Dijkstra’s Algotithm

part –II

Karim City College
P.G. Department of Mathematics
Faculty: Prof. Yashmin Banu
B.Sc(Math Hons) Sem-3

Introduction

Dijkstra's Algorithm derived by a Dutch computer scientist '**Edsger Wybe Dijkstra**' in 1956 and published in 1959

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, such that all edge weights are nonnegative

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

How it works ?

This algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities.

Numerical Algorithm

Formula :

$O(|V|^2+|E|) = O(|V|^2)$

Where,

E= Edges, V= Vertices

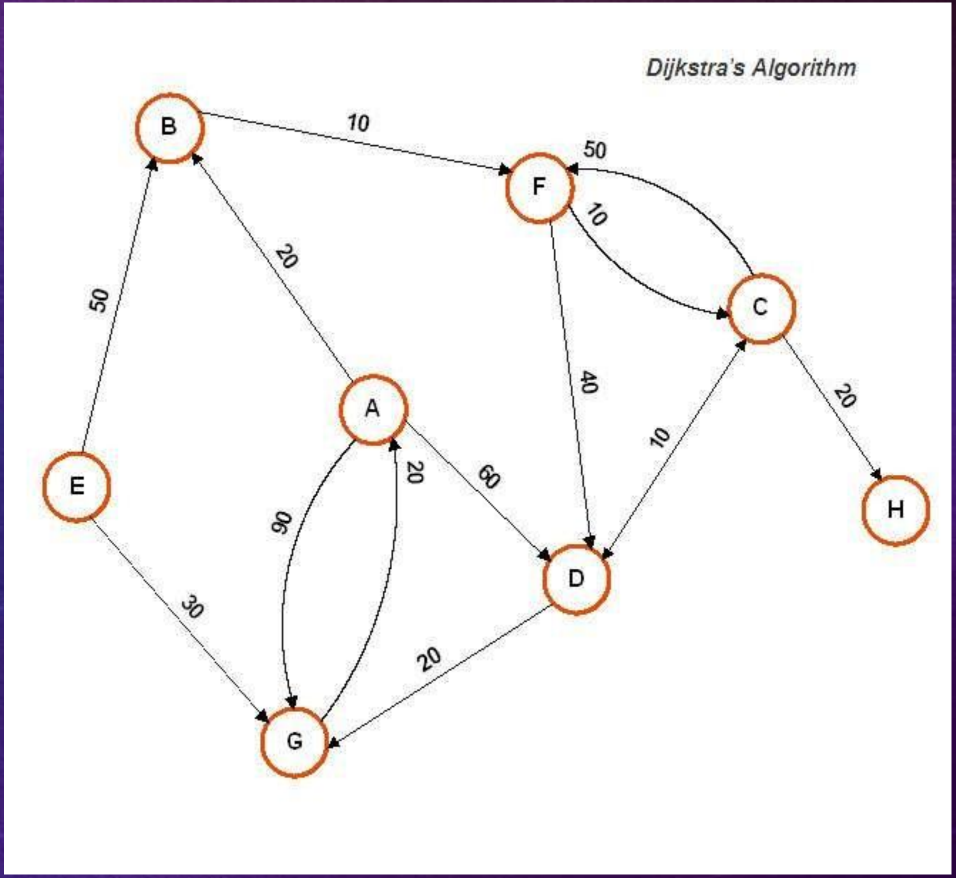
$|E|$ = Function of Edges

$|V|$ = Function of Vertices and

O = Constant

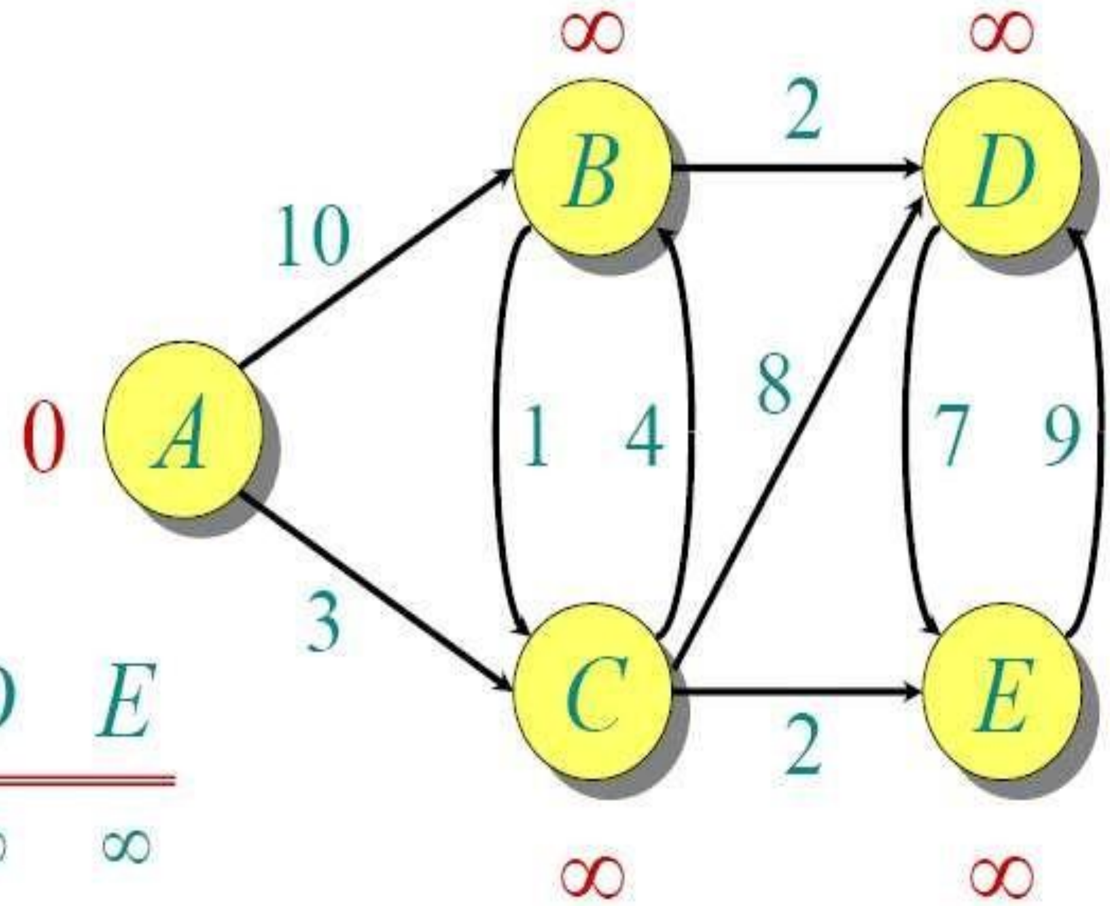
Graph Algorithm

- In this interconnected 'Vertex' we'll use 'Dijkstra's Algorithm'.
- To use this algorithm in this network we have to start from a decided vertex and then continue to others.



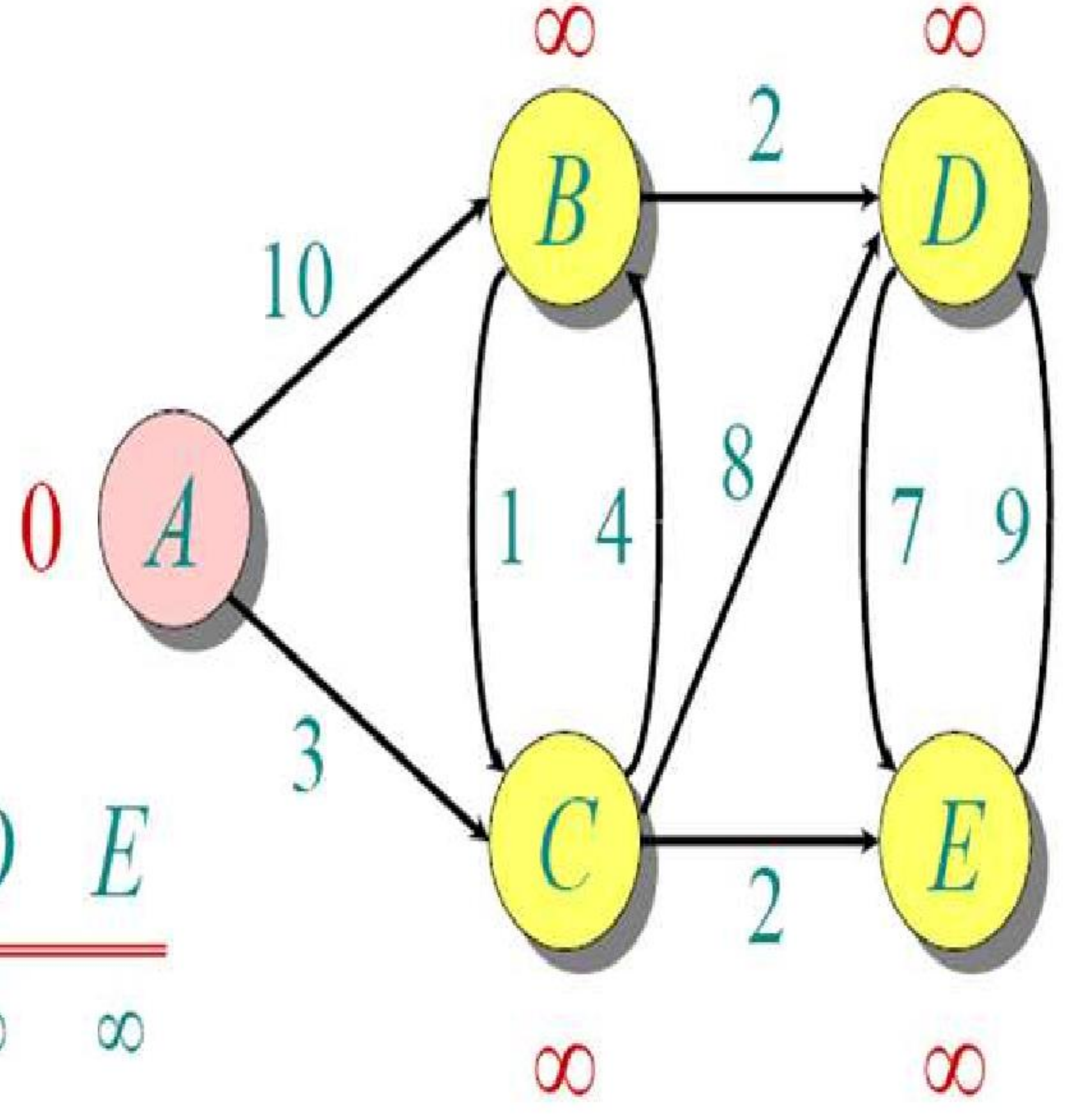
Dijkstra Animated Example

Initialize:

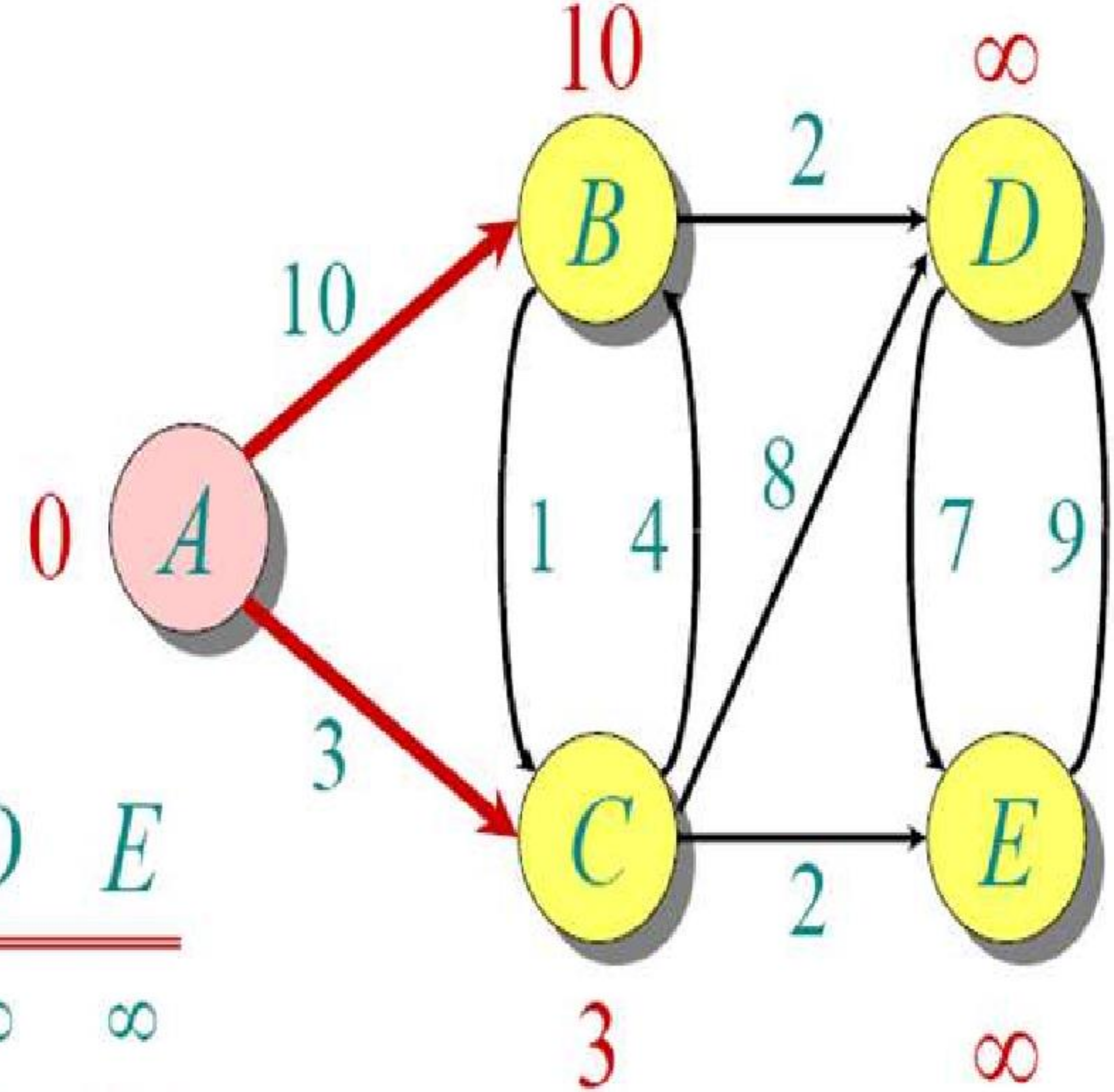


Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: {}

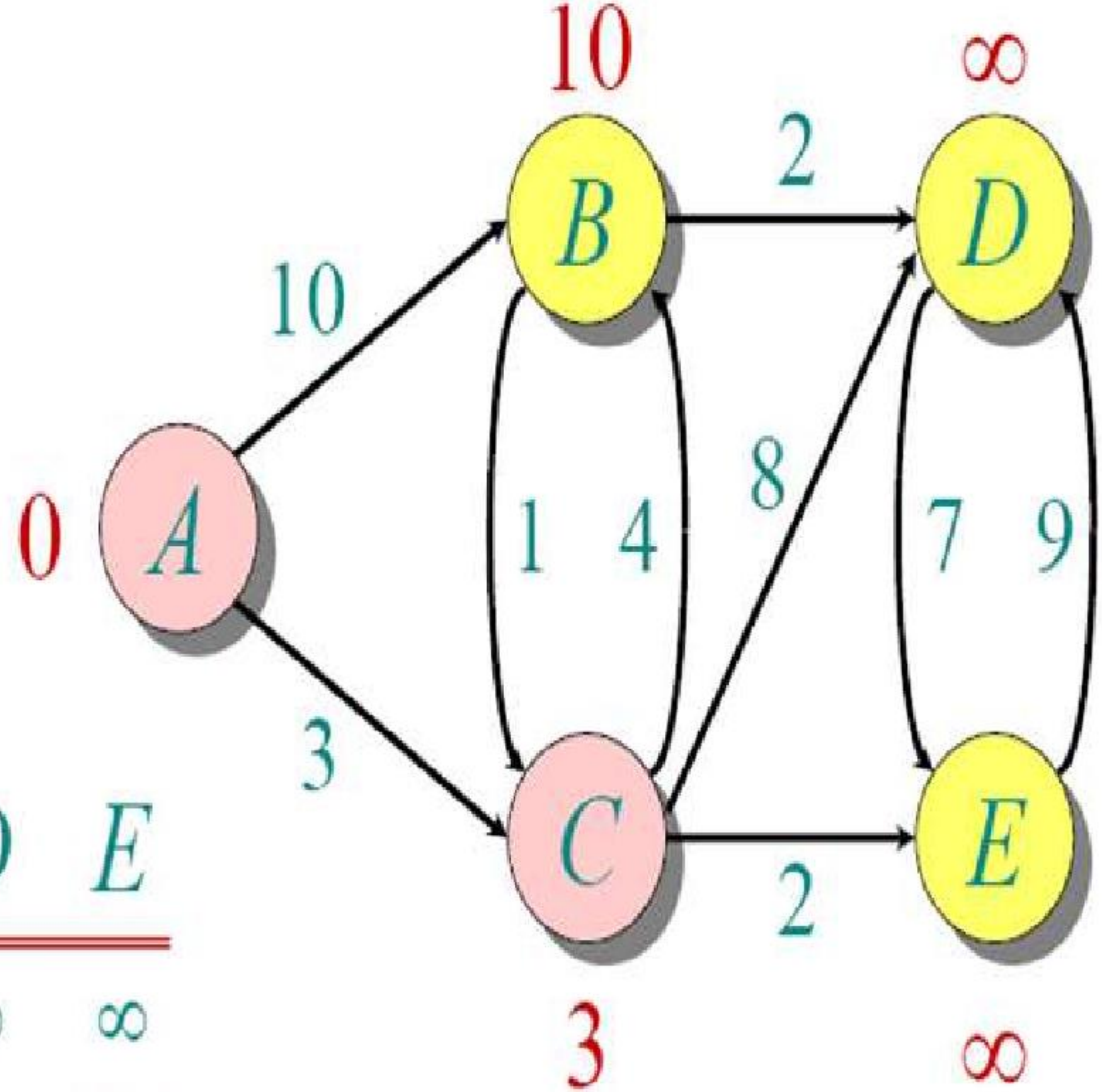


Q:	A	B	C	D	E
	0	∞	∞	∞	∞



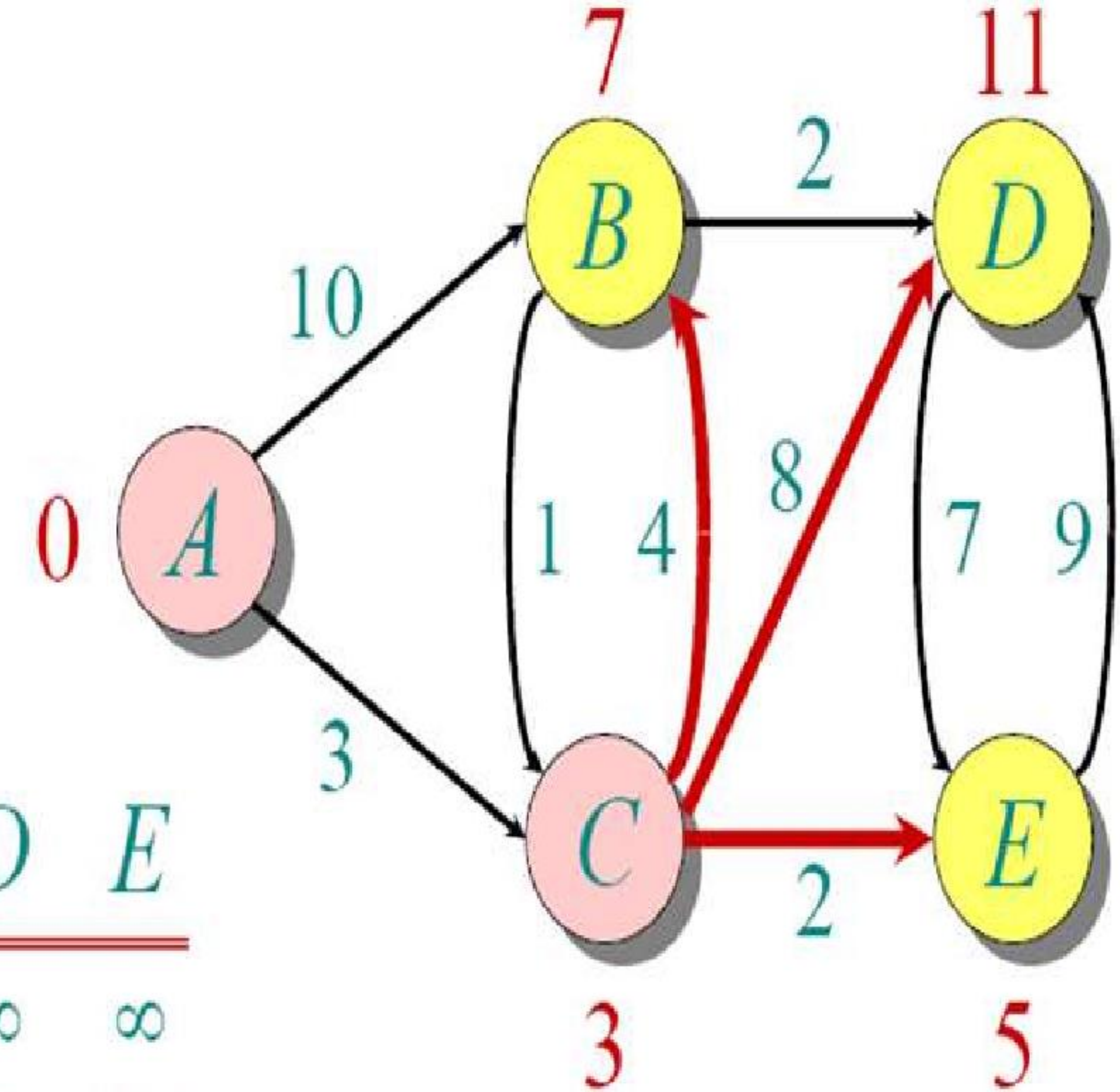
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A }



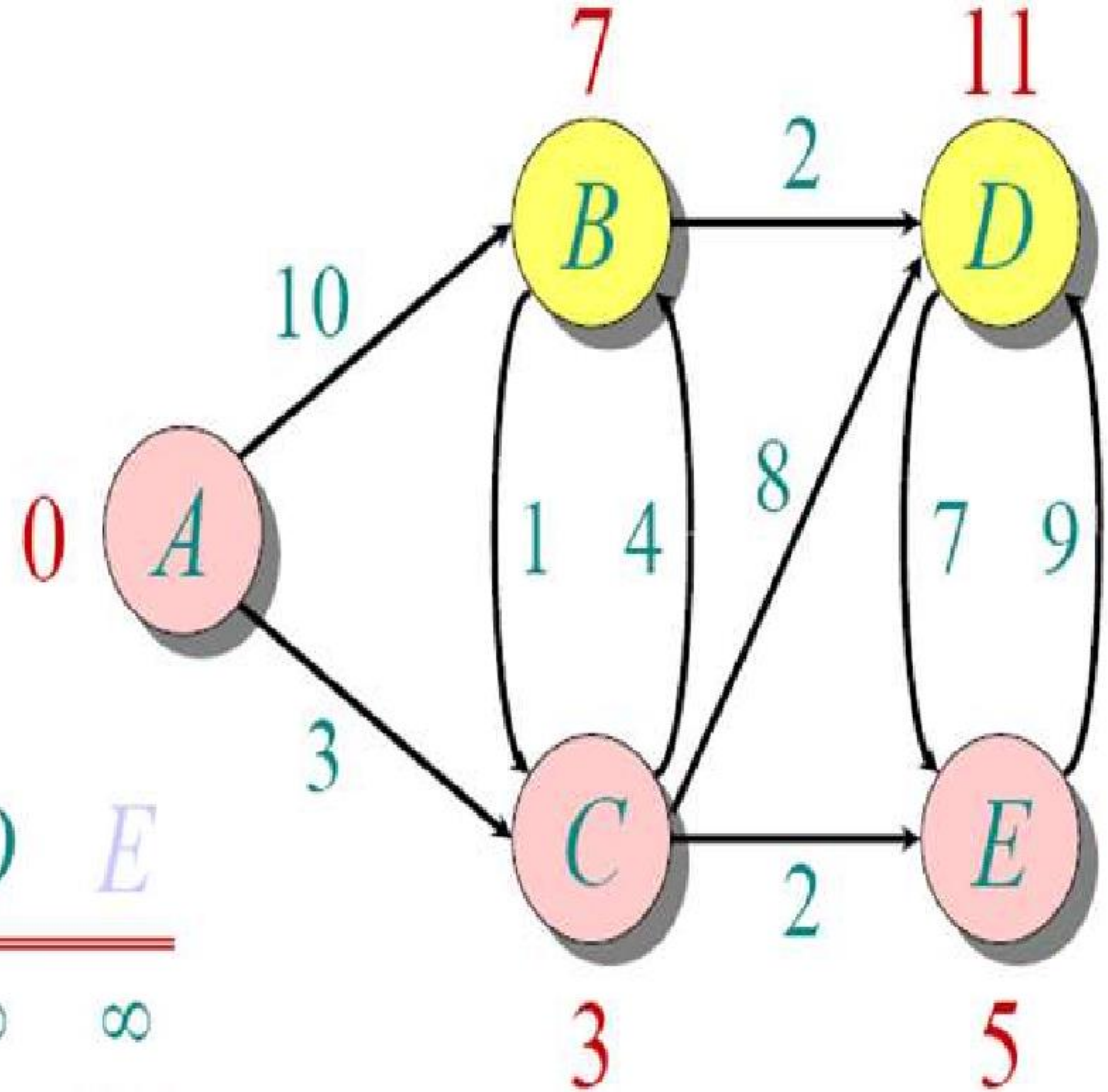
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A, C }



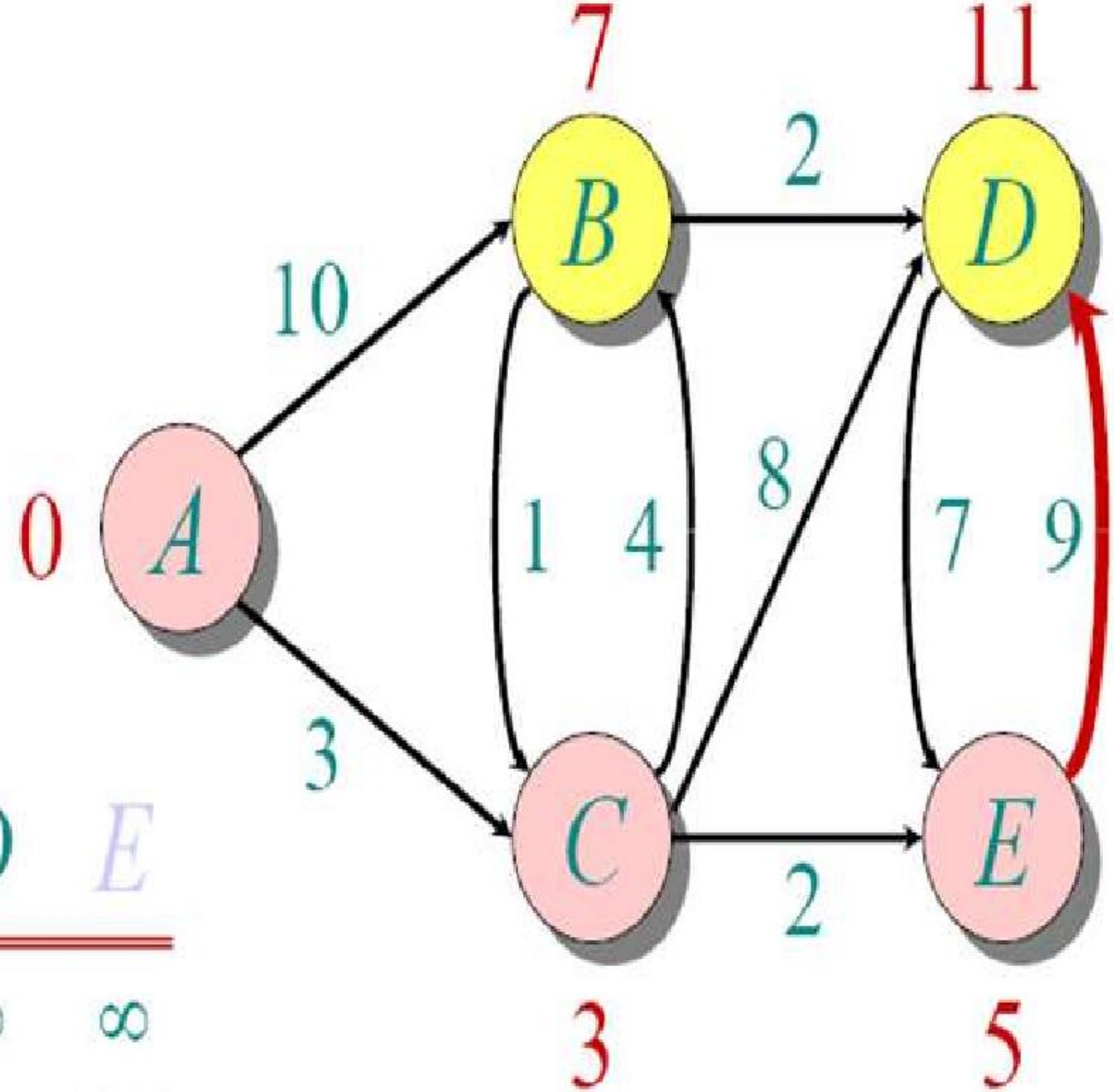
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C }



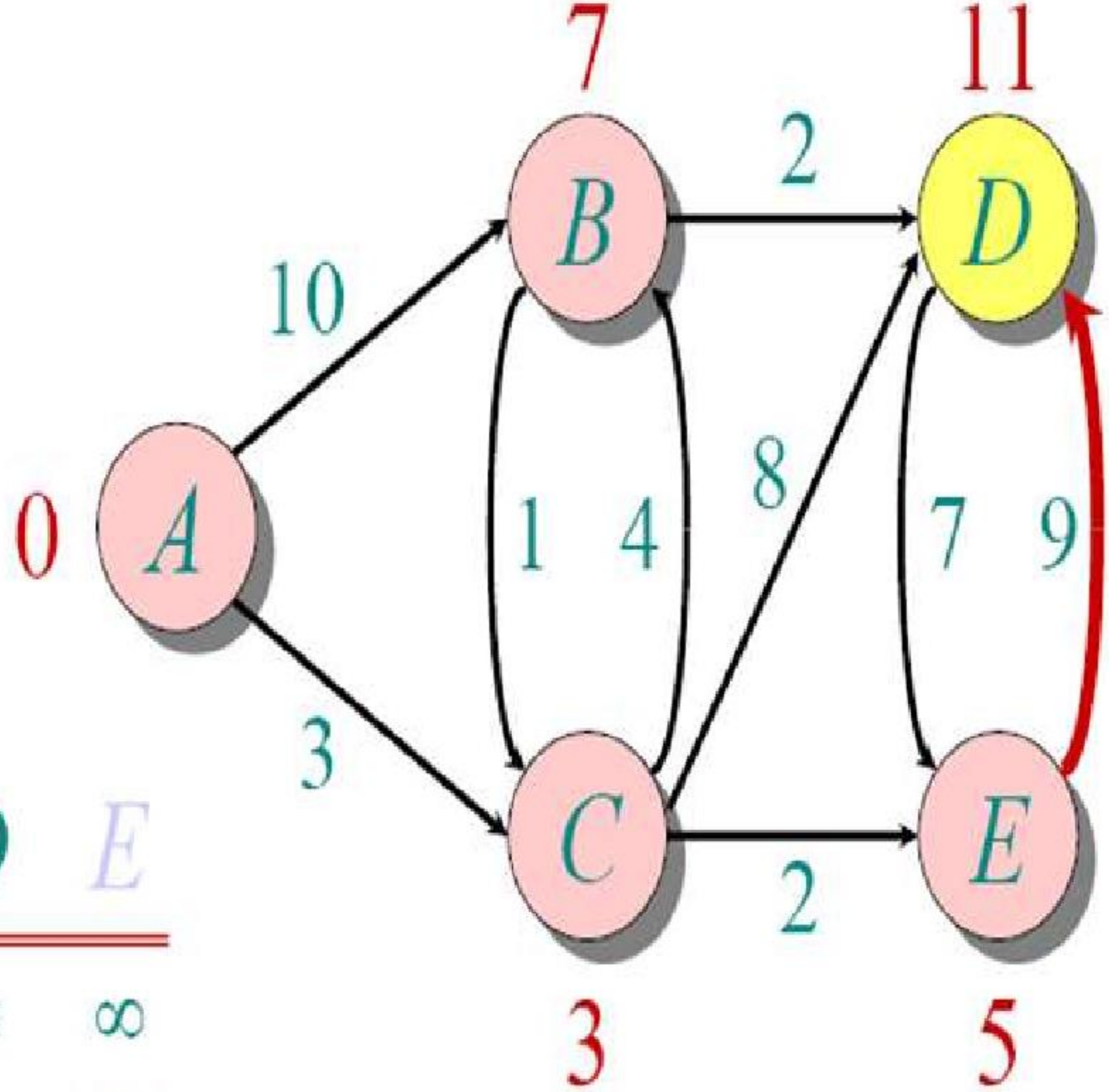
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C, E }



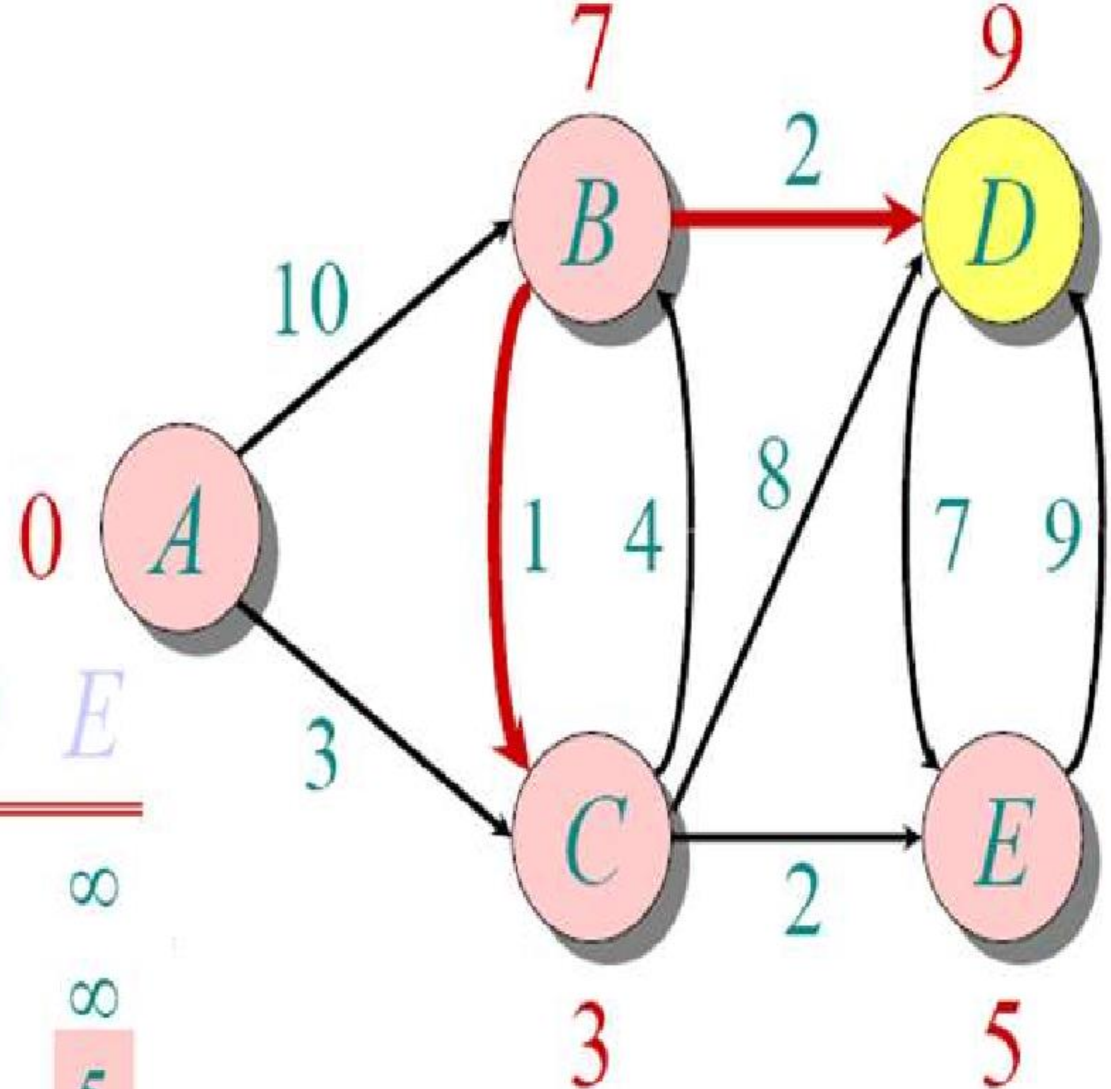
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E }



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

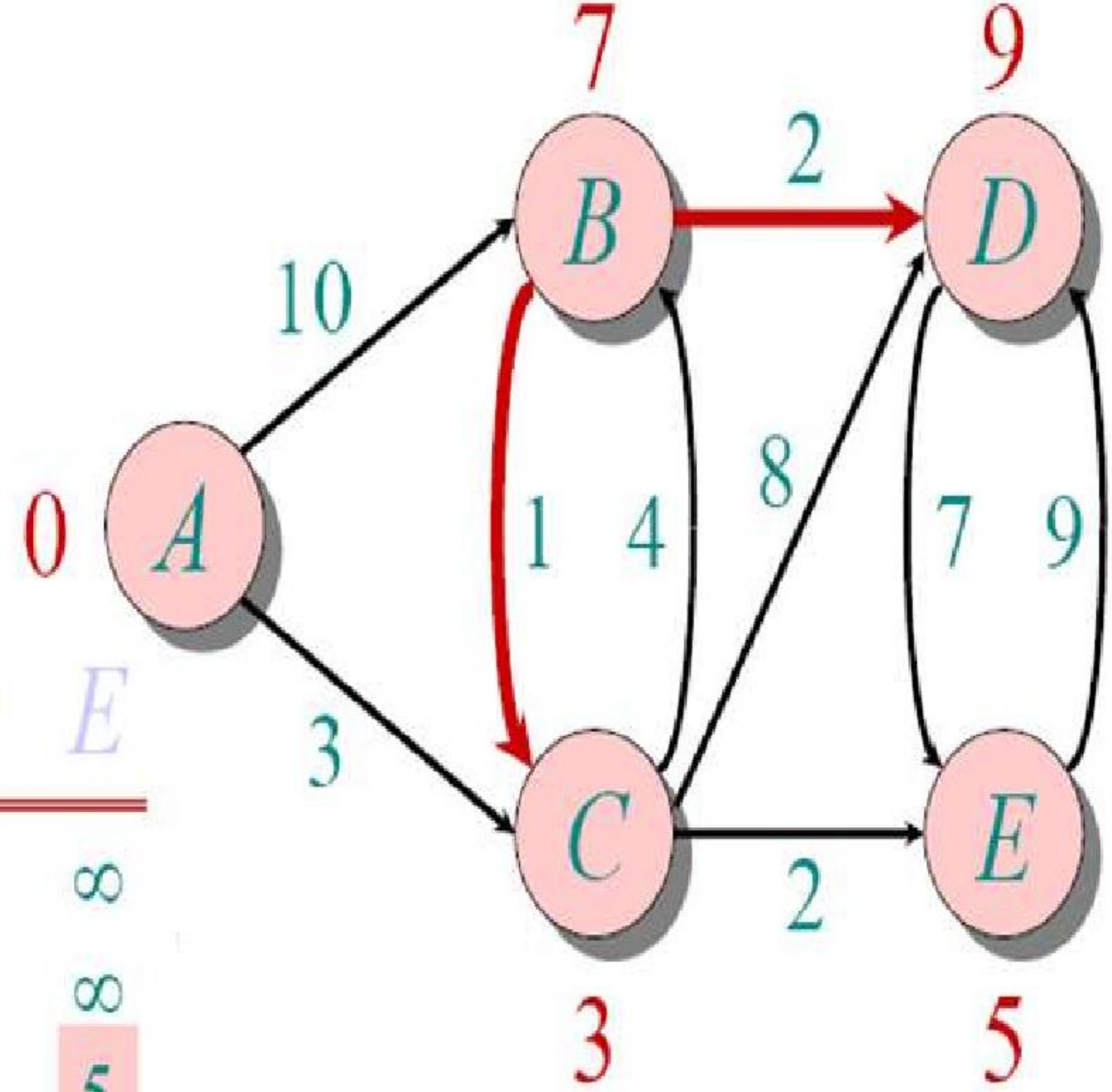
S: { A, C, E, B }



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

S: { A, C, E, B }

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



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

Implementation & Run Time

The simplest implementation is to store vertices in an **Array** or **Linked list**. This will produce a running time of

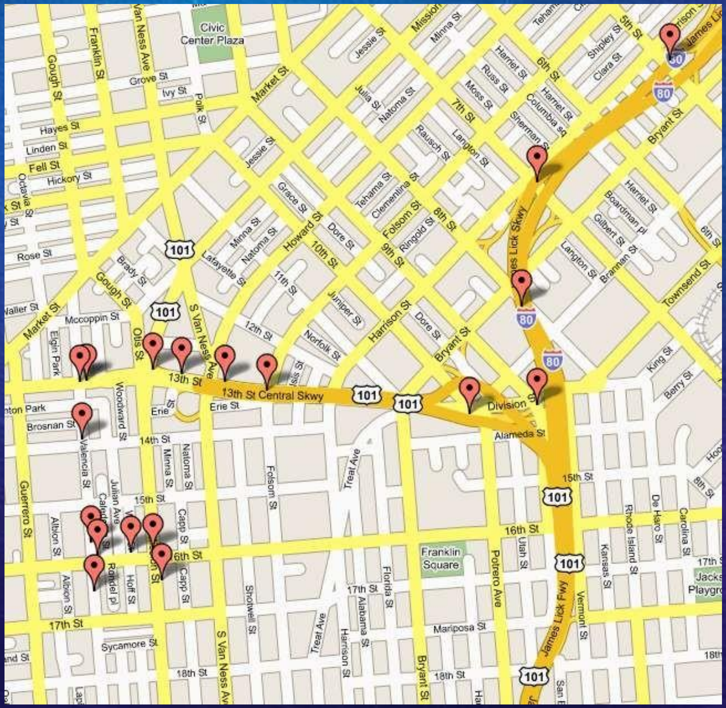
$O(|V|^2 + |E|)$

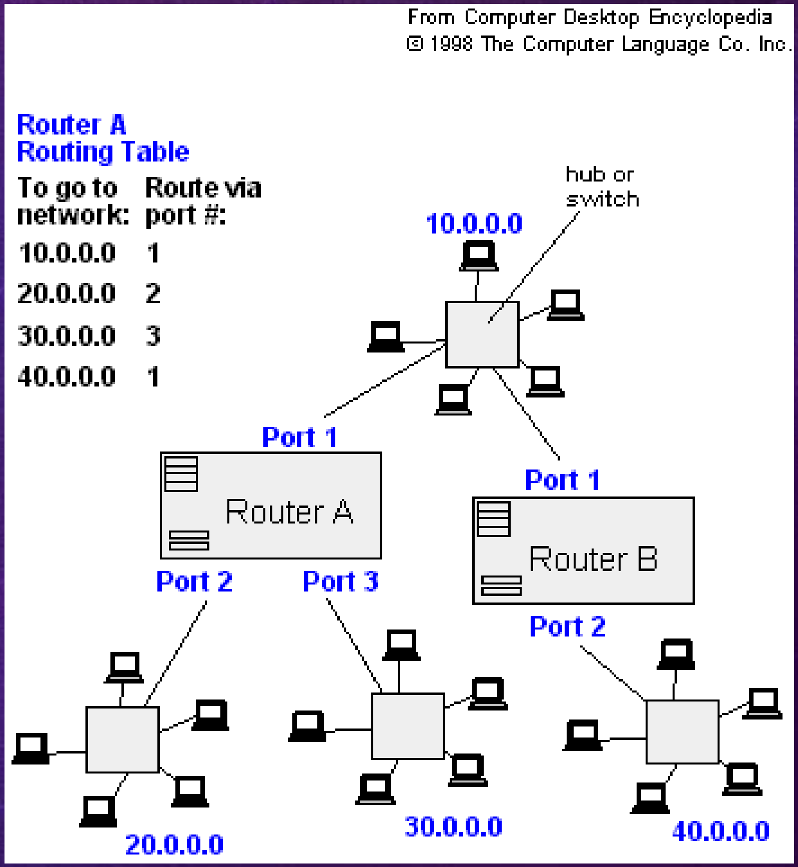
For graphs with very few edges and many nodes, it can be implemented more efficiently storing the graph in an adjacency list using a **Binary Heap** or **Priority Queue**. This will produce a running time of

$O((|E|+|V|) \log |V|)$

Application :

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





Thank You