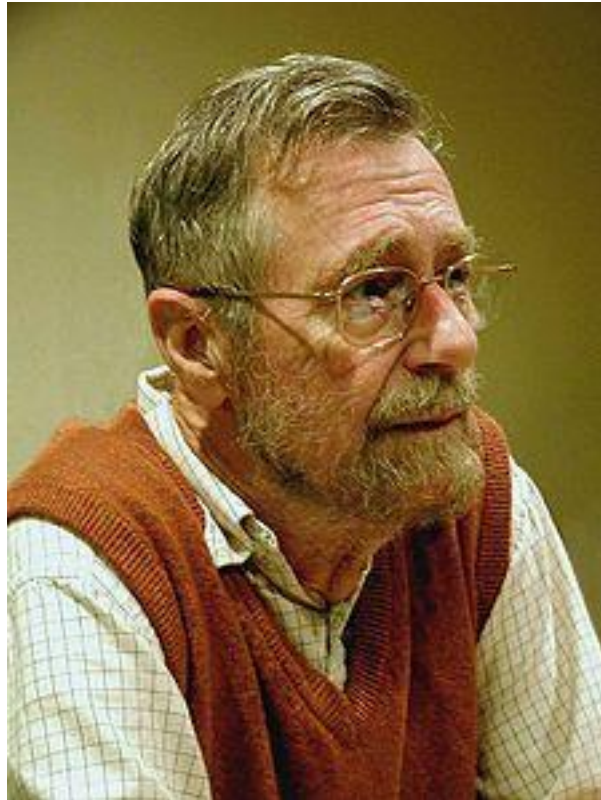


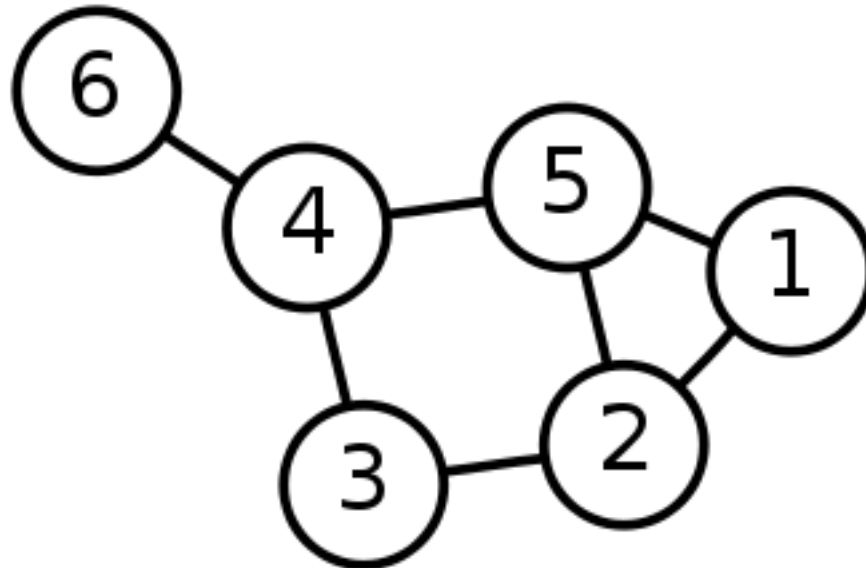
# Dijkstra's Algorithm

# The author: Edsger Wybe Dijkstra



# Single-Source Shortest Path Problem

- ปัญหา Single-Source Shortest Path
- เป็นปัญหาในการหาเส้นทางที่สั้นที่สุดจากโหนด  $v$  ไปยังทุกโหนดที่เหลือในกราฟ



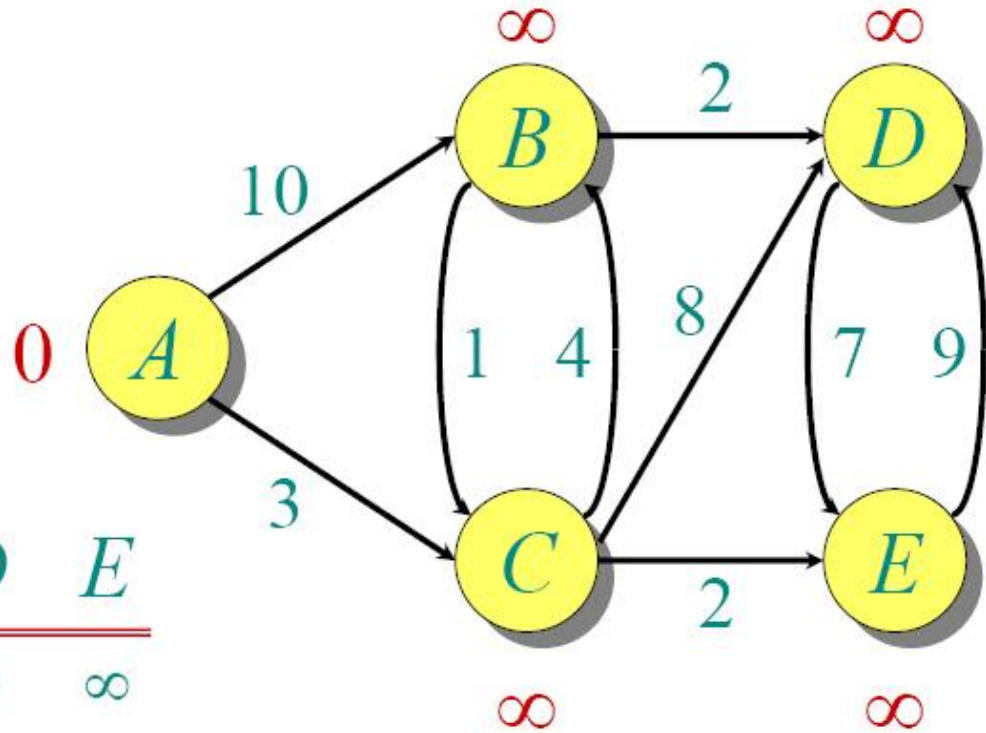
# Dijkstra's algorithm

- Dijkstra's algorithm ให้คำตอบกับปัญหา single-source shortest path โดยทำงานได้ทั้งกราฟแบบมีทิศทางและไม่มีทิศทาง แต่มีเงื่อนไขว่า ทุกเส้นเชื่อมต้องไม่มีค่าน้ำหนักเป็นลบ
- **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

dist[s] ← 0	(distance to source vertex is zero)
for all $v \in V - \{s\}$	
do dist[v] ← $\infty$	(set all other distances to infinity)
S ← $\emptyset$	(S, the set of visited vertices is initially empty)
Q ← 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 ← S $\cup$ {u}	(add u to list of visited vertices)
for all $v \in \text{neighbors}[u]$	
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 Animated Example

**Initialize:**

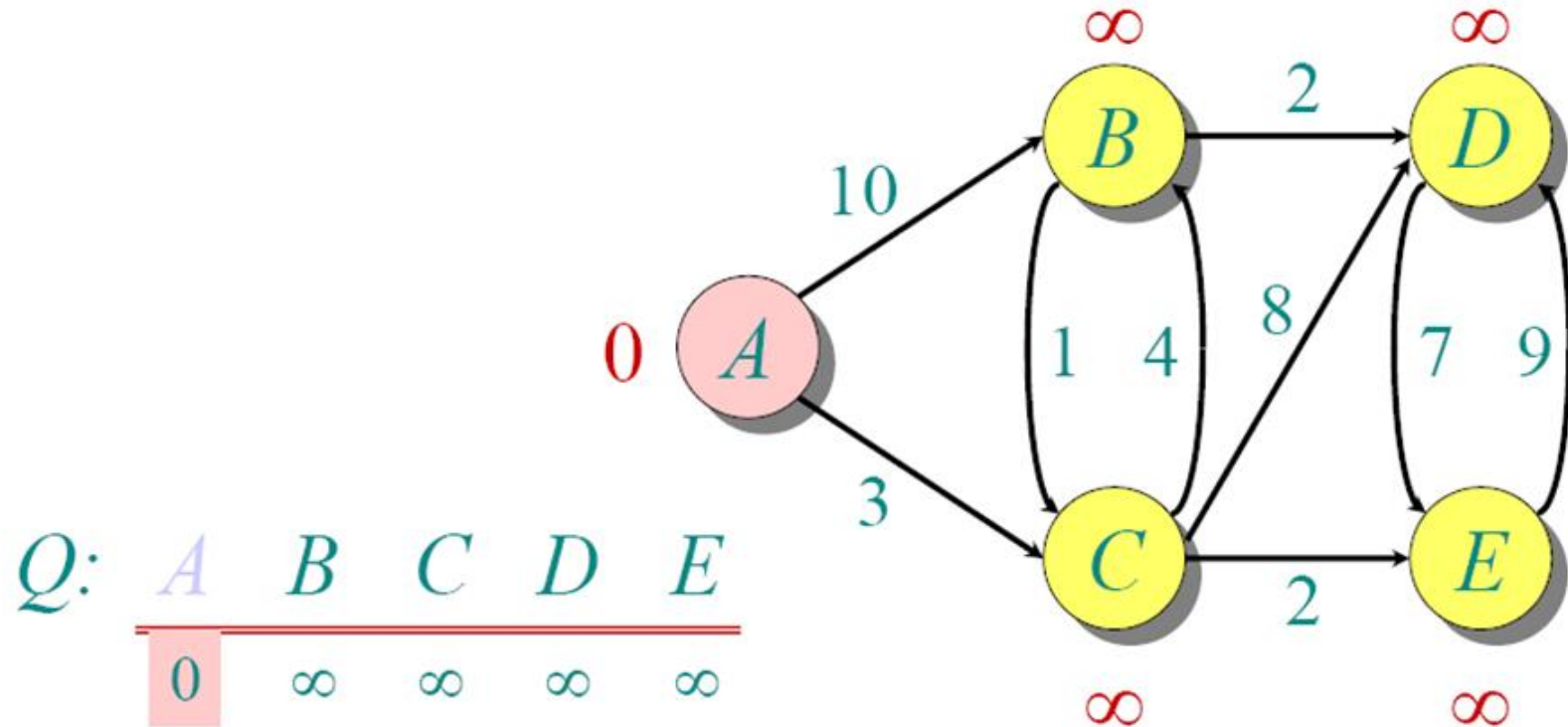


$Q:$

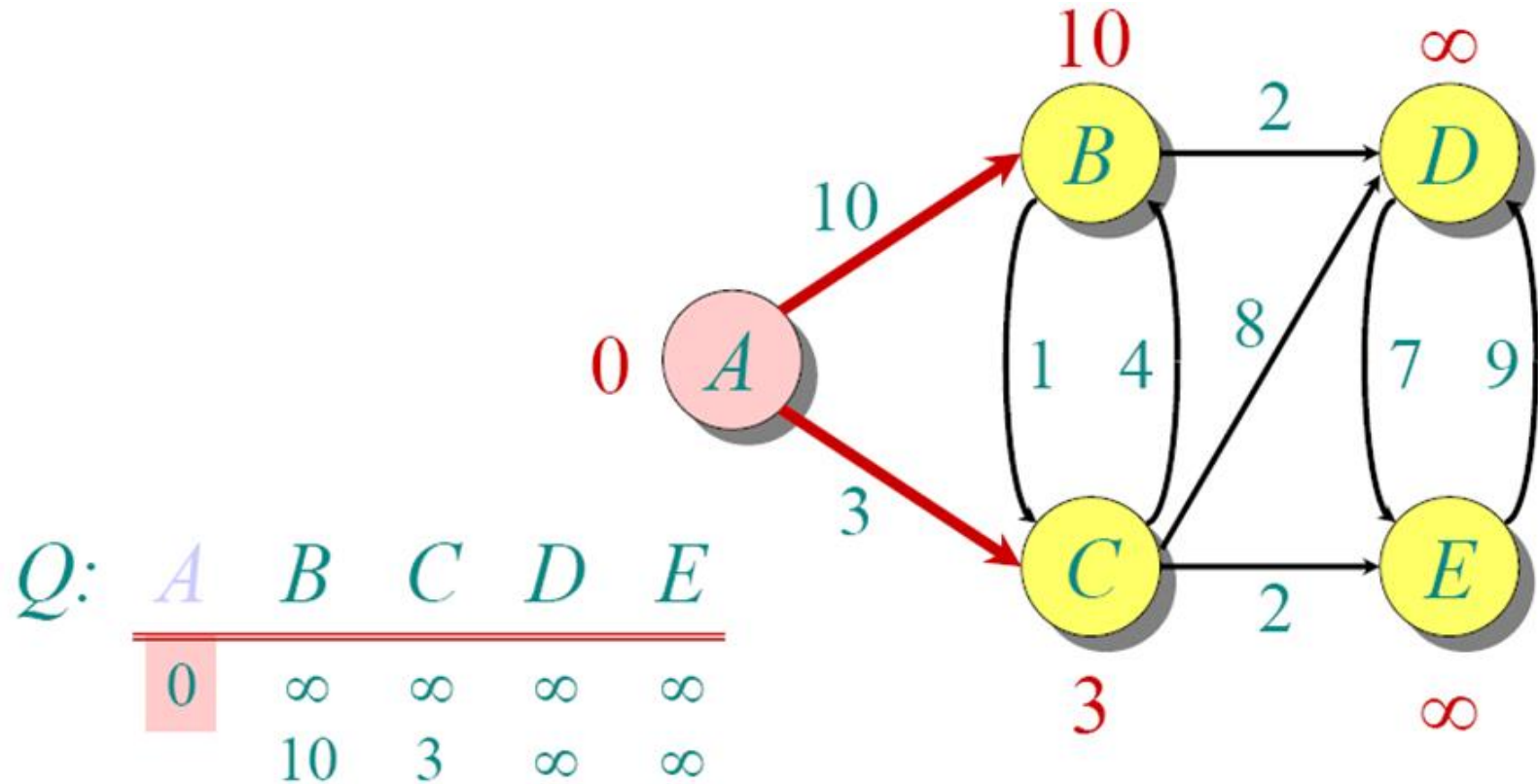
<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>
0	∞	∞	∞	∞

$S: \{\}$

# Dijkstra Animated Example



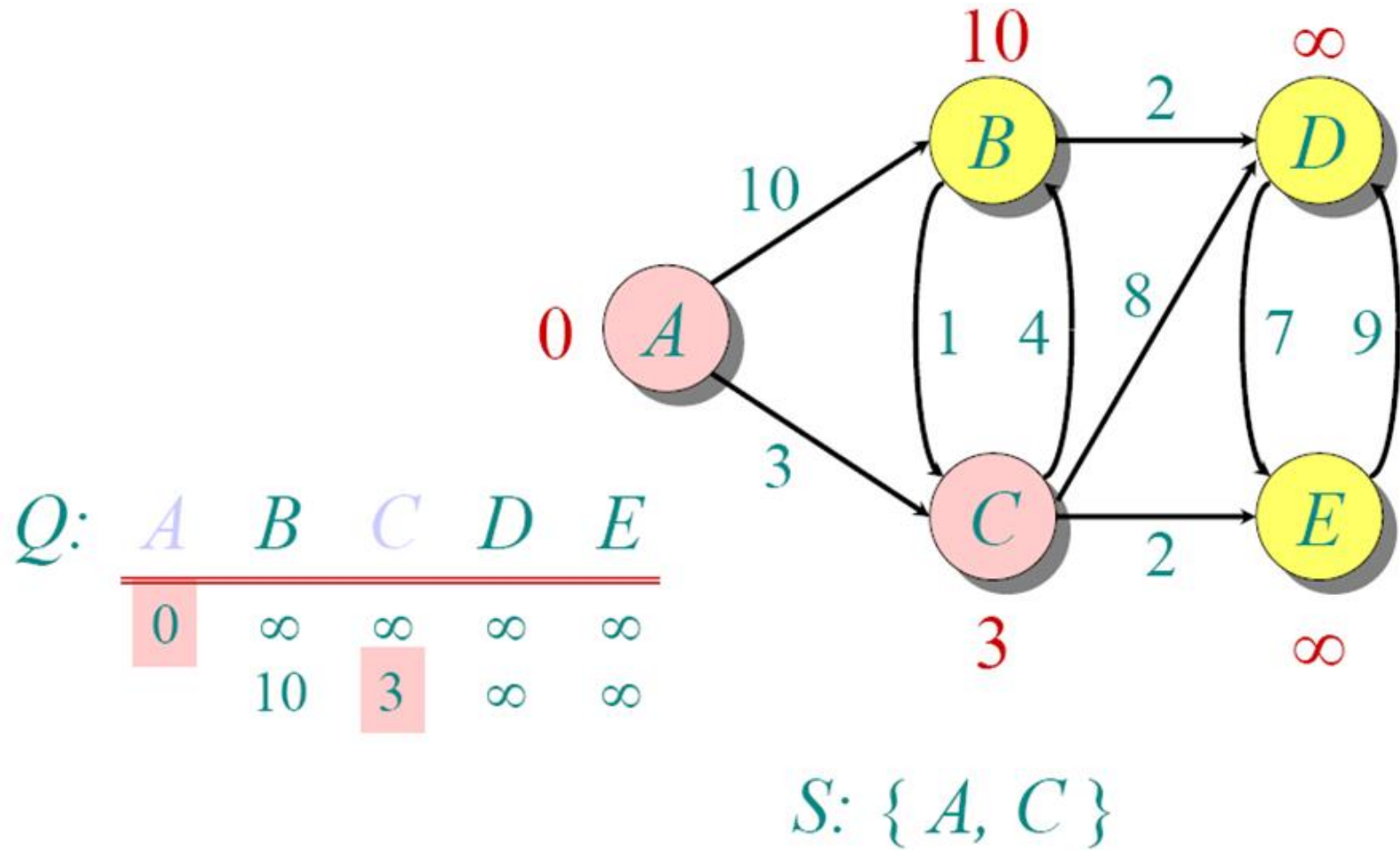
# Dijkstra Animated Example



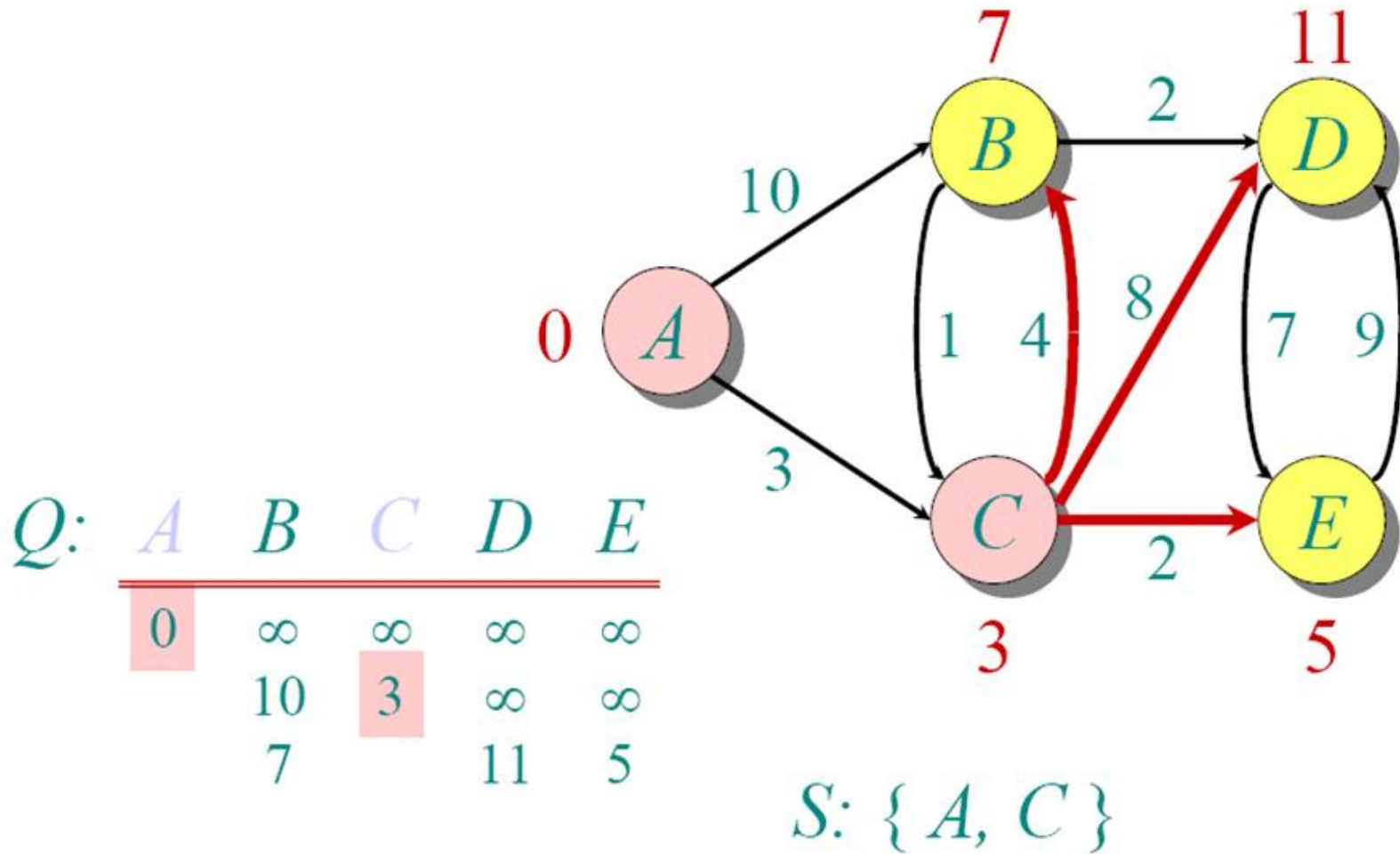
$S: \{A\}$



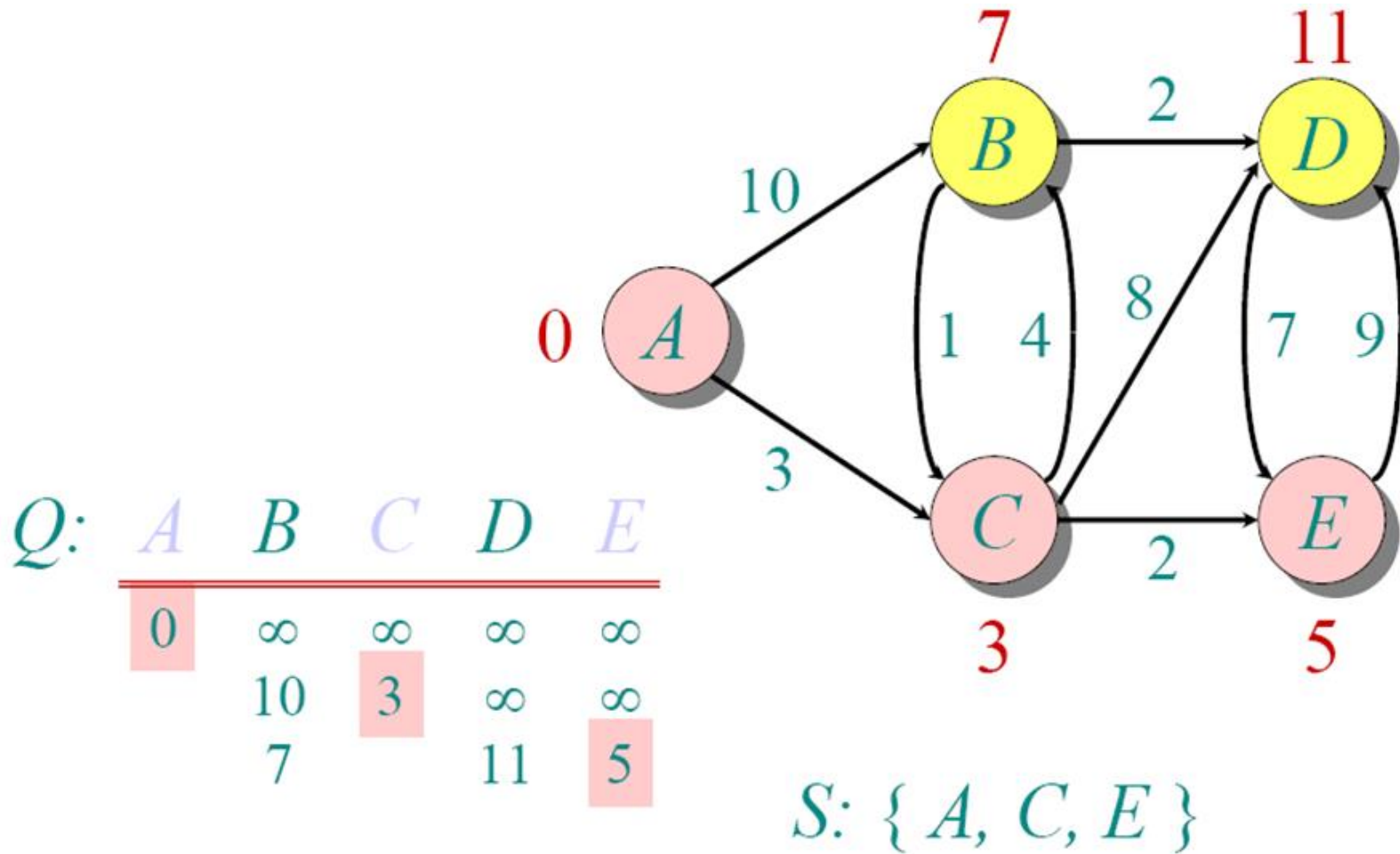
# Dijkstra Animated Example



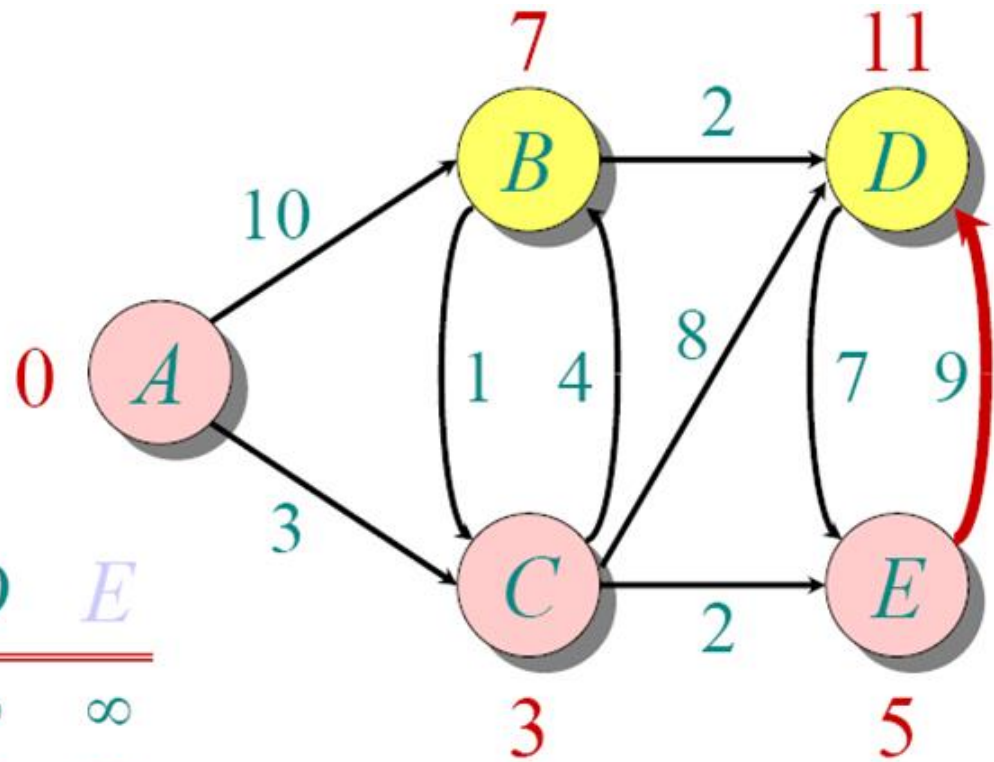
# Dijkstra Animated Example



# Dijkstra Animated Example



# Dijkstra Animated Example

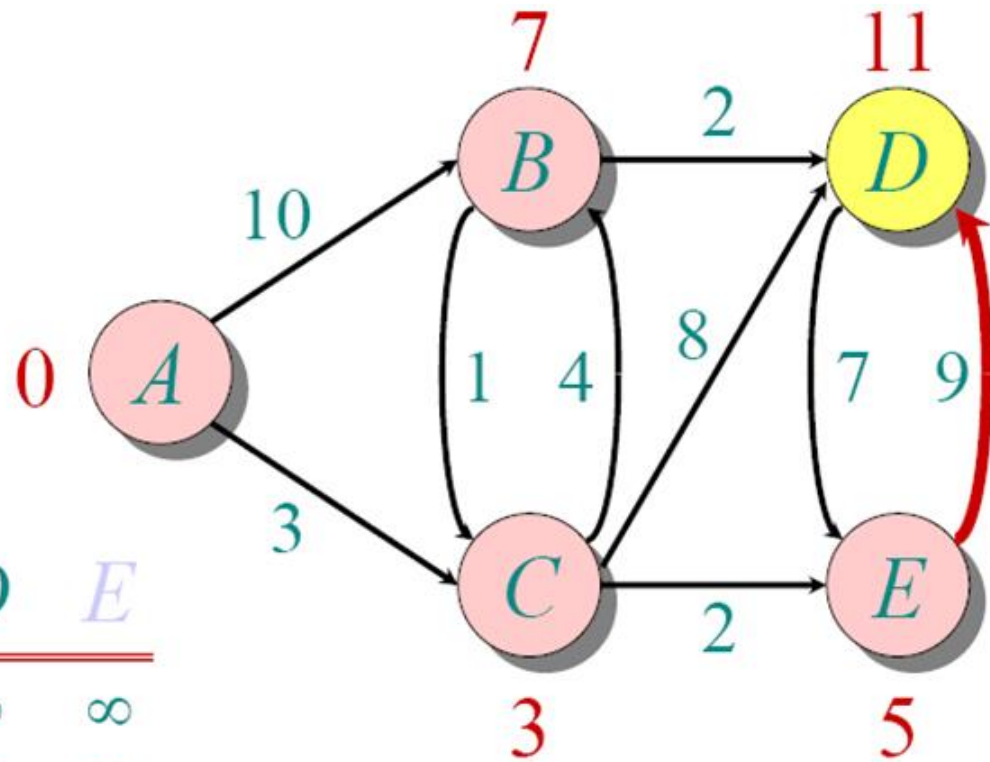


*Q:*

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
0	$\infty$	$\infty$	$\infty$	$\infty$
	10	3	$\infty$	$\infty$
	7		11	5
	7		11	

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

# Dijkstra Animated Example

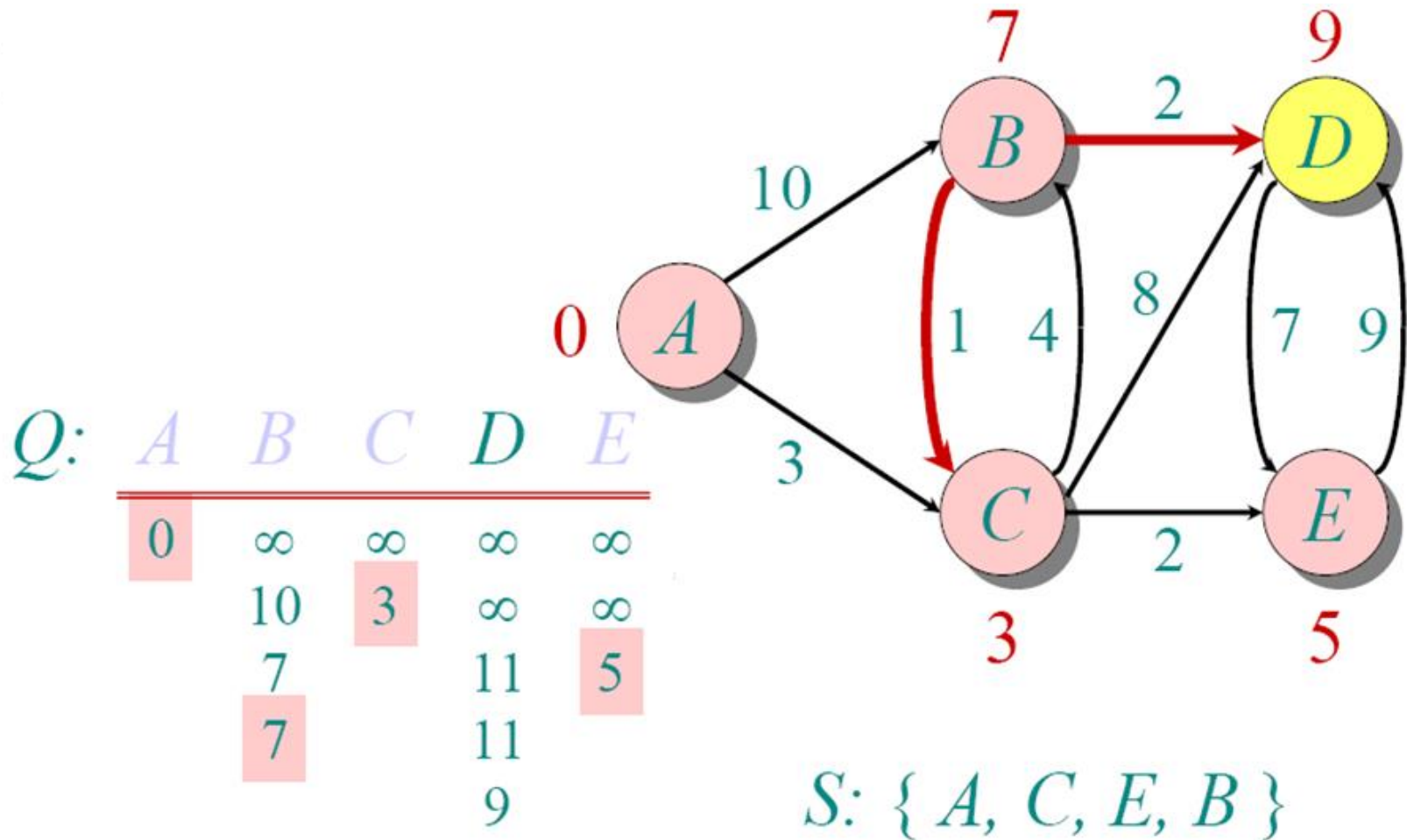


Q:

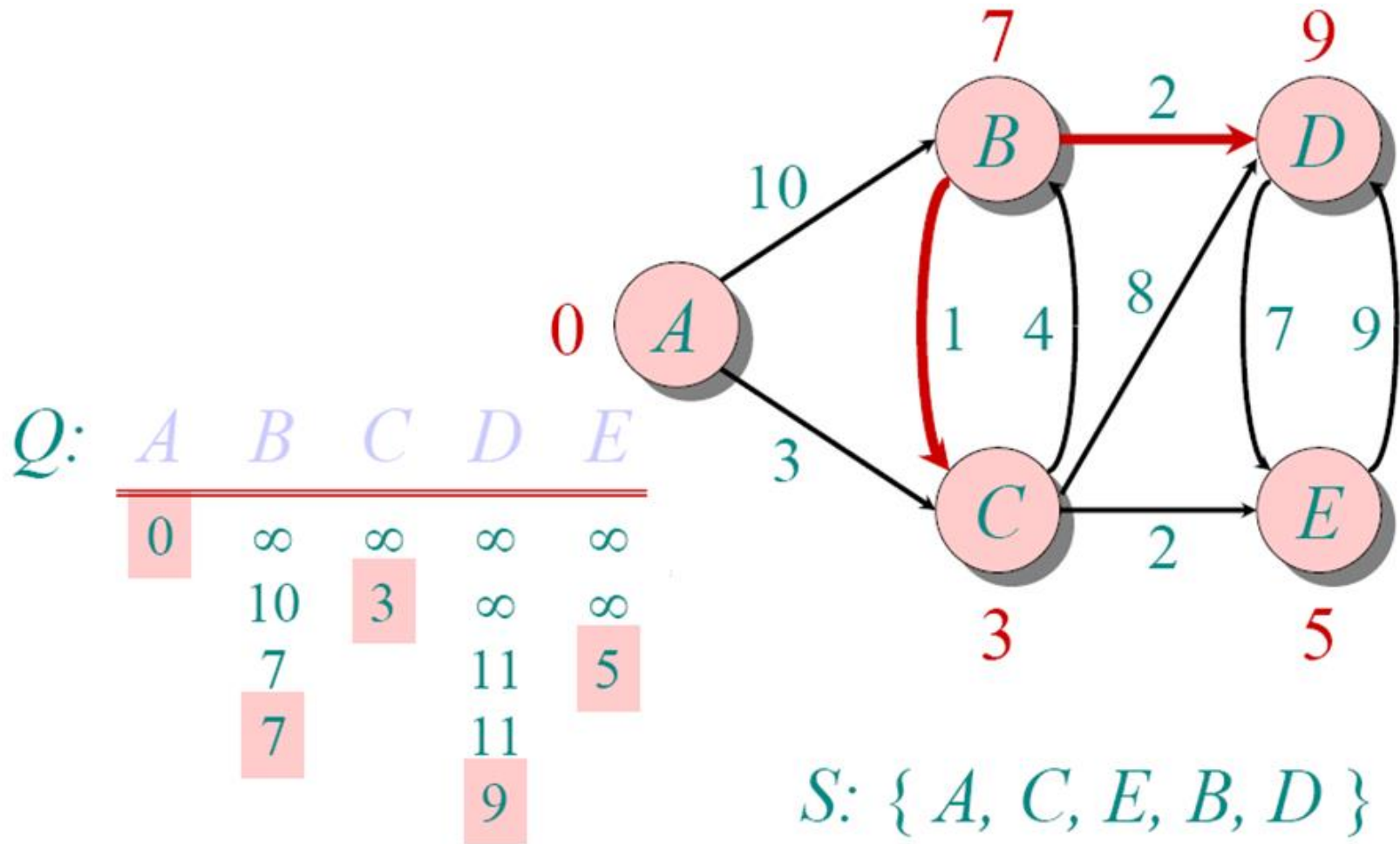
A	B	C	D	E
0	$\infty$	$\infty$	$\infty$	$\infty$
	10	3	$\infty$	$\infty$
	7		11	5
	7		11	

S: { A, C, E, B }

# Dijkstra Animated Example



# Dijkstra Animated Example



# Implementations and Running Times

- วิธีที่ง่ายที่สุดในการ implement คือ เก็บโหนดไว้ใน array หรือ linked list ซึ่งวิธีนี้ใช้เวลาในการทำงานเป็น  $O(|V|^2 + |E|)$
- ใน sparse graphs หรือ กราฟ ที่มีเส้นเชื่อมน้อยๆ โหนดเยอะๆ เราสามารถ implement การเก็บกราฟอย่างมีประสิทธิภาพไว้ใน adjacency list โดยใช้ binary heap หรือ priority queue ทำให้ได้เวลาในการทำงานเป็น  $O((|E|+|V|) \log |V|)$



# Dijkstra's Algorithm - Why It Works

- เพื่อให้เข้าใจว่าทำไมมันทำงานได้ เราจะดูตัวอย่างก่อนหน้านี้อีกรอบ อย่างไรก็ตาม เราต้องใช้คุณสมบัติต่อไปนี้

- **Lemma 1:** Triangle inequality

If  $\delta(u,v)$  is the shortest path length between  $u$  and  $v$ ,

$$\delta(u,v) \leq \delta(u,x) + \delta(x,v)$$

- **Lemma 2:**

The subpath of any shortest path is itself a shortest path.

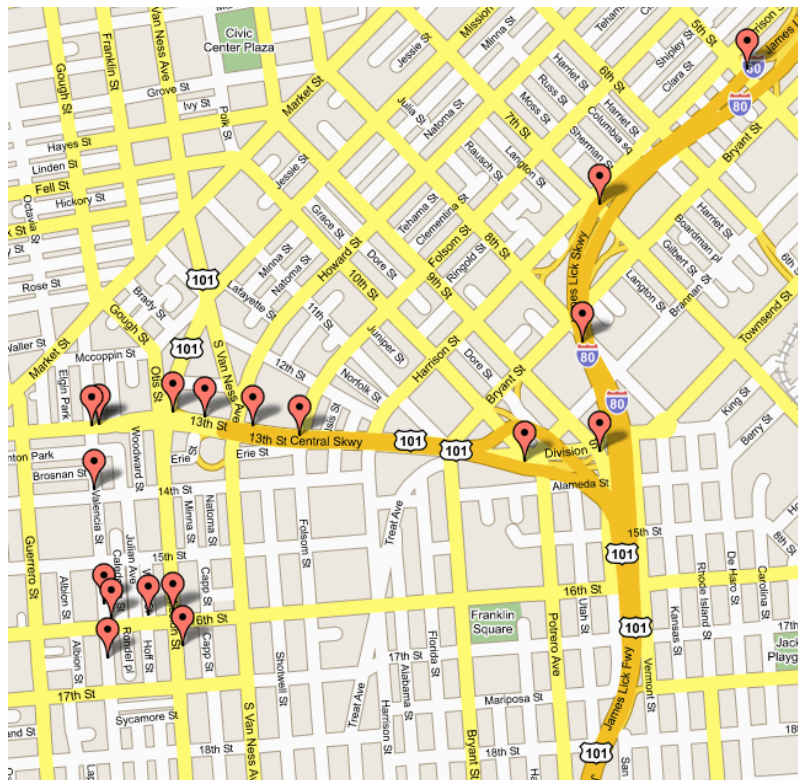
- หัวใจคือการเข้าใจว่าทำไมเราสามารถอ้างได้ว่าไม่ว่าเวลาใดที่เราเพิ่มโหนดใหม่เข้ามาใน  $S$  เราสามารถบอกได้ว่าเรารู้ shortest path ถึงมัน

- ย้อนกลับไปที ตัวอย่าง

# Applications of Dijkstra's Algorithm

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

From Computer Desktop Encyclopedia  
© 1998 The Computer Language Co. Inc.



## Router A Routing Table

To go to network:	Route via port #:
10.0.0.0	1
20.0.0.0	2
30.0.0.0	3
40.0.0.0	1

