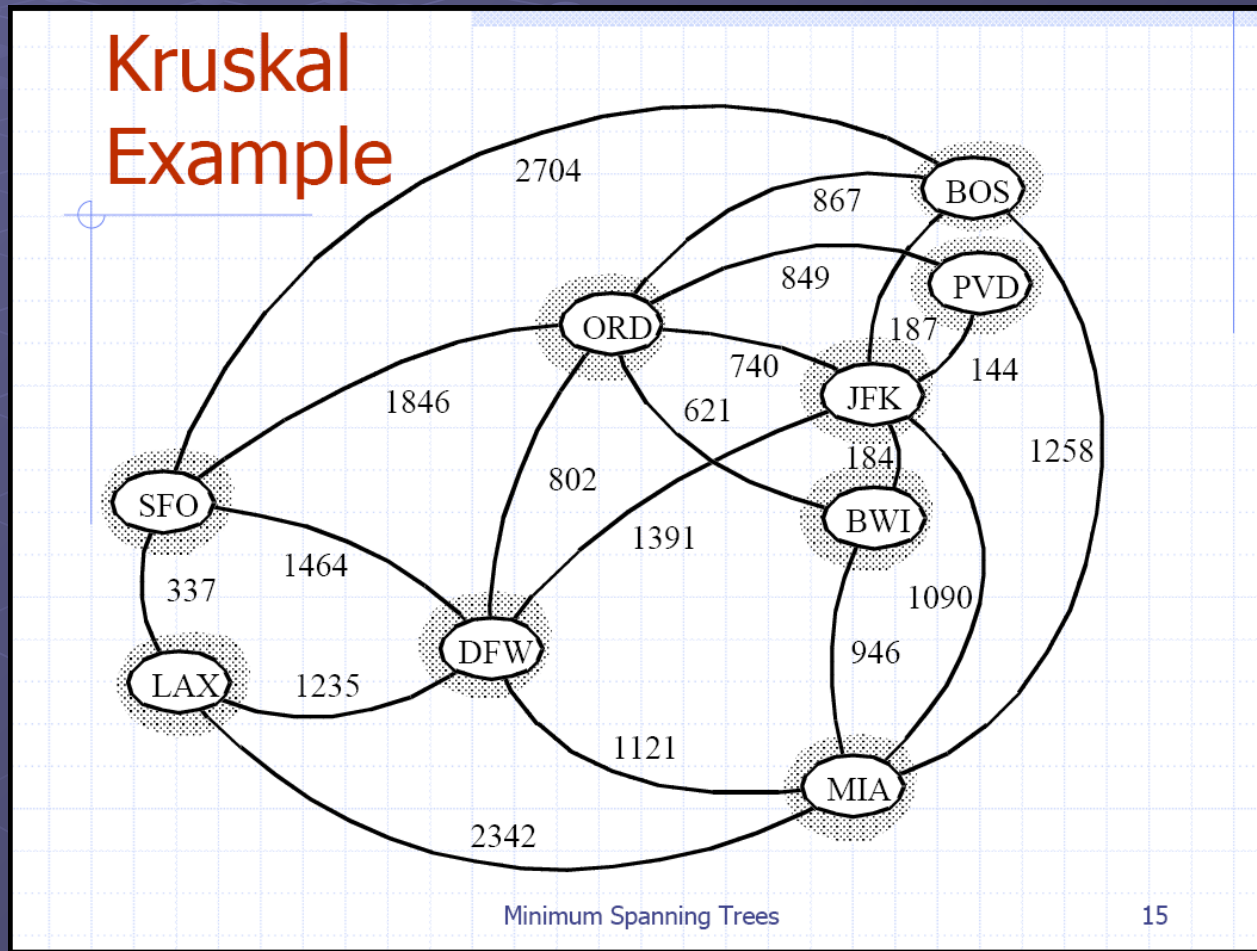
Example (contd.)



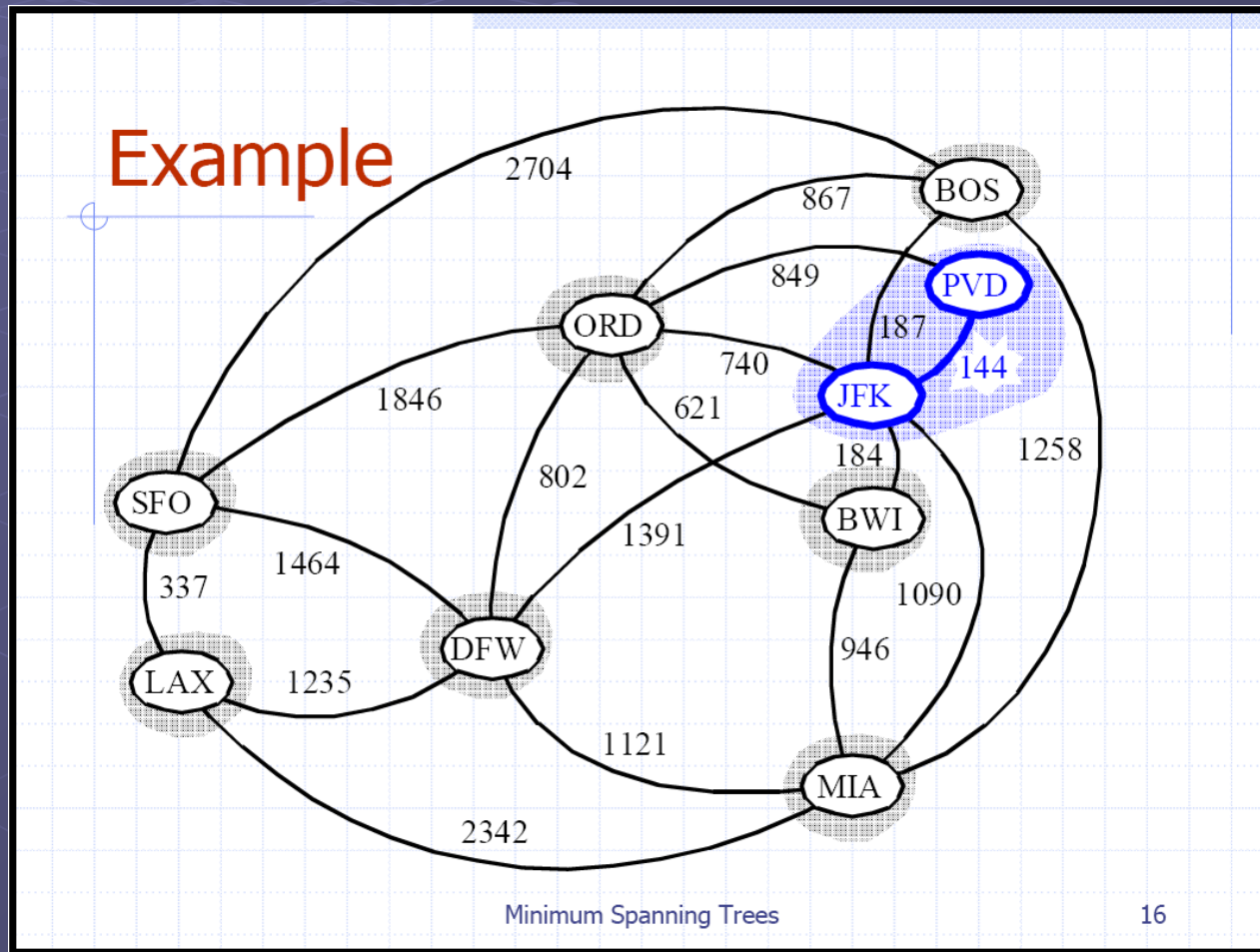
Kruskal's Algorithm

- Created in 1957 by Joseph Kruskal
- Finds the MST by taking the smallest weight in the graph and connecting the two nodes and repeating until all nodes are connected to just one tree
- This is done by creating a priority queue using the weights as keys
- Each node starts off as it's own tree
- While the queue is not empty, if the edge retrieved connects two trees, connect them, if not, discard it
- Once the queue is empty, you are left with the minimum spanning tree

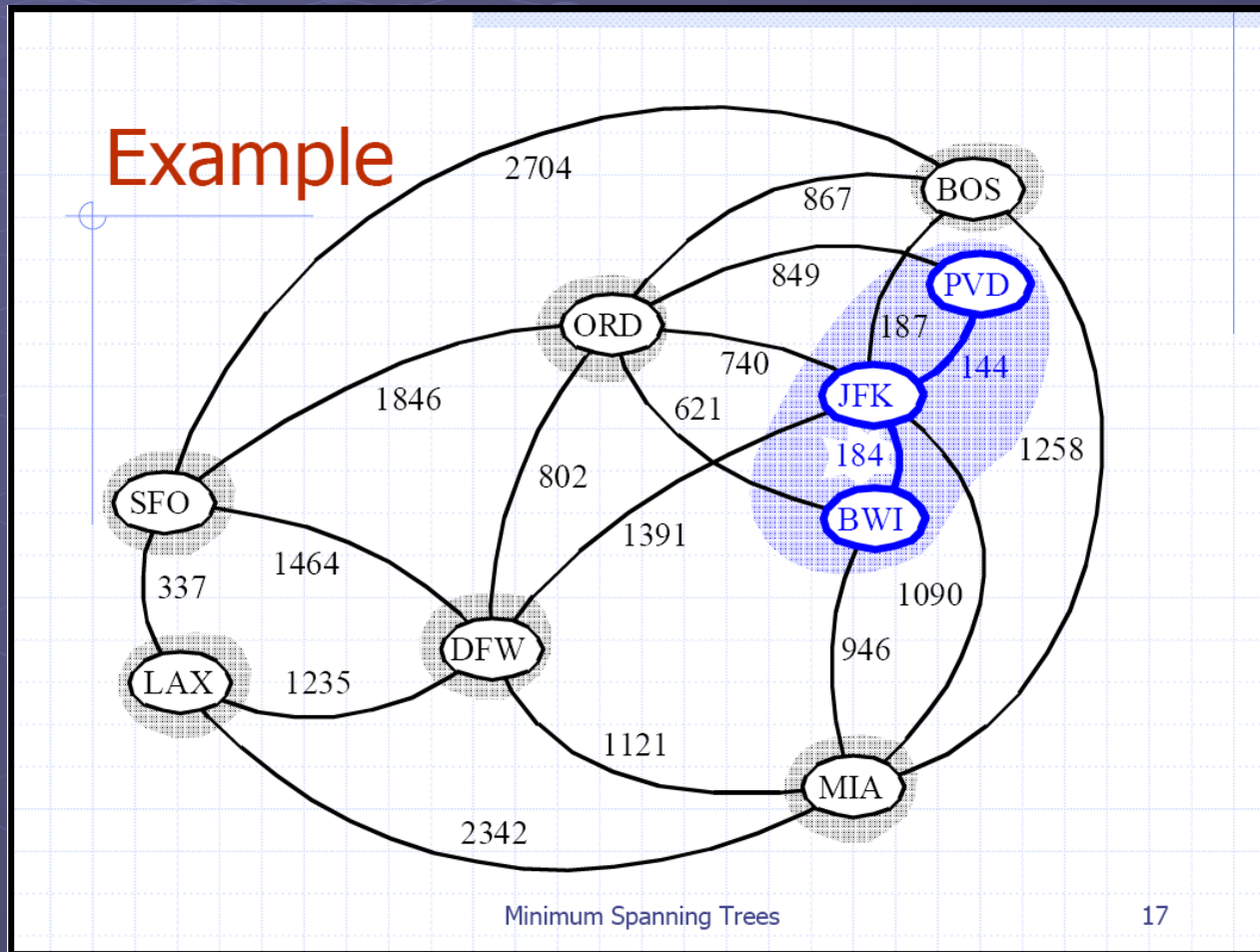
Kruskal's Algorithm Example



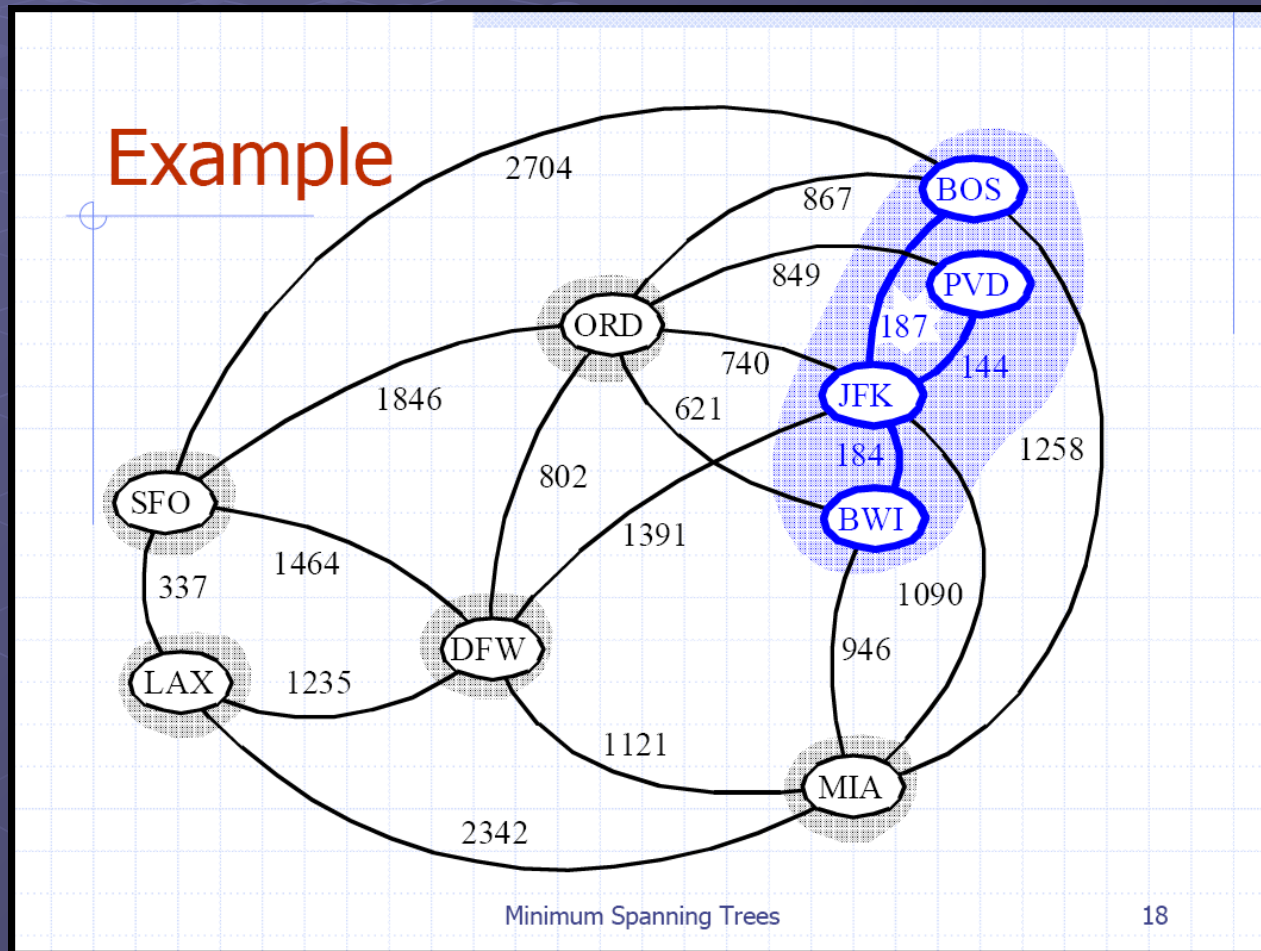
Kruskal's Algorithm Example



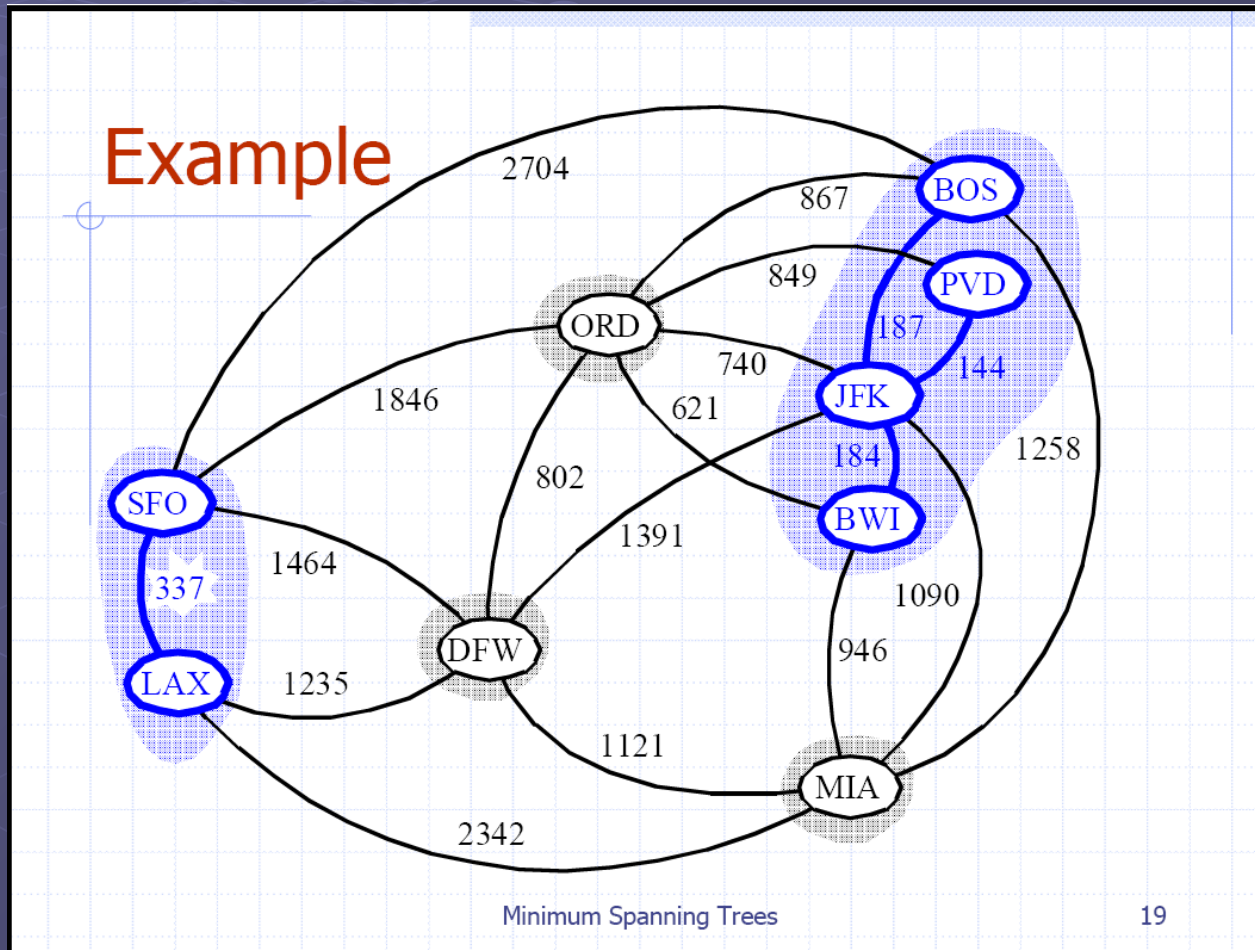
Kruskal's Algorithm Example



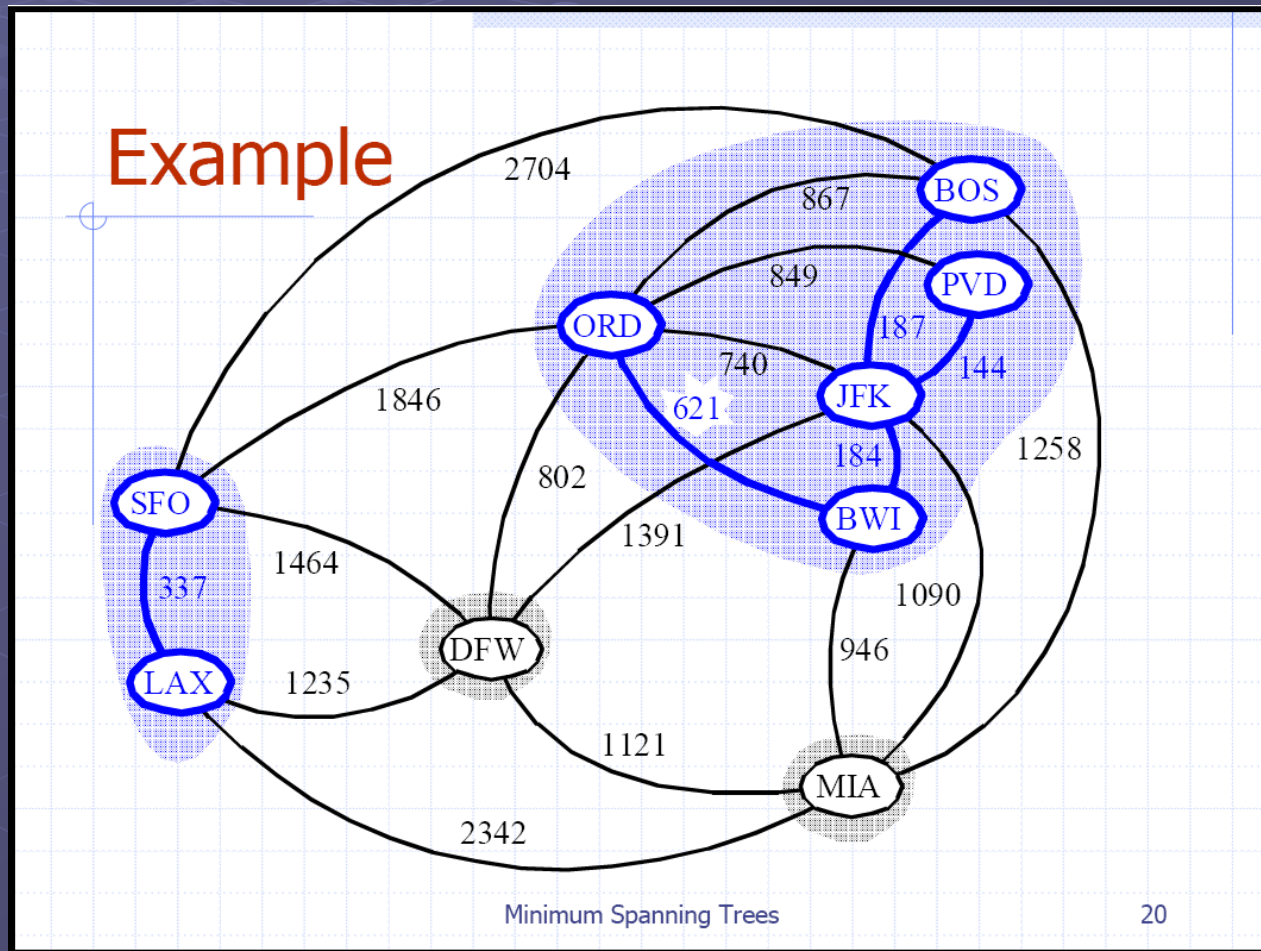
Kruskal's Algorithm Example



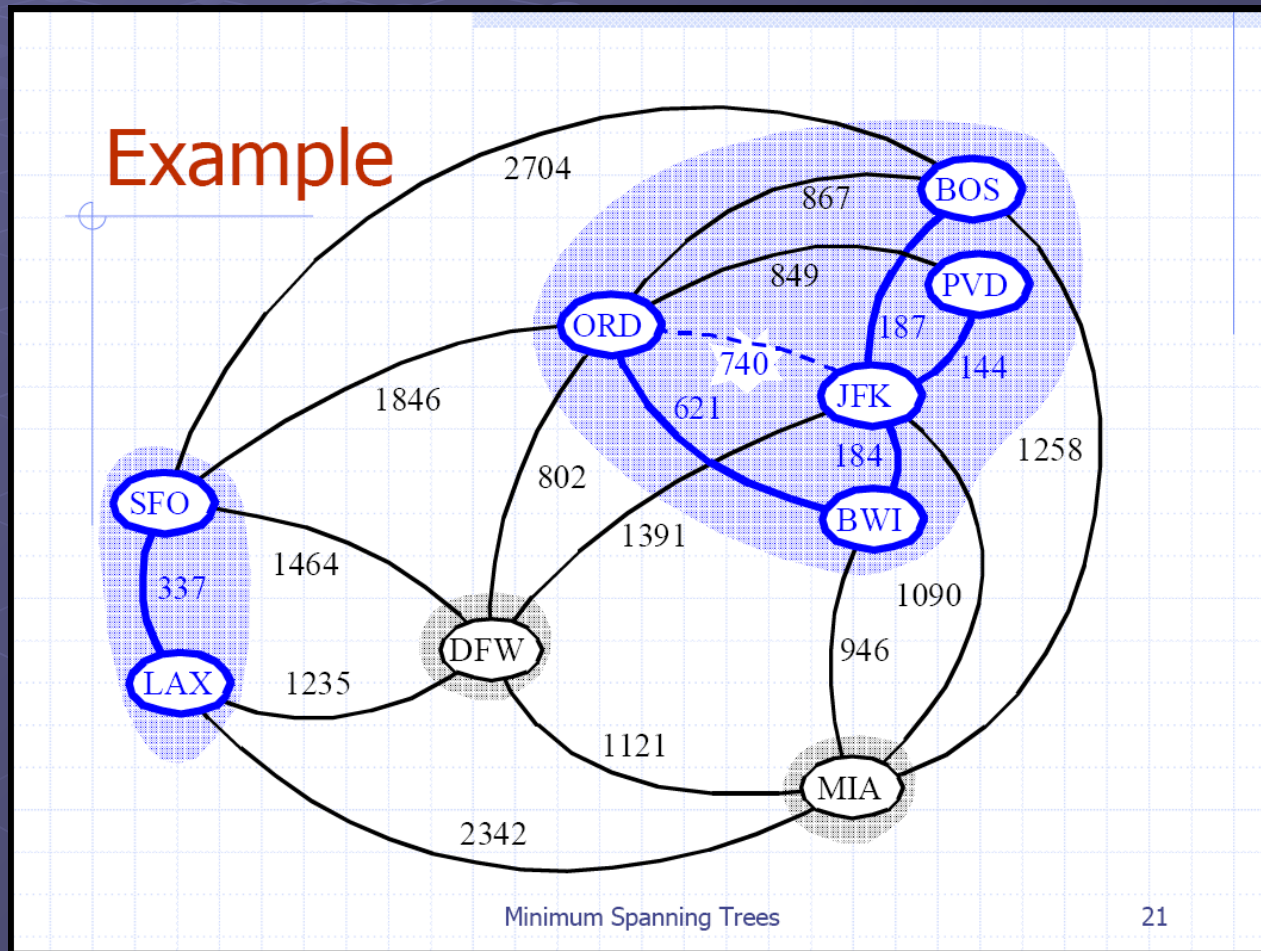
Kruskal's Algorithm Example



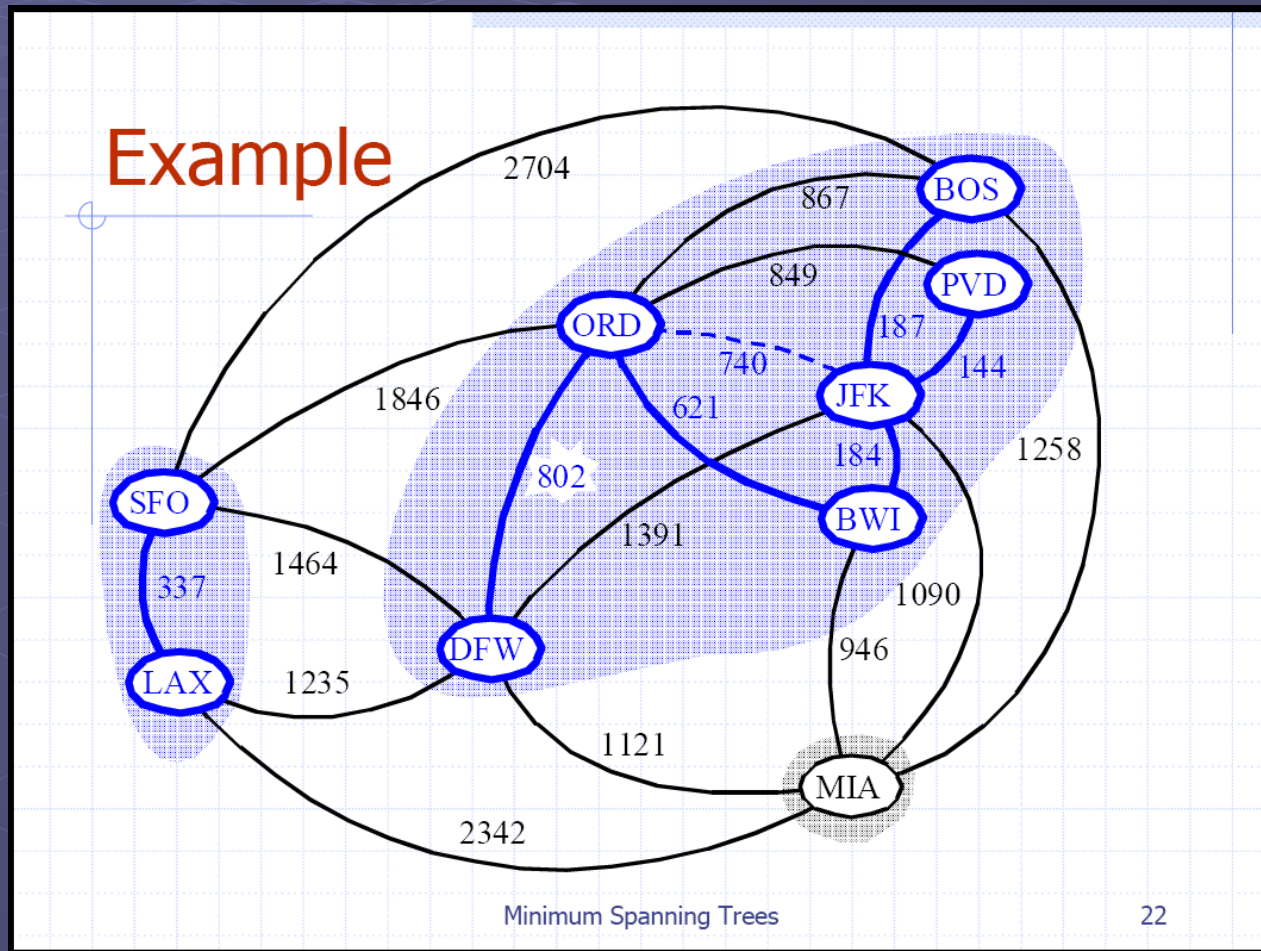
Kruskal's Algorithm Example



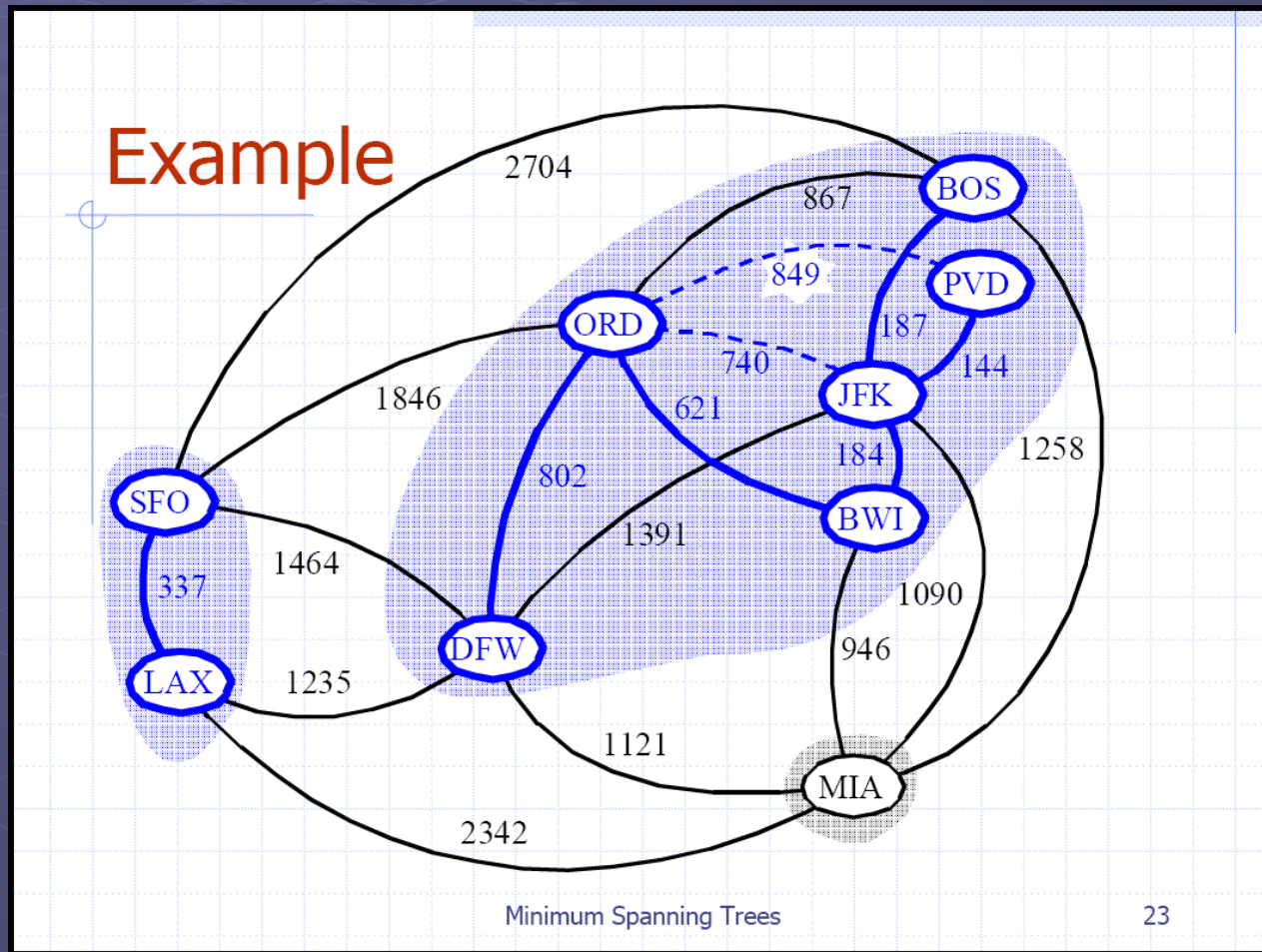
Kruskal's Algorithm Example



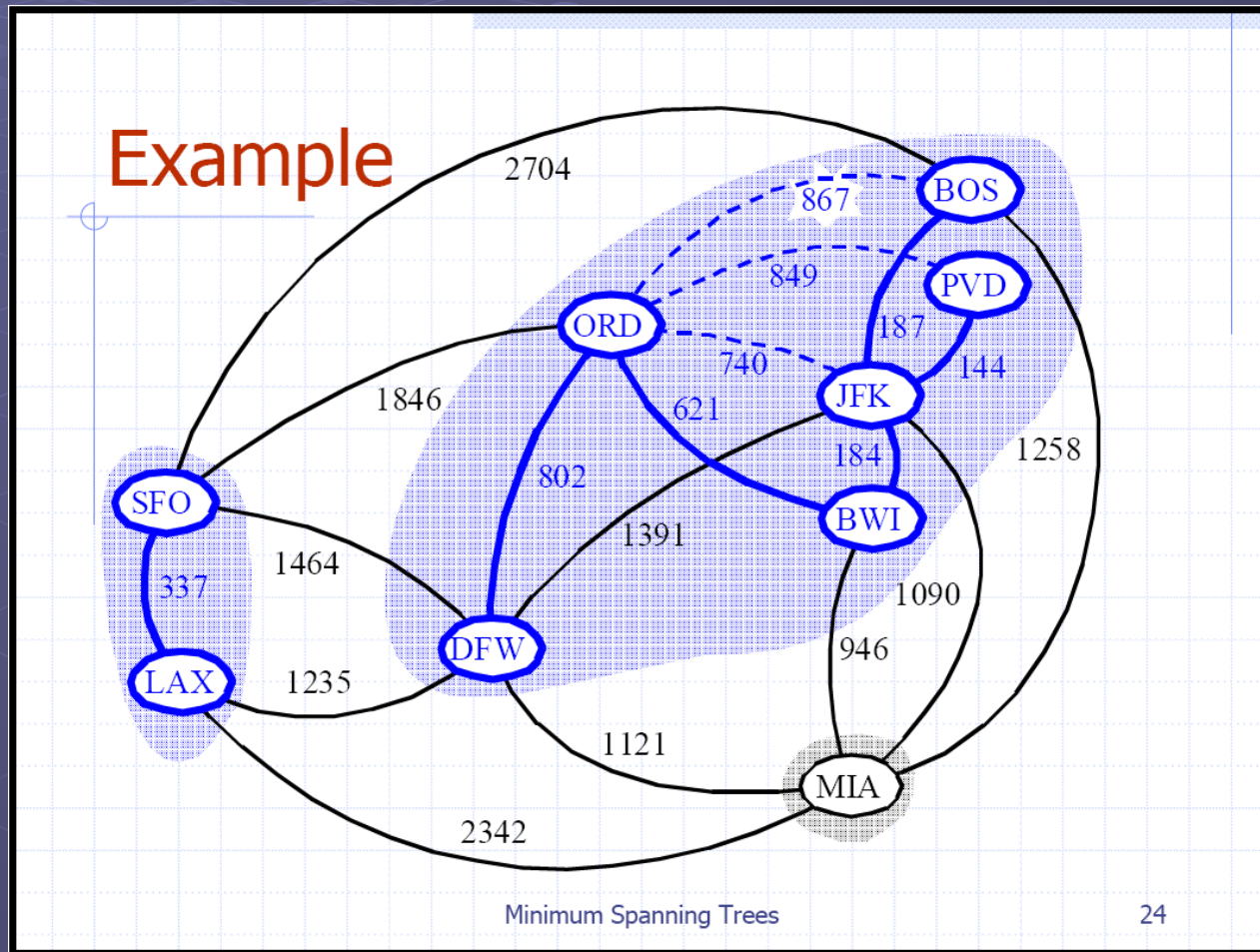
Kruskal's Algorithm Example



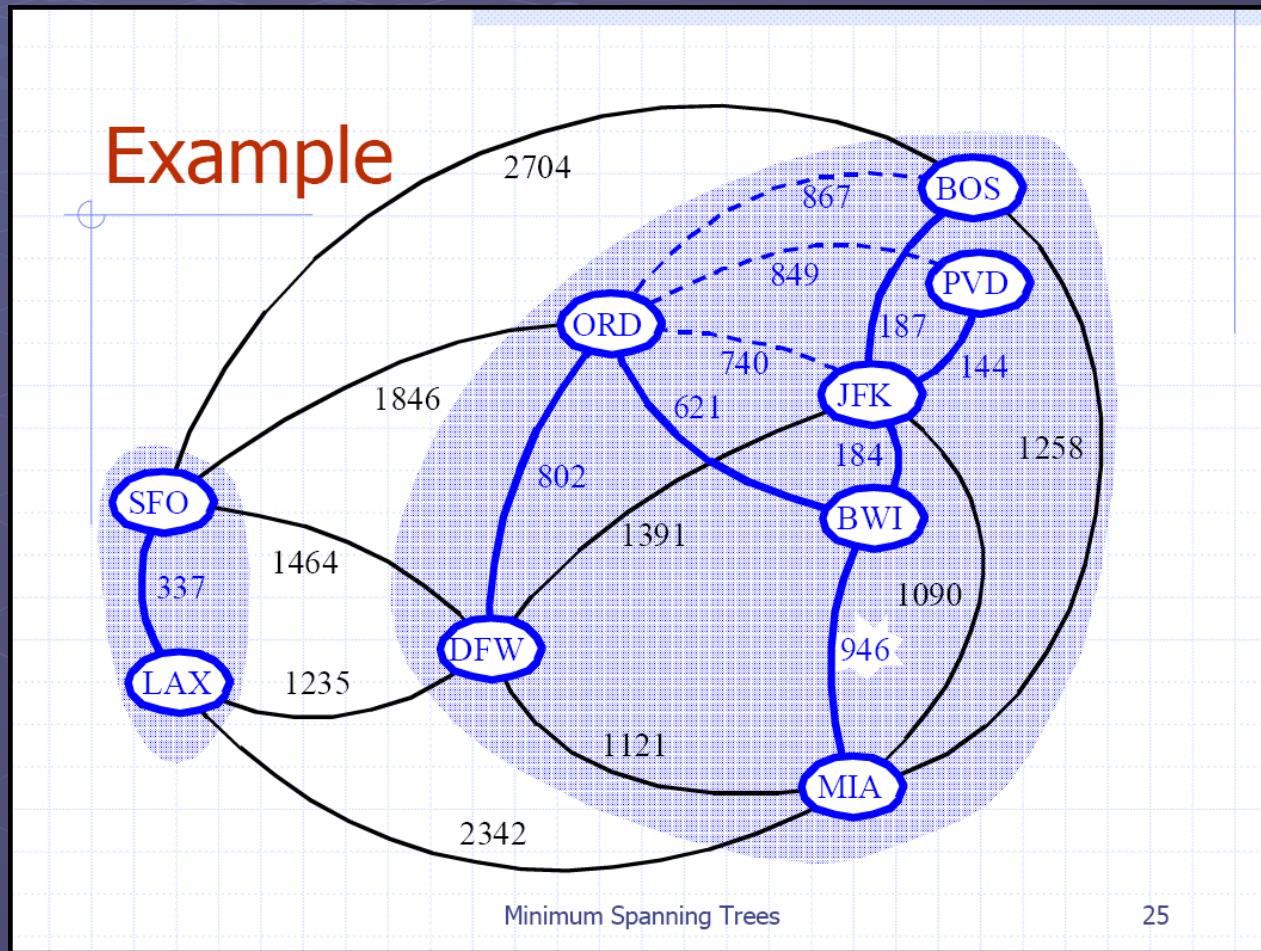
Kruskal's Algorithm Example



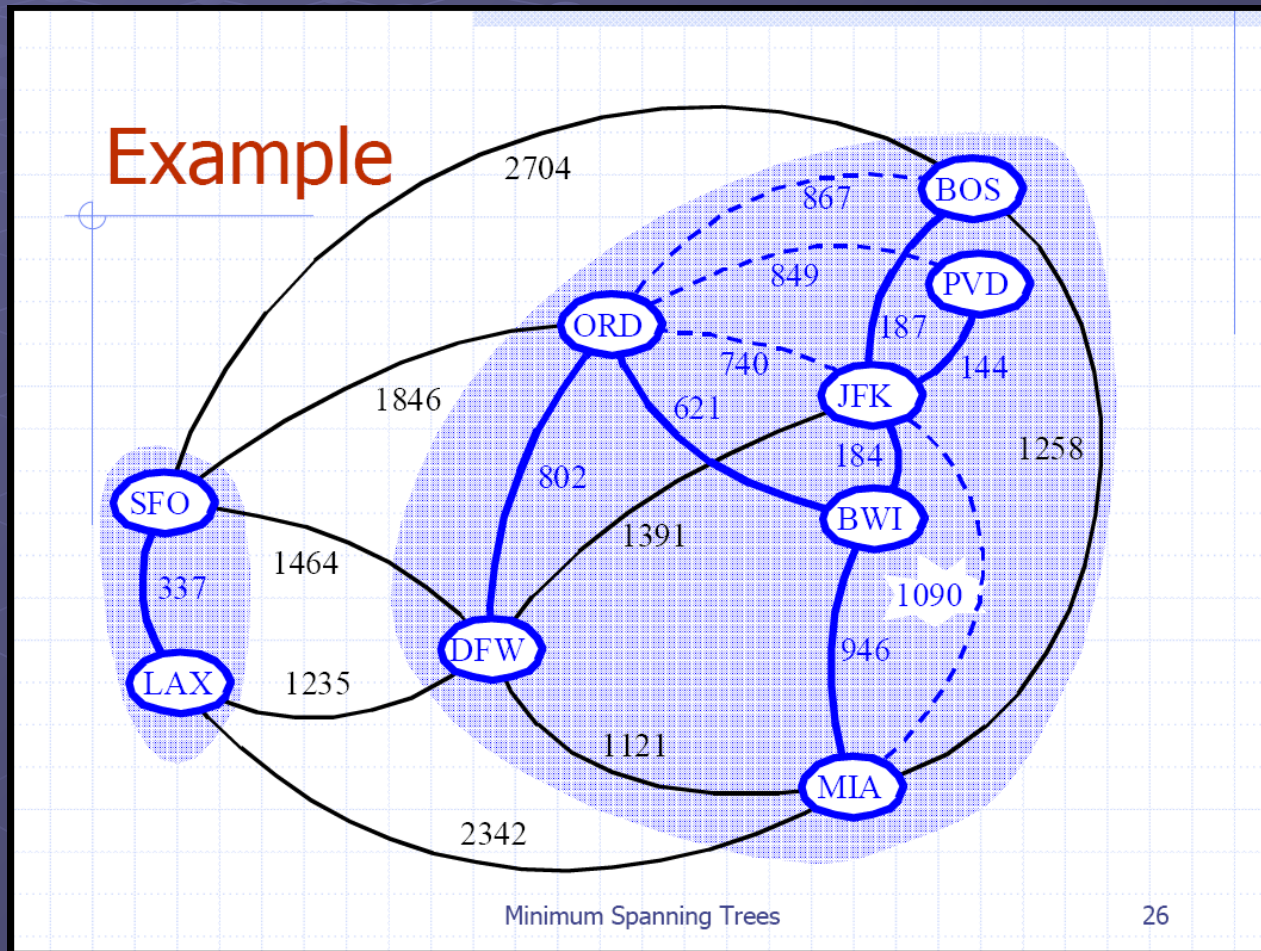
Kruskal's Algorithm Example



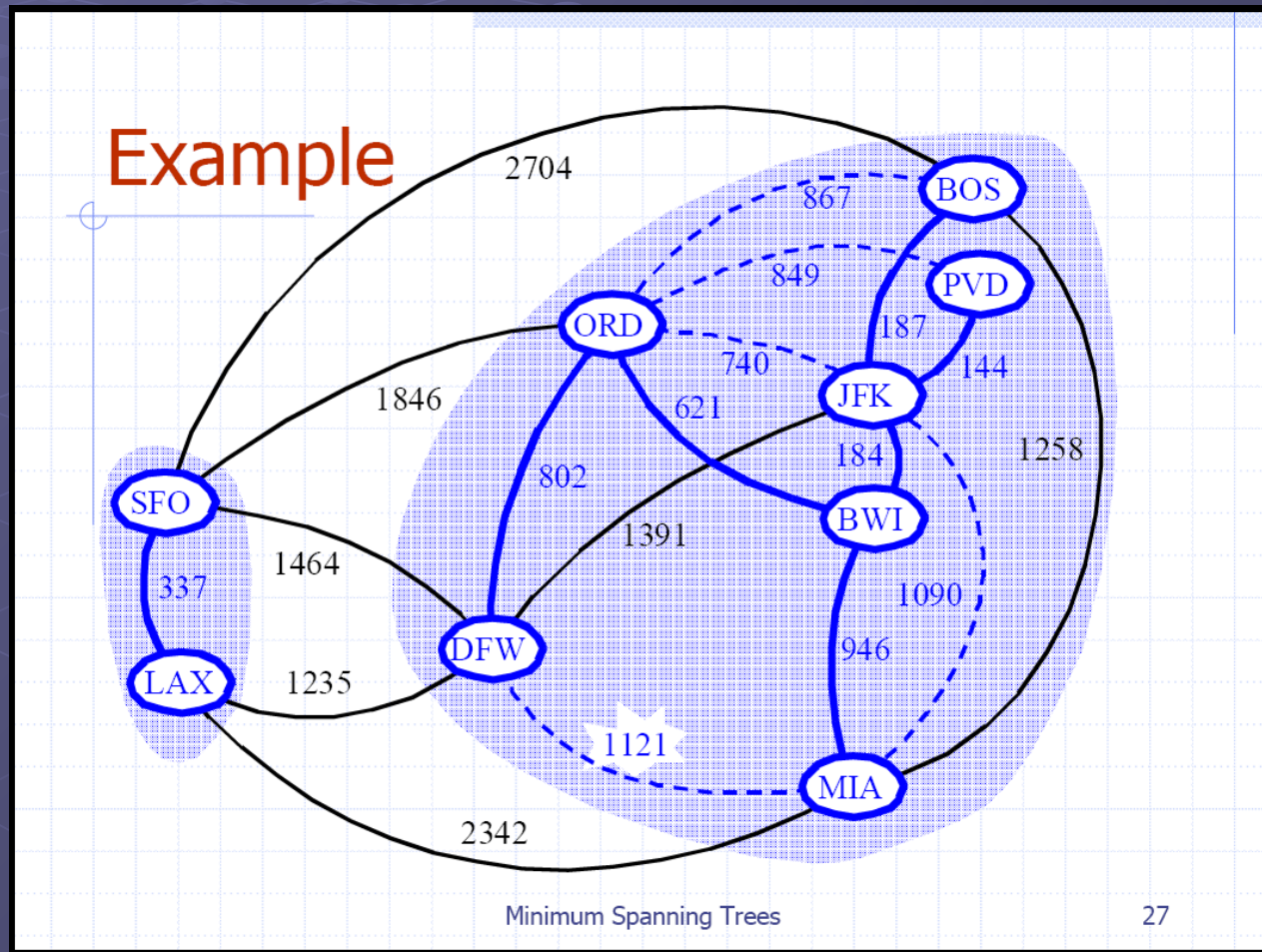
Kruskal's Algorithm Example



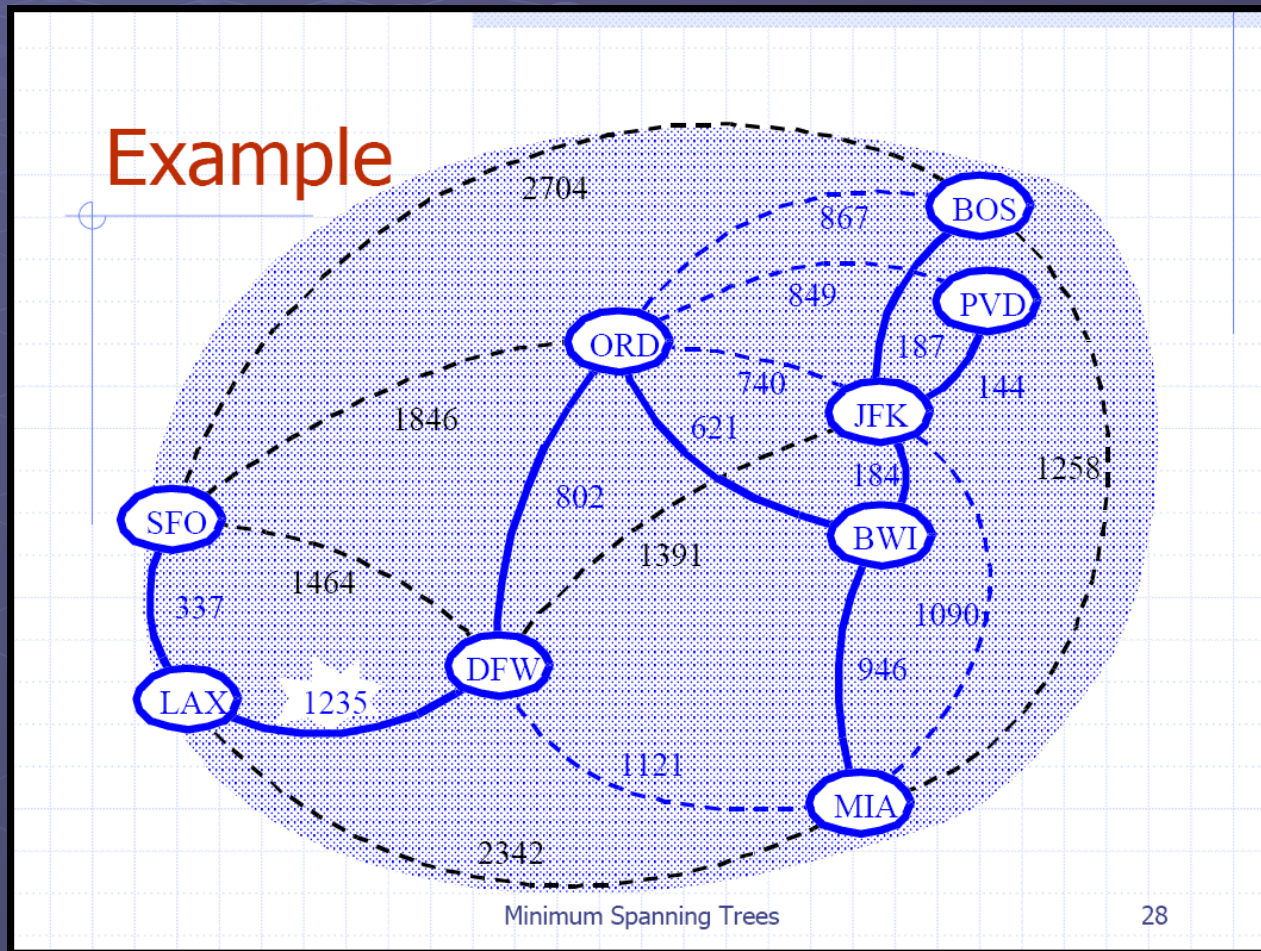
Kruskal's Algorithm Example



Kruskal's Algorithm Example



Kruskal's Algorithm Example



Dijkstra's algorithm

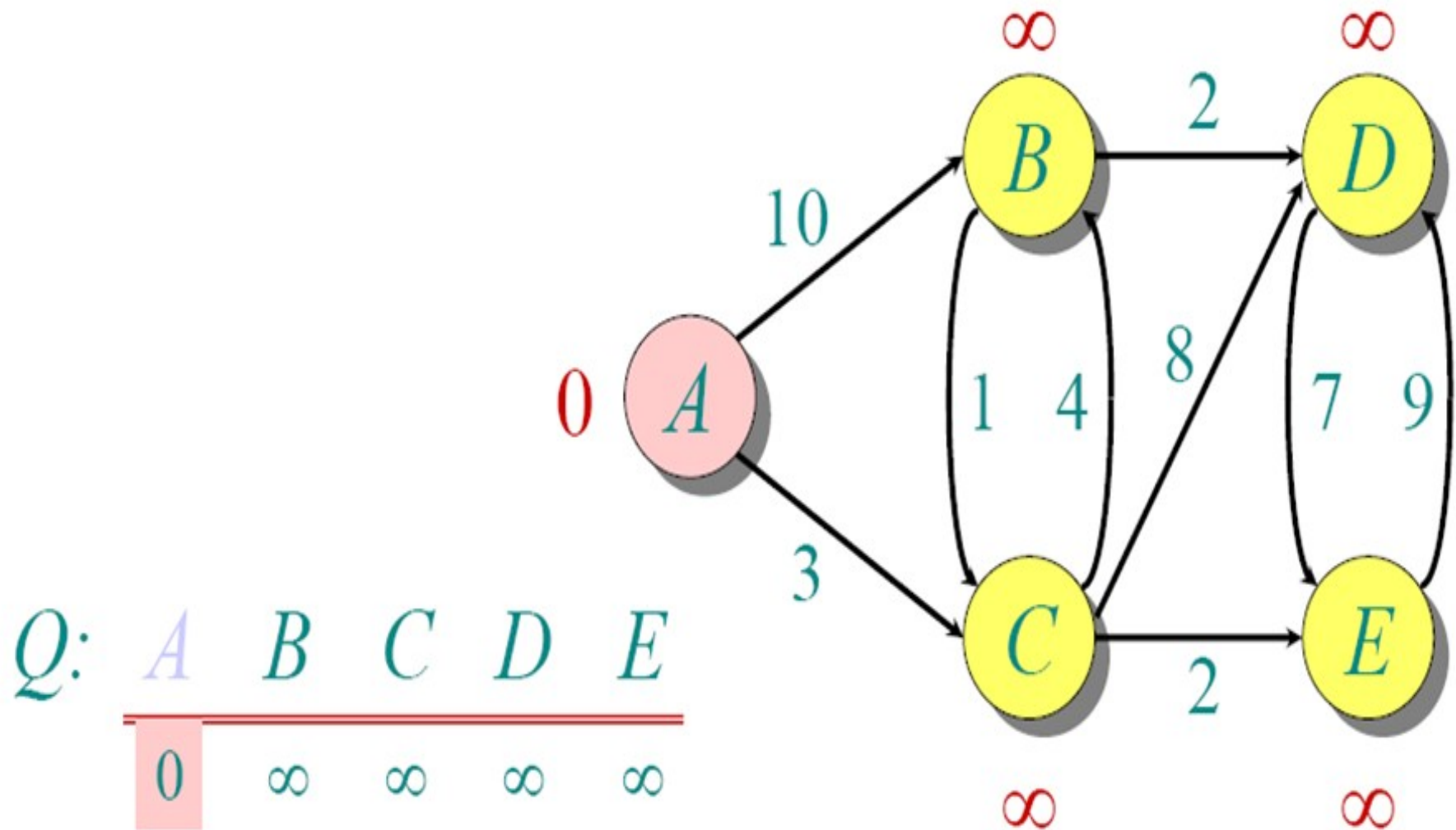
Dijkstra's algorithm - is a solution to the single-source shortest path problem in graph theory.

Works on both directed and undirected graphs. However, all edges must have nonnegative weights.

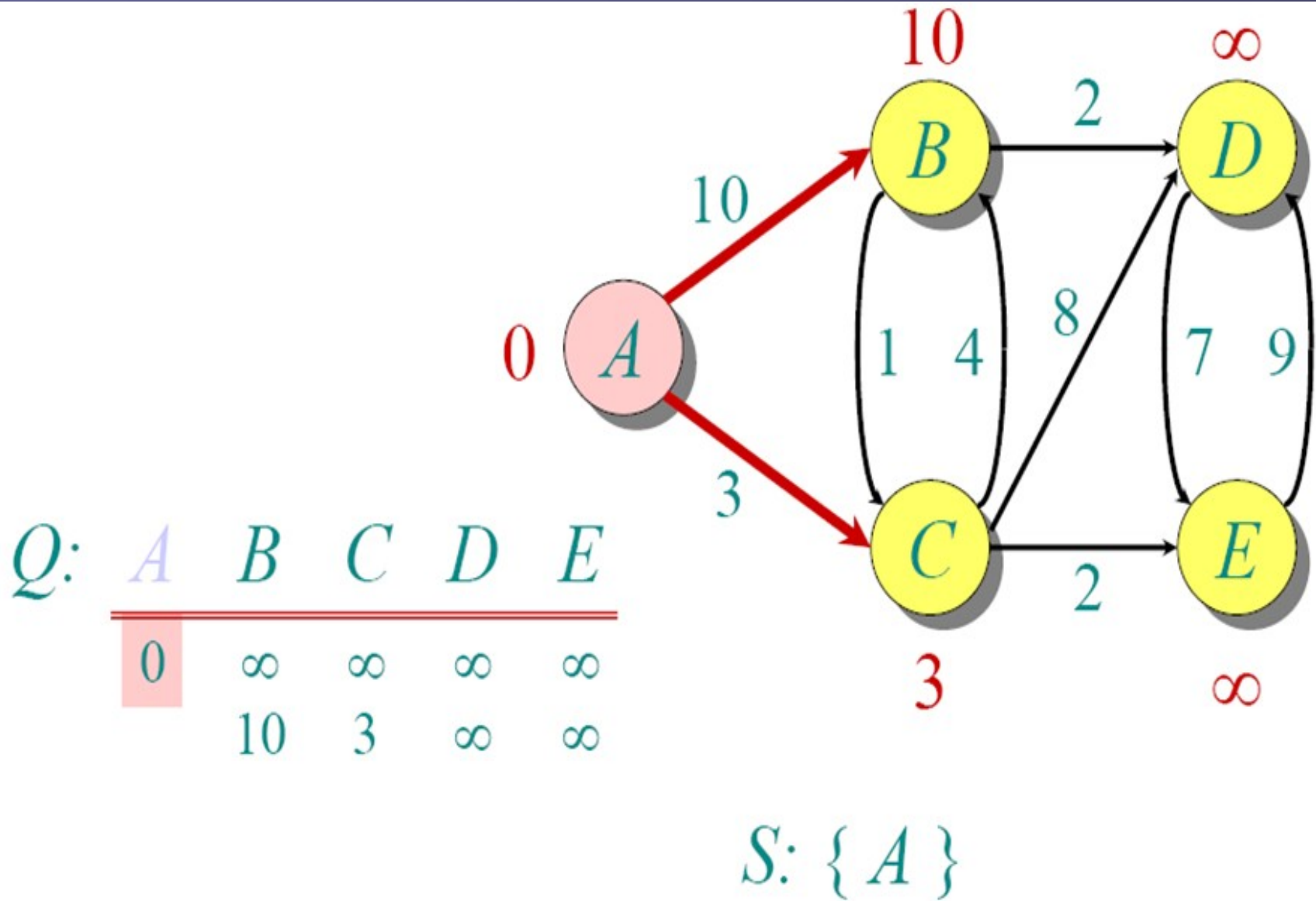
Input: Weighted graph $G=\{E,V\}$ and source vertex $v \in V$, such that all edge weights are nonnegative

Output: Lengths of shortest paths (or the shortest paths themselves) from a given source vertex $v \in V$ to all other vertices

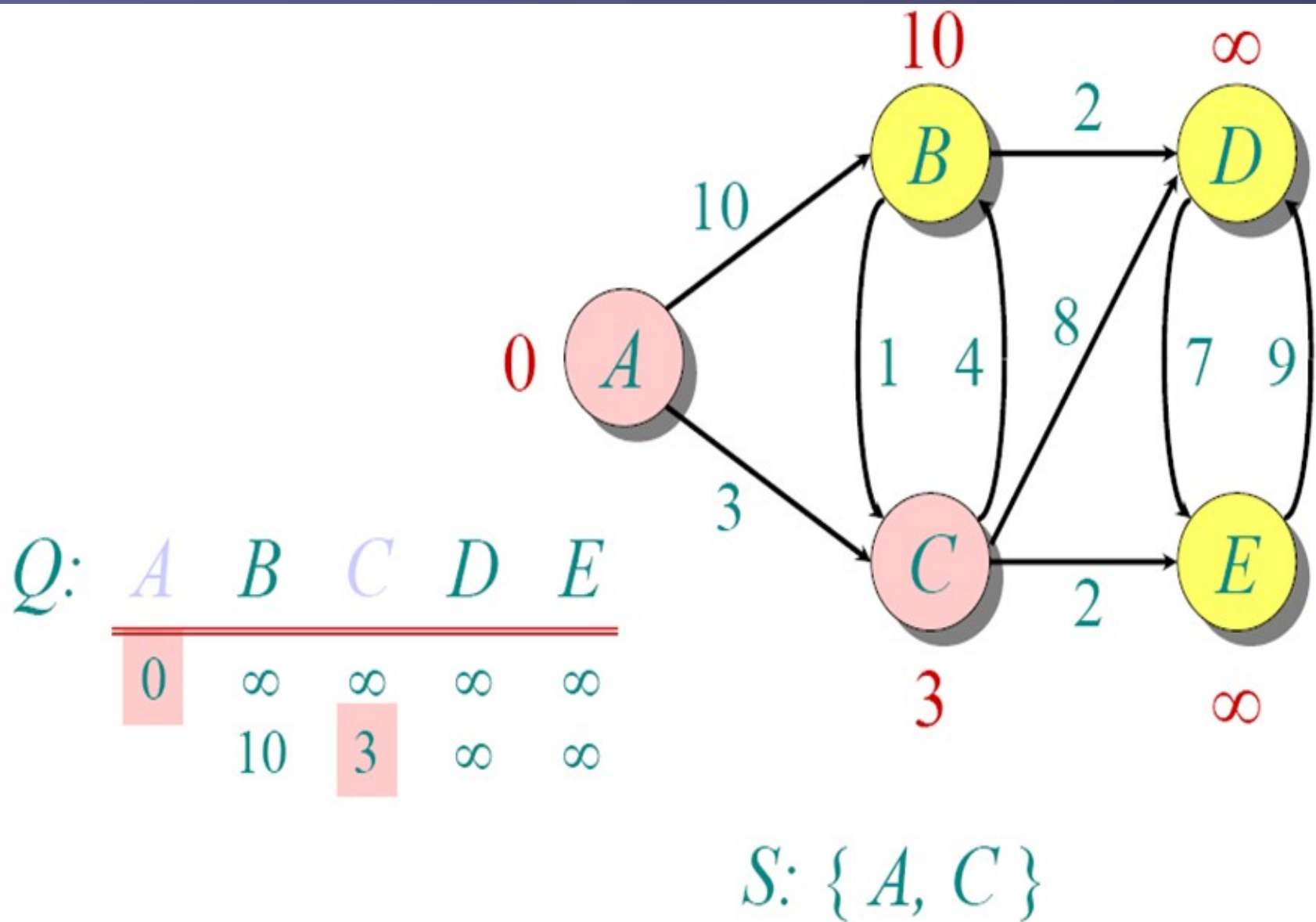
Example :-



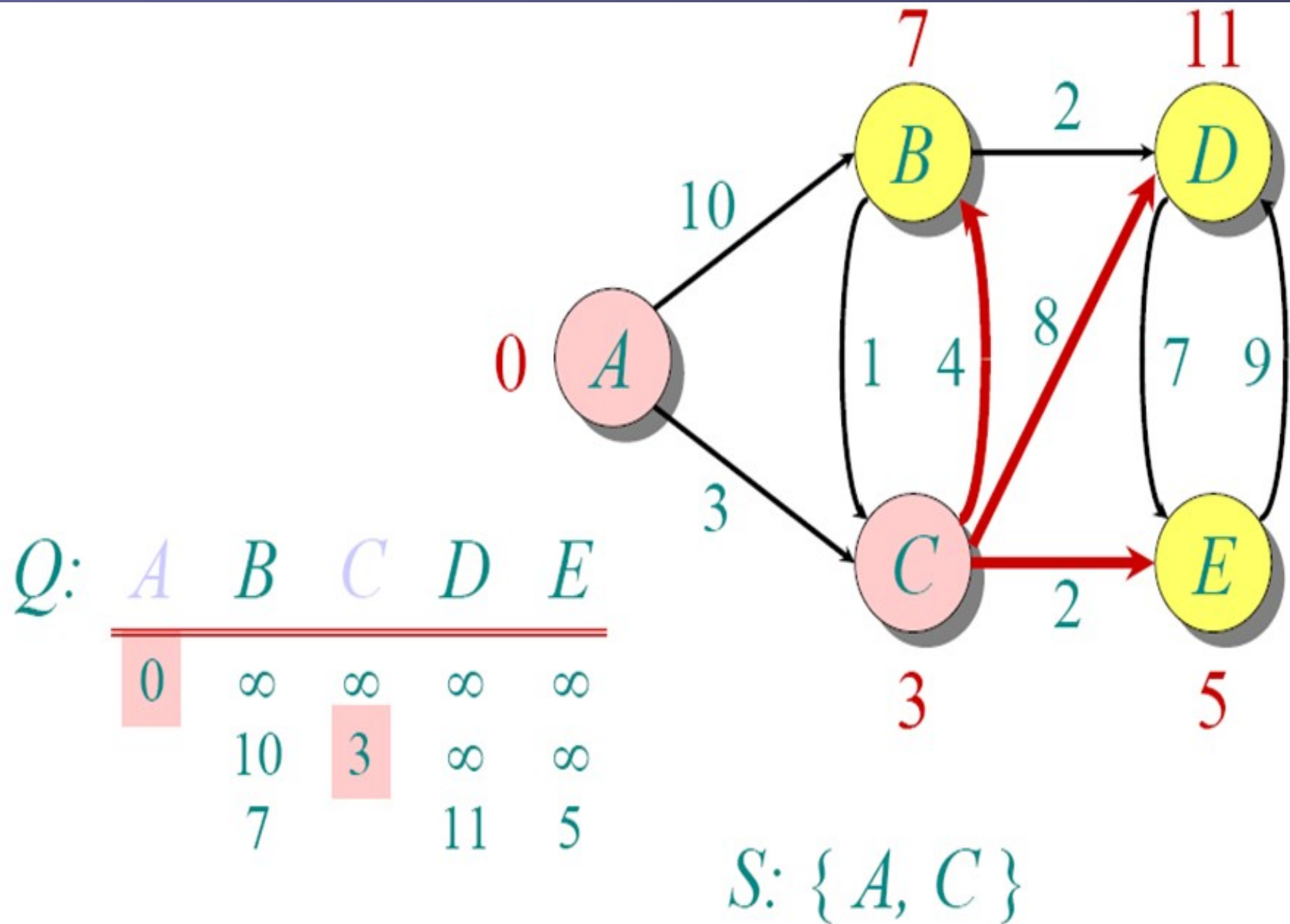
Cont.



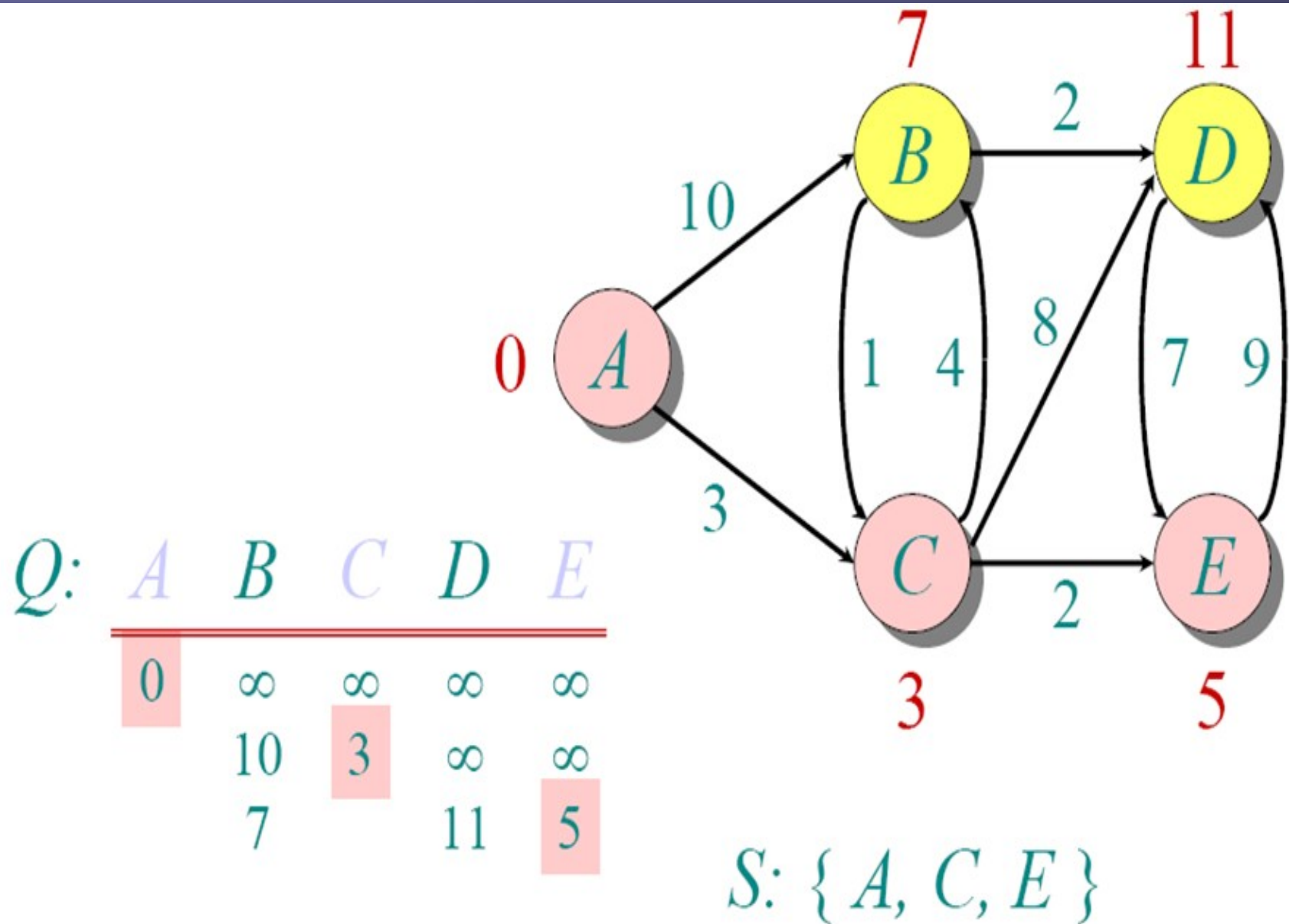
Cont.



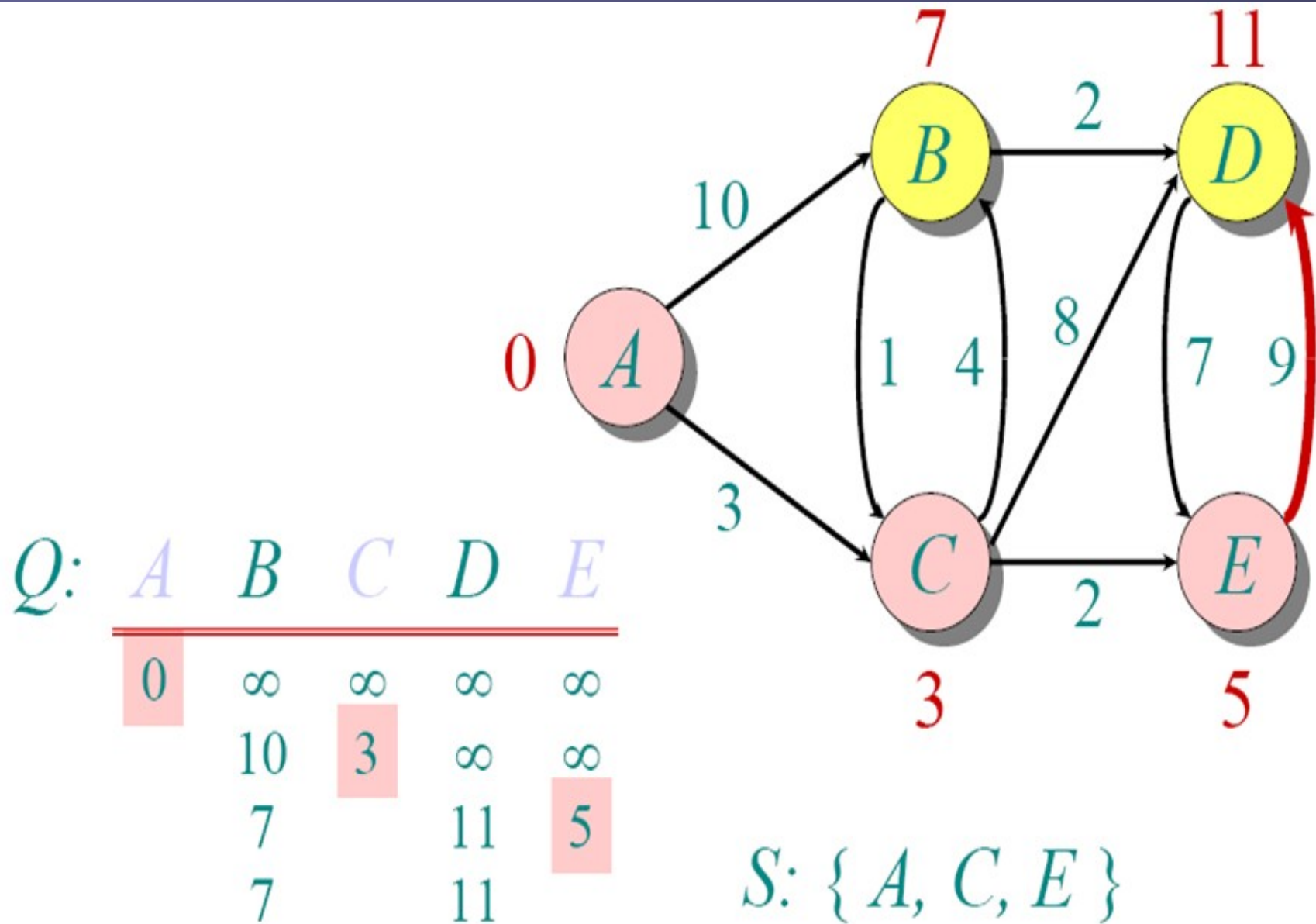
Cont.



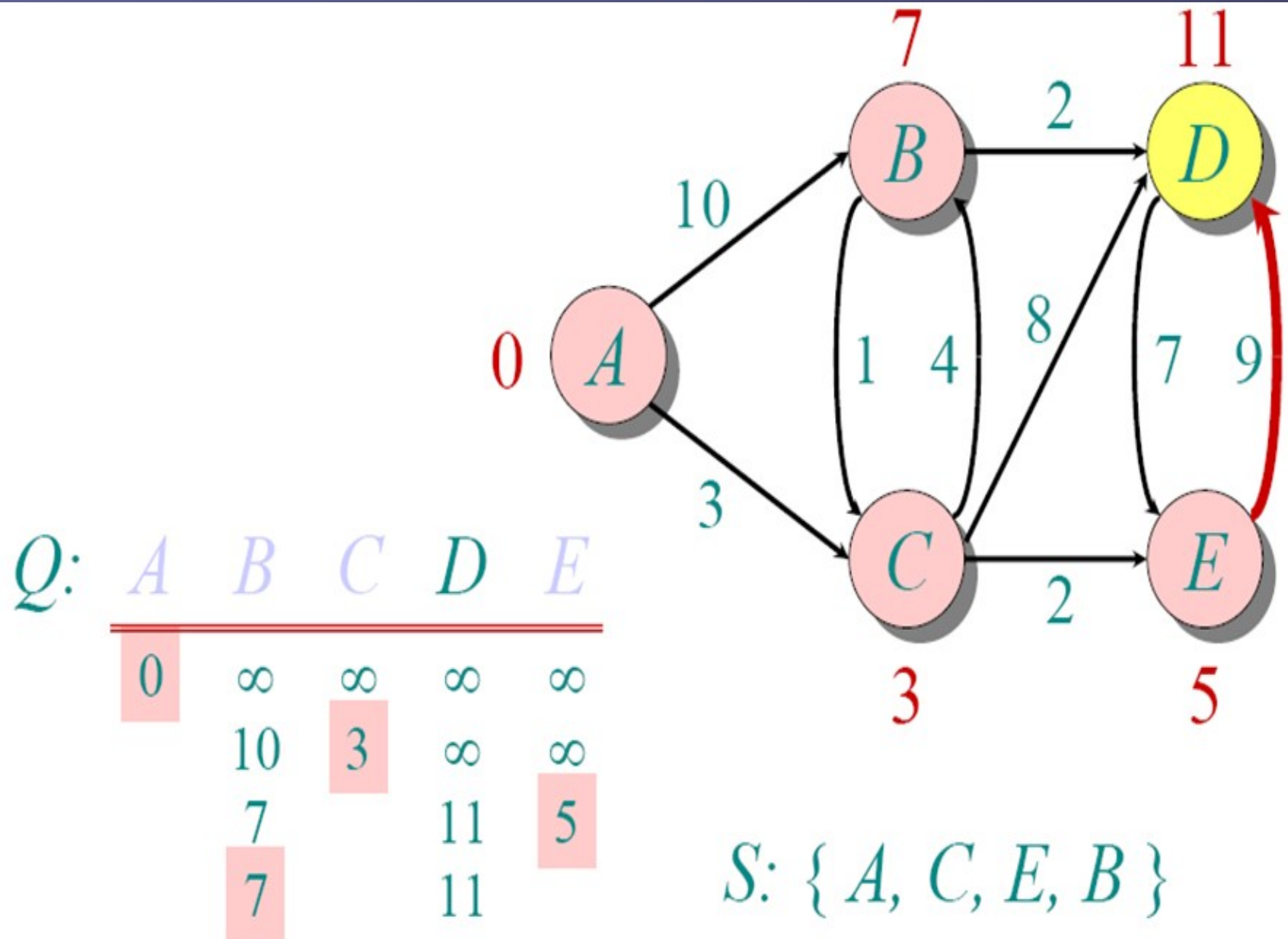
Cont.



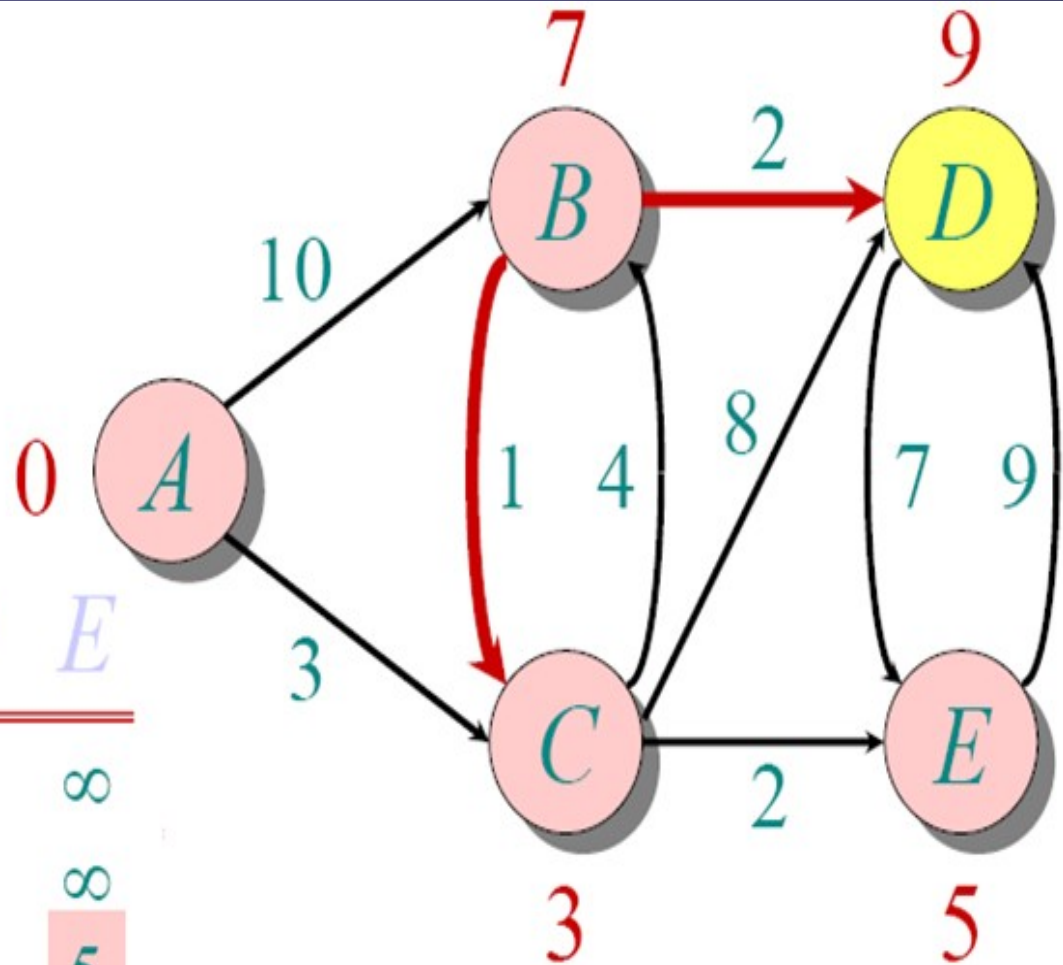
Cont.



Cont.



Cont.



$Q:$

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

$S: \{ A, C, E, B \}$

Cont.

