

# Shortest Paths Algorithms Exercises

---

Lectures 5–6, Spring 2026

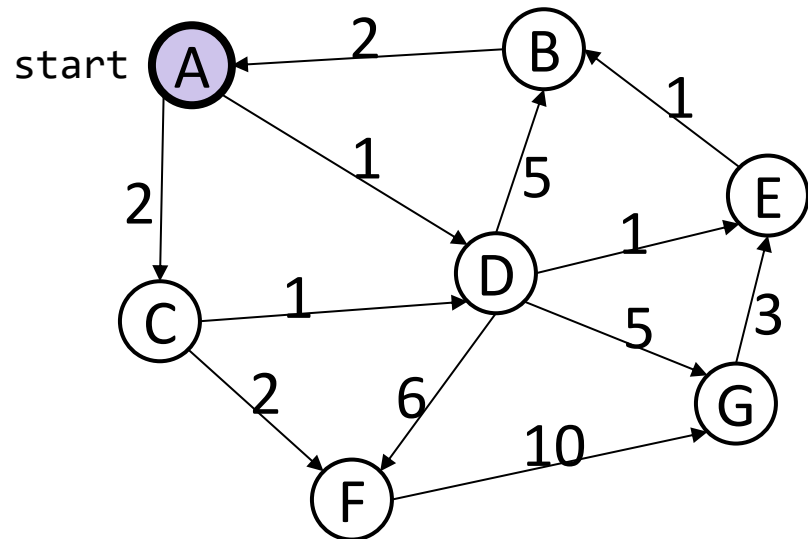
BFS

Dijkstra's Algorithm

Bellman-Ford Algorithm

# Q. Dijkstra's Algorithm

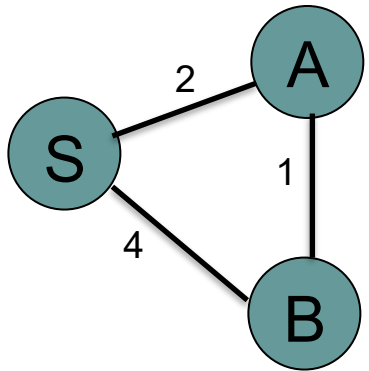
*Exam question:* Given this directed graph, run Dijkstra's Algo to find shortest paths starting from **source node A**. Give the node visit order, and fill in this table of SN (Shortest Distance) and PN (Previous Node), crossing out old SD and PN as you find a shortcut path with smaller SD



Visit Order

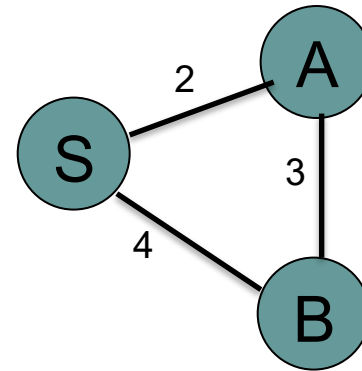
Node	SD	PN
A		
B		
C		
D		
E		
F		
G		

# Q. Dijkstra's Algorithm (Source Node S)



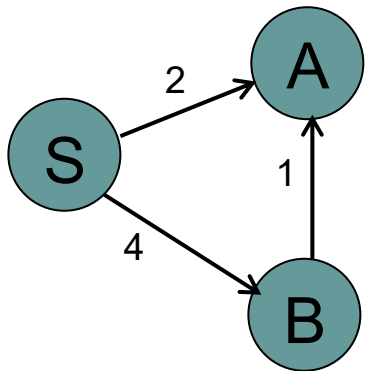
ANS

Node	SD	PN
S	0	/
A		
B		



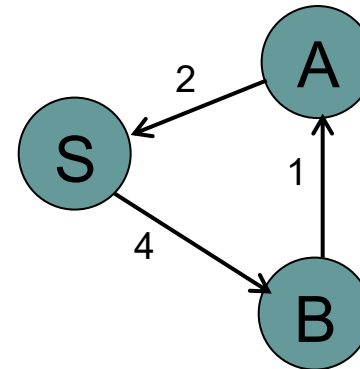
ANS

Node	SD	PN
S	0	/
A		
B		



ANS

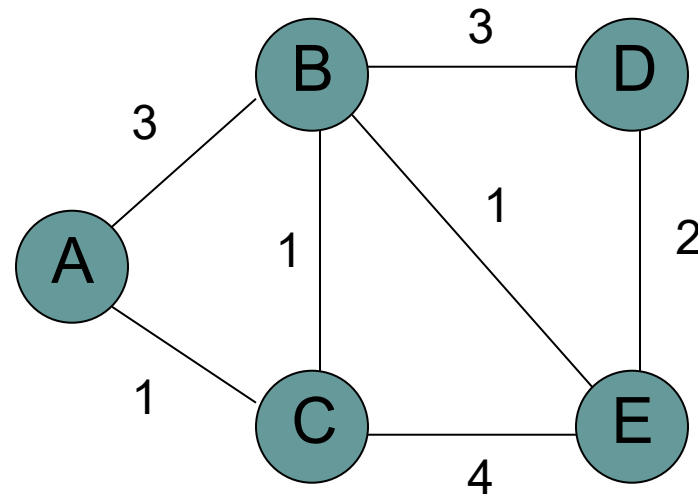
Node	SD	PN
S	0	/
A		
B		



ANS

Node	SD	PN
S	0	/
A		
B		

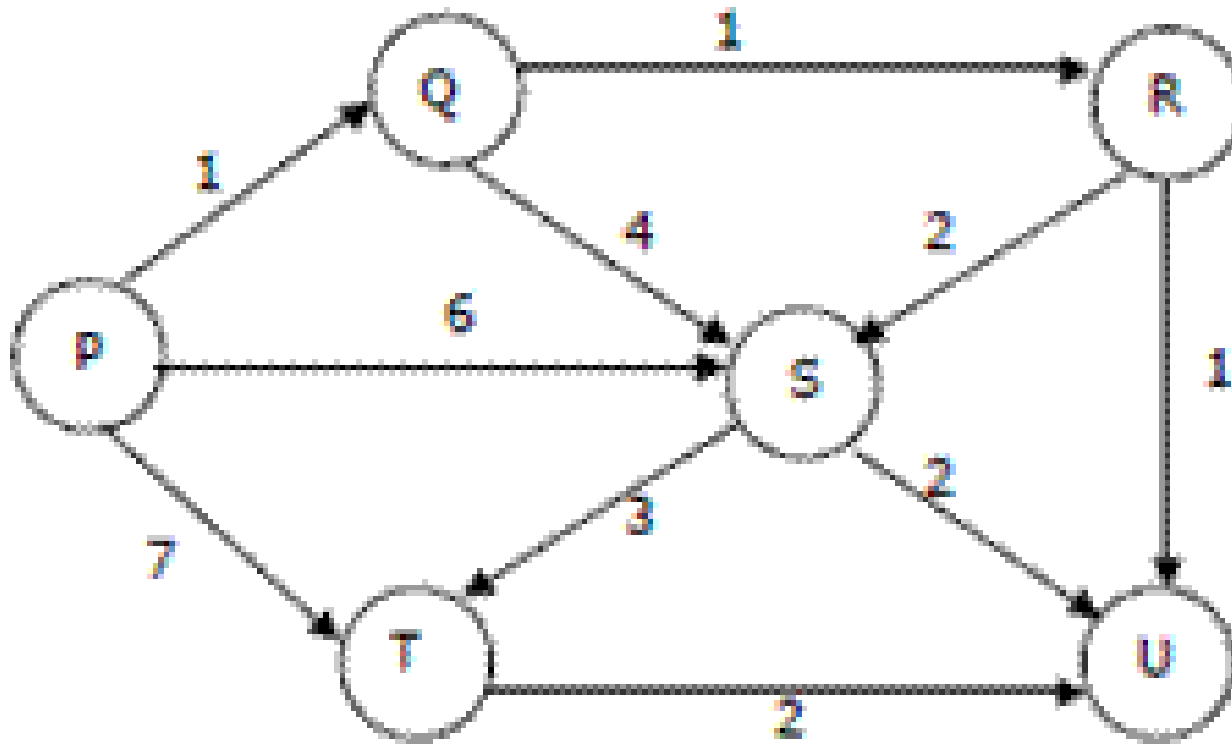
# Q. Dijkstra's Algorithm (Source Node A, Undirected Graph)



Visit Order

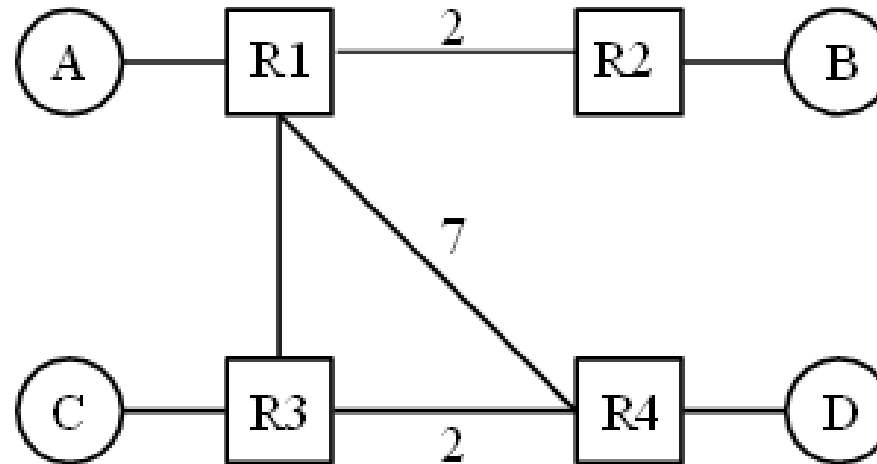
Node	SD	PN
A		
B		
C		
D		
E		

# Q. Dijkstra's Algorithm (Source Node P, Directed Graph)



# Q. Distance-Vector

Consider running the distance-vector protocol on the topology below. Unlabeled links have cost 1. Fill in the routing tables based on Distance-Vector algorithm. (refer to the handout for details.)



R1's table

Dest.	Hop, Dist.
A	Direct, 1

R2's table

Dest.	Hop, Dist.
B	Direct, 1

R3's table

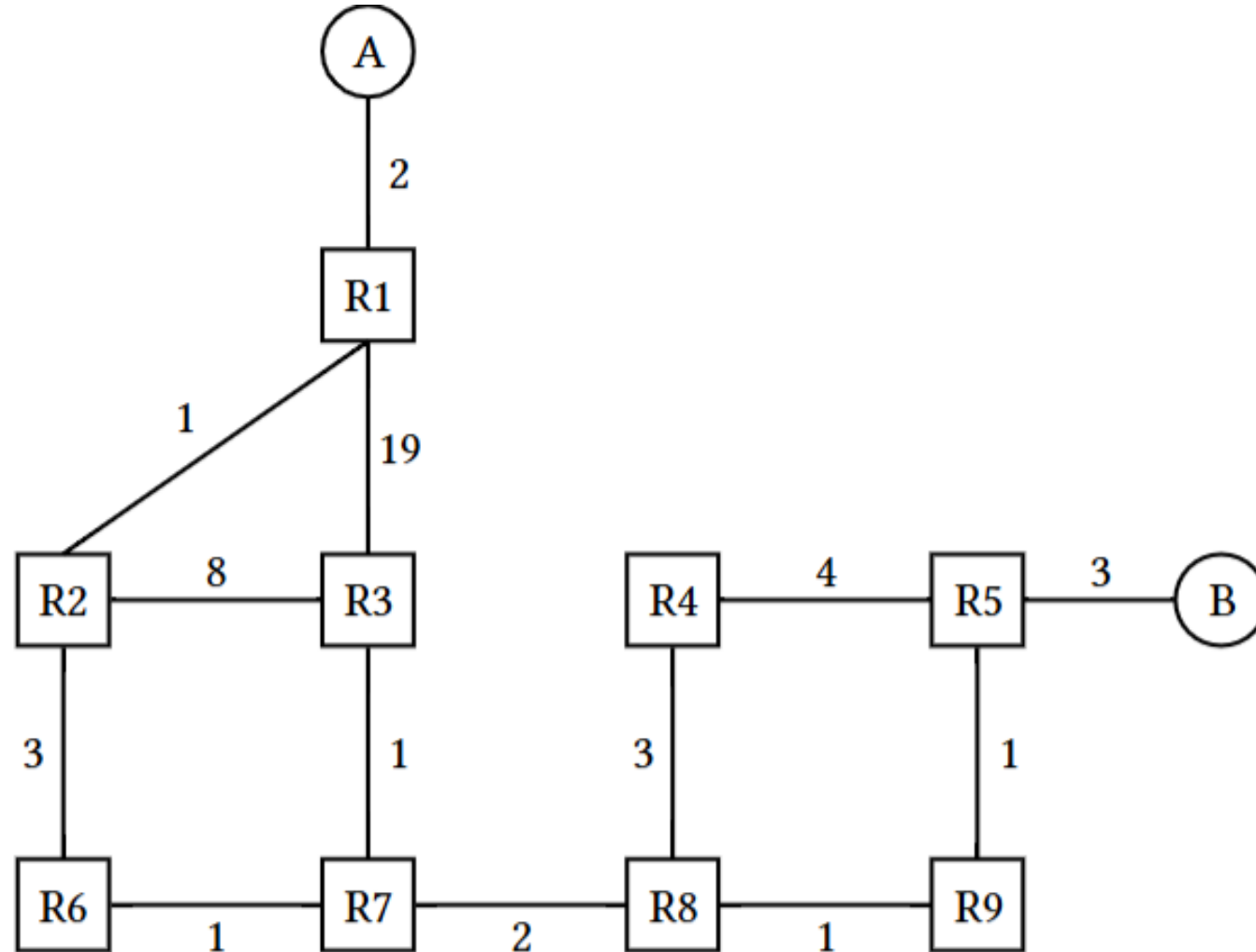
Dest.	Hop, Dist.
C	Direct, 1

R4's table

Dest.	Hop, Dist.
D	Direct, 1

# Q. Link-State Routing

Refer to the handout for details.



# Q. More L3 Link State

Refer to the handout for details.

