Programming and Algorithms

COMP1038.PGA

Week 9 – Lecture 3: Trees and Graphs

Dr. Pushpendu Kar

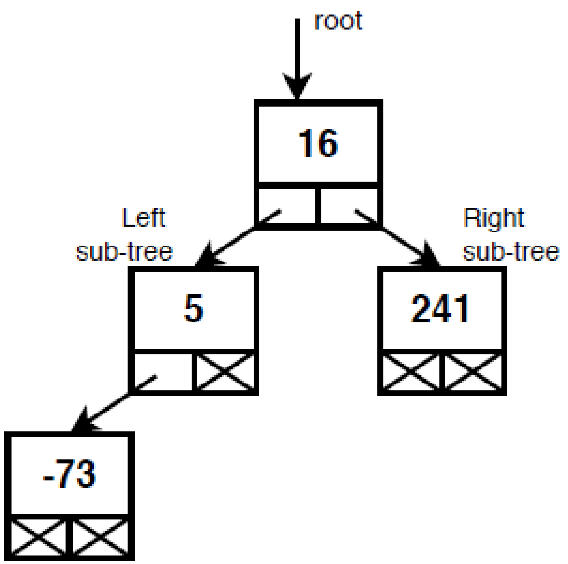
Overview

- Tree
- Graph
  - Dijkstra’s Algorithm

Tree

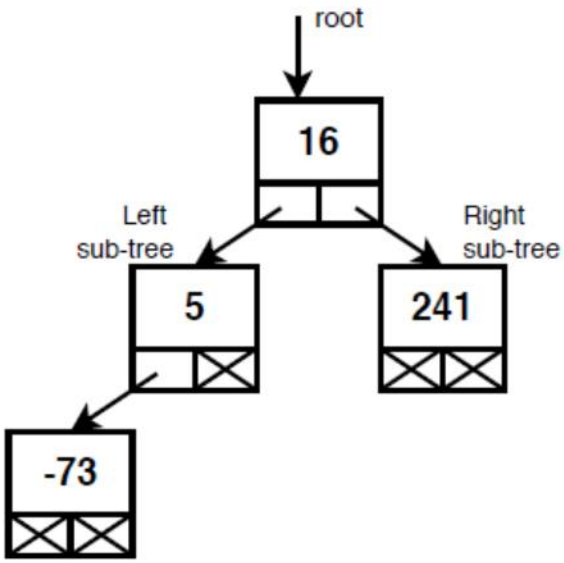
Tree: introduction

- Trees are hierarchical data structures containing nodes which store a value and references to 2 or more subtrees.
- The "start" of the tree is the root node.
- Nodes with no subtrees are called leaf nodes.
- A binary tree is a tree where each node has exactly two possible children.



Tree: binary search tree

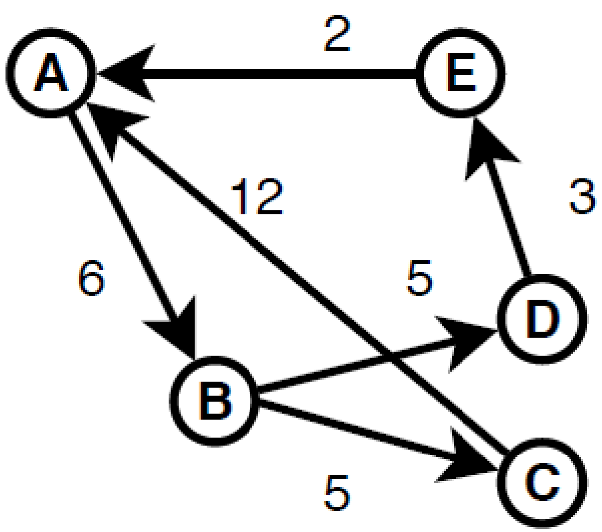
- BSTs are binary trees in which the values are stored in the tree in some specified order.
- Eg, for an integer BST,
  - every value in the left sub-tree < value in the node
  - every value in the right sub-tree >= the value in the node.
- Searching for values in a BST can be extremely quick because each comparison discards half the remaining values (on average).
- Inserting/removing nodes is more complex as it may require moving existing nodes.



Graph

Graph: introduction

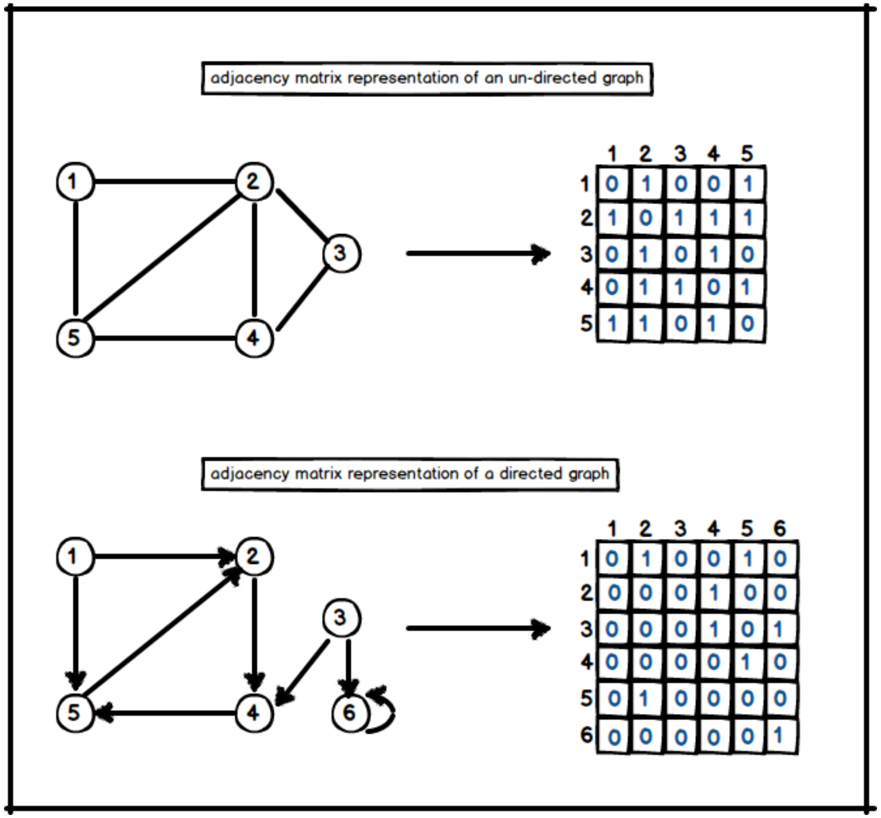
- A graph represents a set of vertices (or nodes) and a set of edges which are connections between vertices.
- Edges can be directed or undirected.
- Edges can be weighted or unweighted.
- Graphs can be connected (for any 2 vertices, there is a path between them) or unconnected.
- …and many other possible properties
  - Graph theory is a major branch of discrete maths.



Graph: adjacency matrix

- Vertices are assigned to integer IDs.
- Store graph as a 2D array of integers.
- A_{ij} contains a value if there is an edge from i to j.
  - Store edge present? Number of edges? Weight of edge?
- Very fast look-up for edge between two vertices.
- Low memory usage but wastes lots of space for sparse graphs.
- Requires whole array to be changed when adding/removing vertices.

Graph: adjacency matrix cont…



Graph: applications

- Modeling relationships between users in social media networks (vertex=user, edge=friends).
- Navigation through cities vertex=road junction, edge=road between junction, weight=distance).
- Modelling state machines (vertex=state, edge=valid transition).

Introduction to Dijkstra's Algorithm

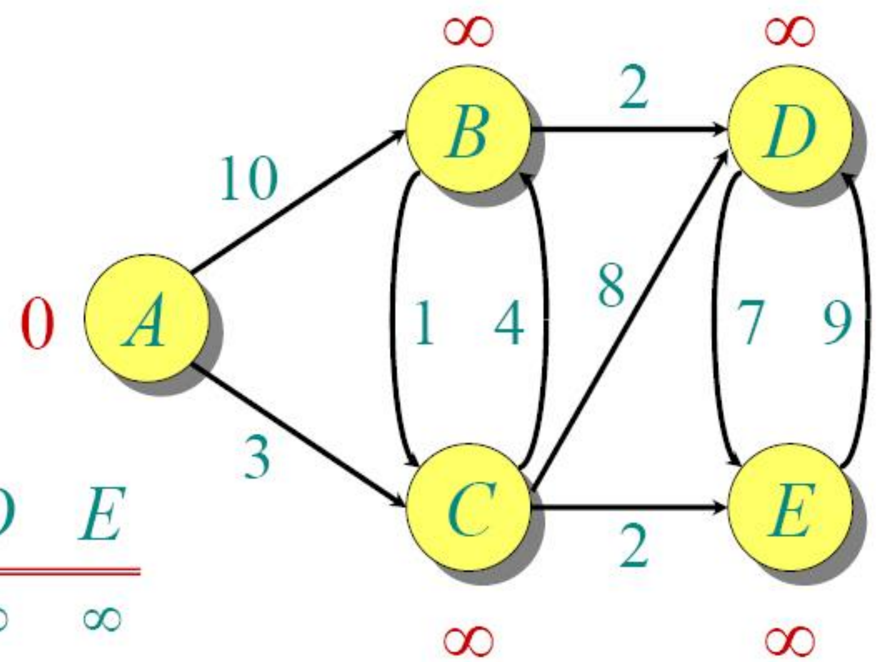
- **Dijkstra's algorithm** - is a solution to the single-source shortest path problem in graph theory.
- Works on both directed and undirected graphs. However, all edges must have nonnegative weights.
- Approach: Greedy
- Input: Weighted graph G={E,V} and source vertex $v \in V$, such that all edge weights are nonnegative
- Output: Lengths of shortest paths (or the shortest paths themselves) from a given source vertex $v \in V$ to all other vertices

Dijkstra's Algorithm: Pseudocode

dist[s] ←0	(distance to source vertex is zero)
for all $v \in V-\{s\}$	
do dist[v] ←∞	(set all other distances to infinity)
$S \leftarrow \emptyset$	(S, the set of visited vertices is initially empty)
$Q \leftarrow V$	(Q, the queue initially contains all vertices)
while $Q \neq \emptyset$	(while the queue is not empty)
do u ← mindistance(Q,dist)	(select the element of Q with the min. distance)
$S \leftarrow S \cup \{u\}$	(add u to list of visited vertices)
for all $v \in$ neighbors[u]	
do if dist[v] > dist[u] + w(u, v)	(if new shortest path found)
then d[v] ←d[u] + w(u, v)	(set new value of shortest path)
	(if desired, add traceback code)
return dist	

Dijkstra Example

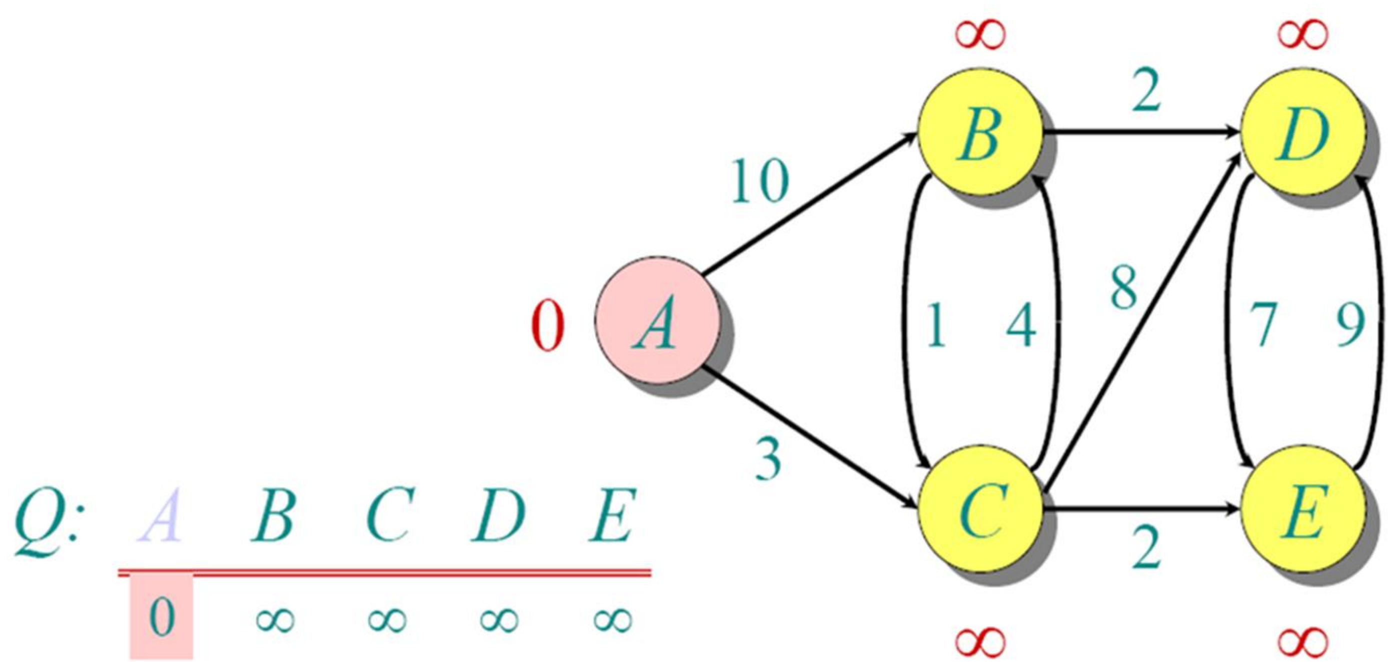
Initialize:



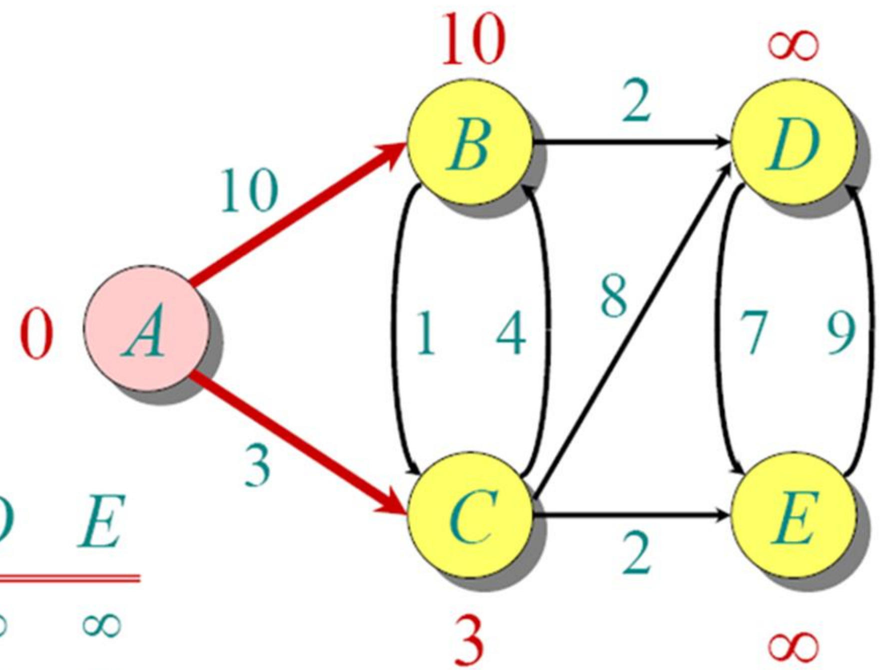
Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: {}

Dijkstra Example



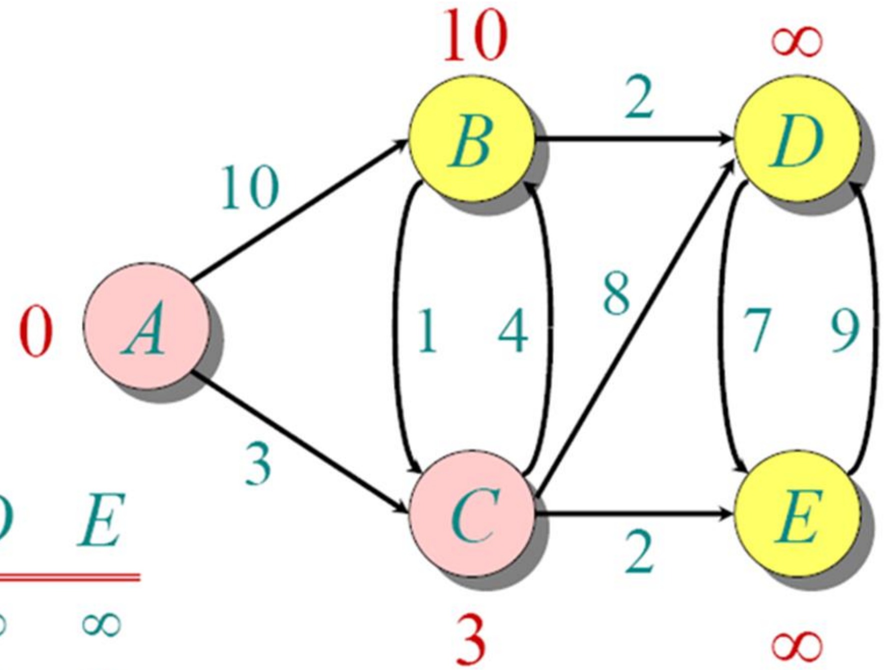
Dijkstra Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A }

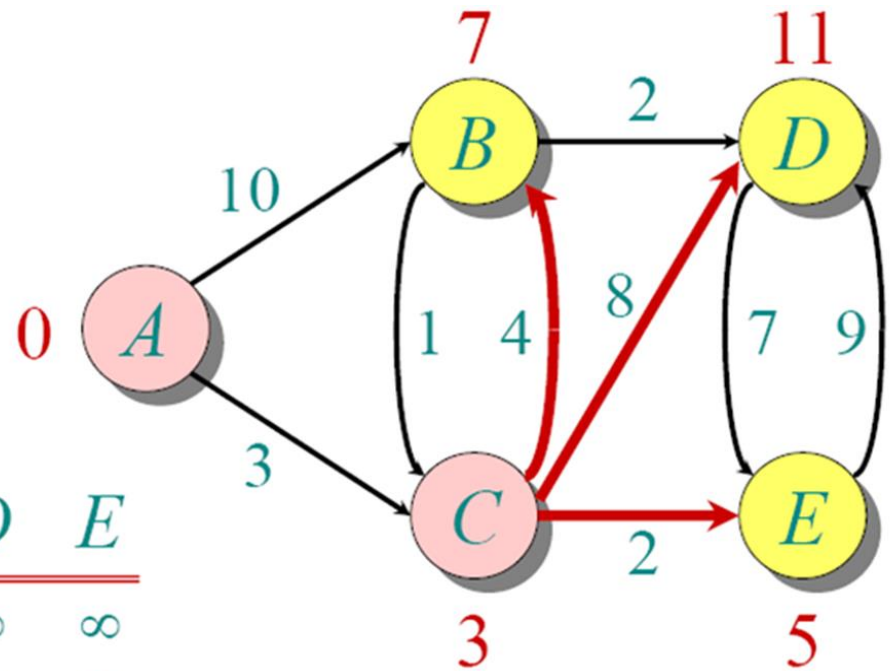
Dijkstra Example



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A, C }

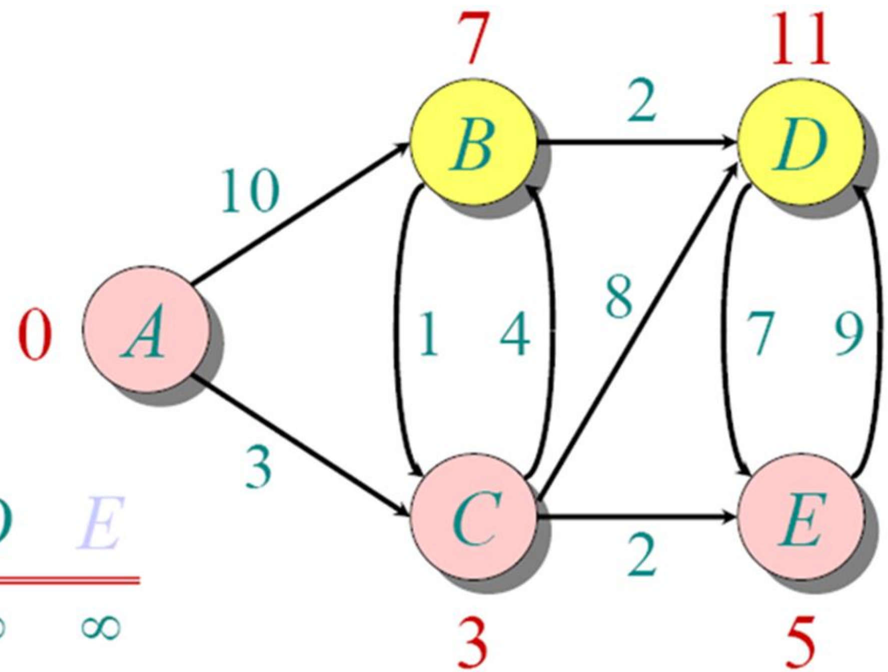
Dijkstra Example



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

S: { A, C }

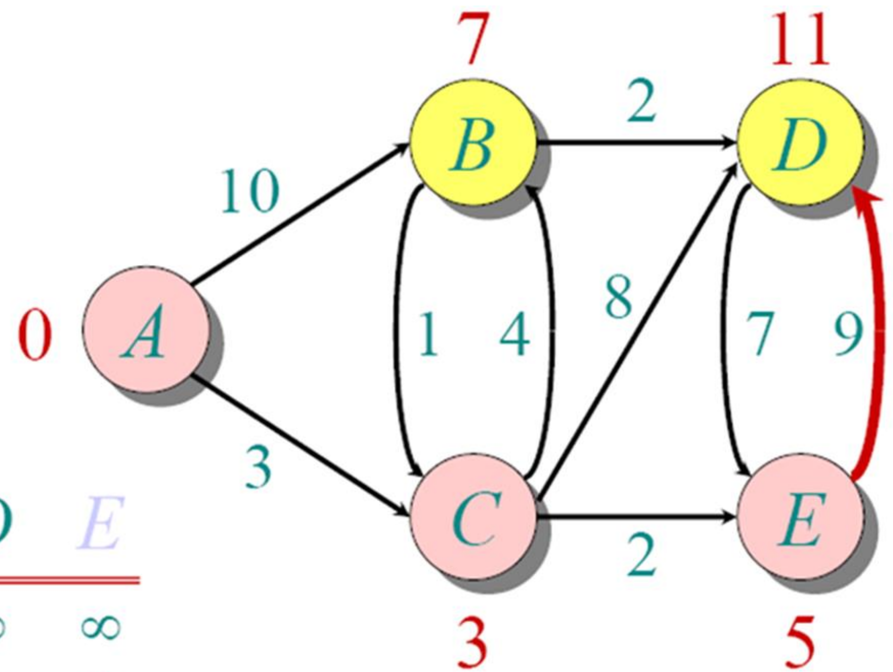
Dijkstra Example



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

S: { A, C, E }

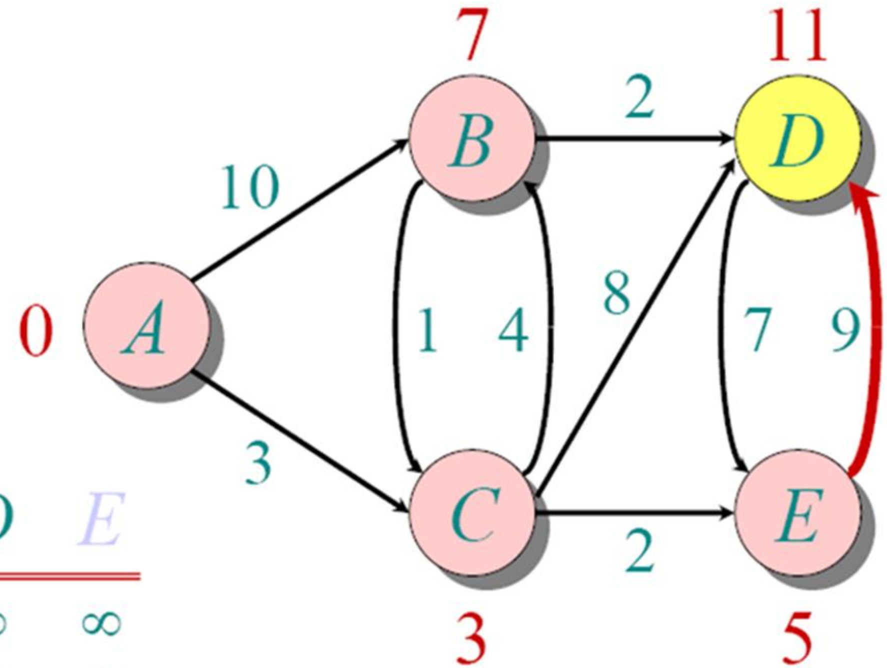
Dijkstra Example



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

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

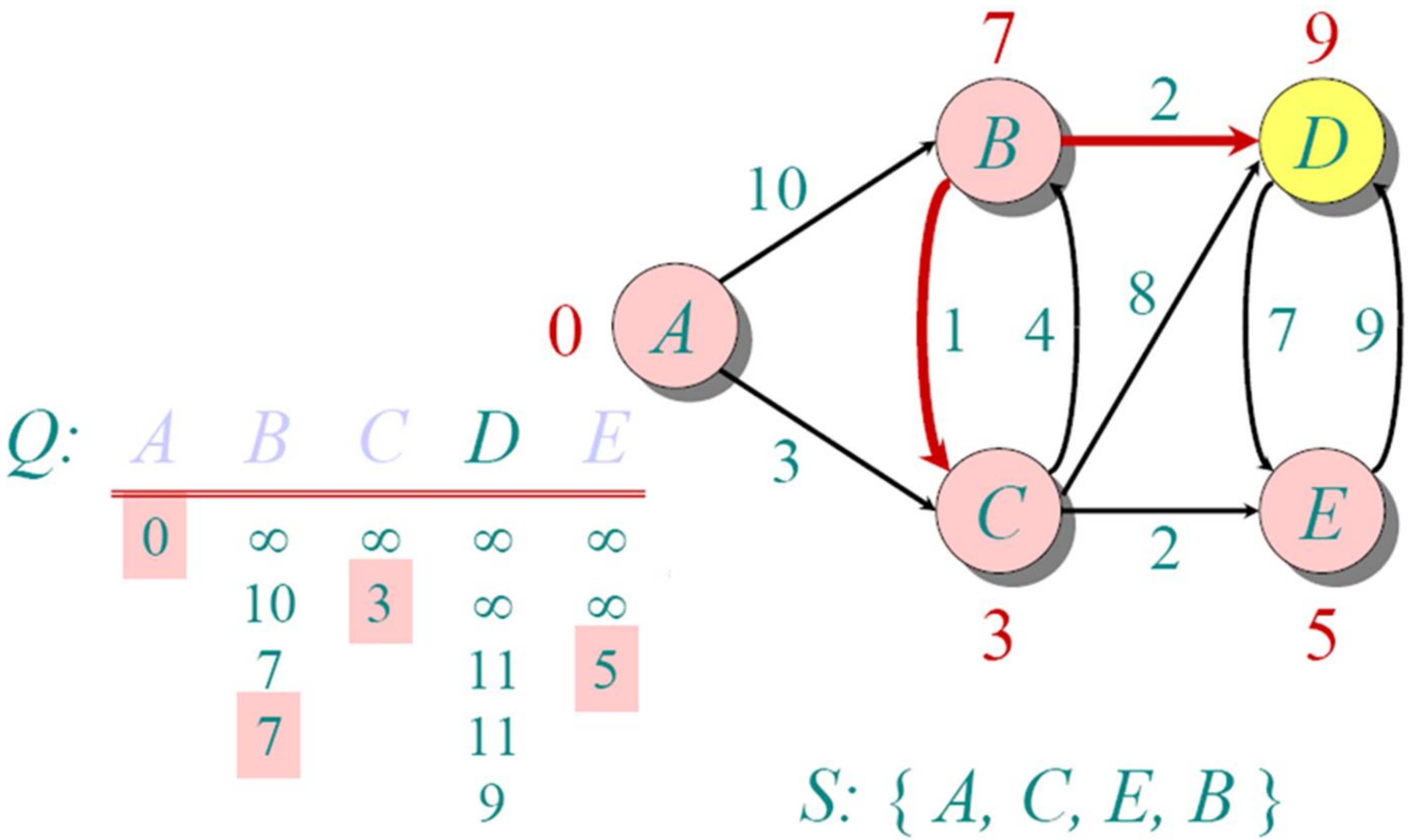
Dijkstra Example



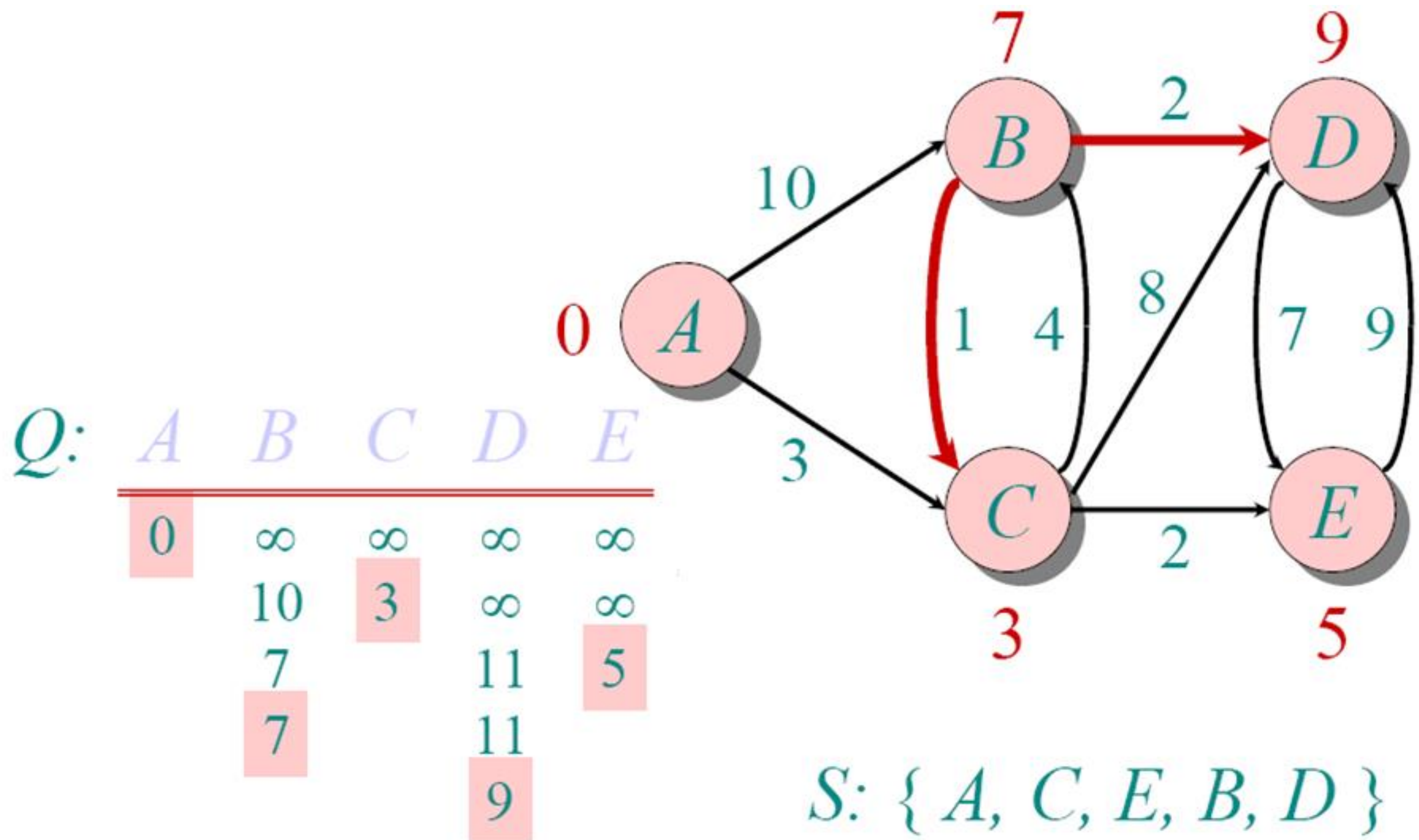
Q:	A	B	C	D	E
	$\boxed{0}$	∞	∞	∞	∞
		10	$\boxed{3}$	∞	∞
		7		11	$\boxed{5}$
		$\boxed{7}$		11	

S: { A, C, E, B }

Dijkstra Example

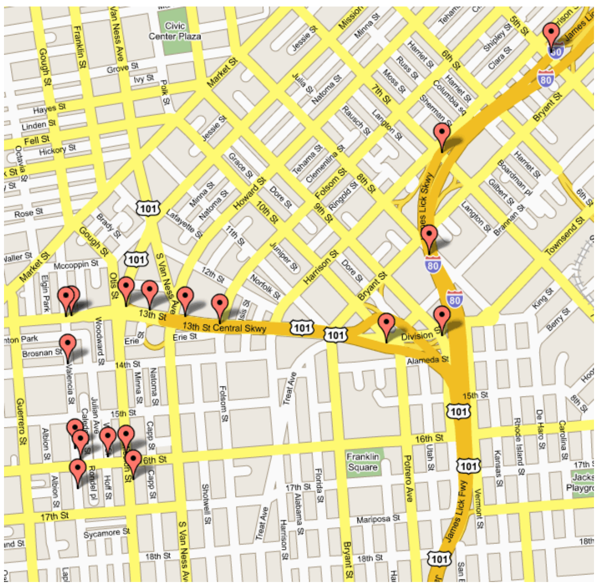


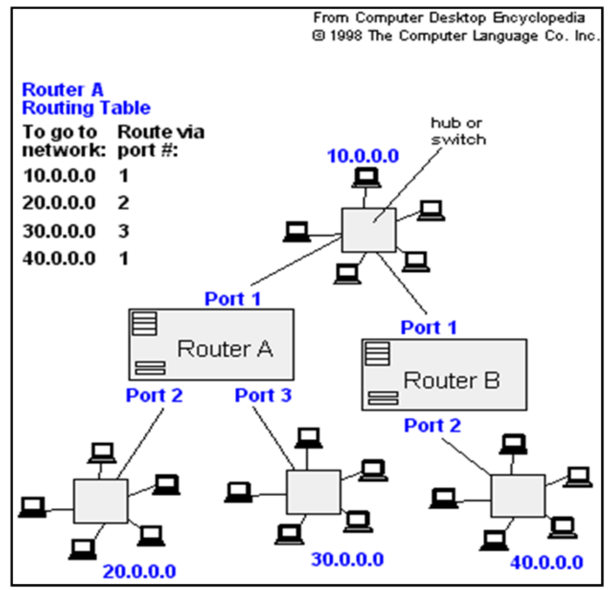
Dijkstra Example



Applications of Dijkstra's Algorithm

- Traffic Information Systems are most prominent use.
- Mapping (Map Quest, Google Maps)
- Routing Systems





Thank you!