

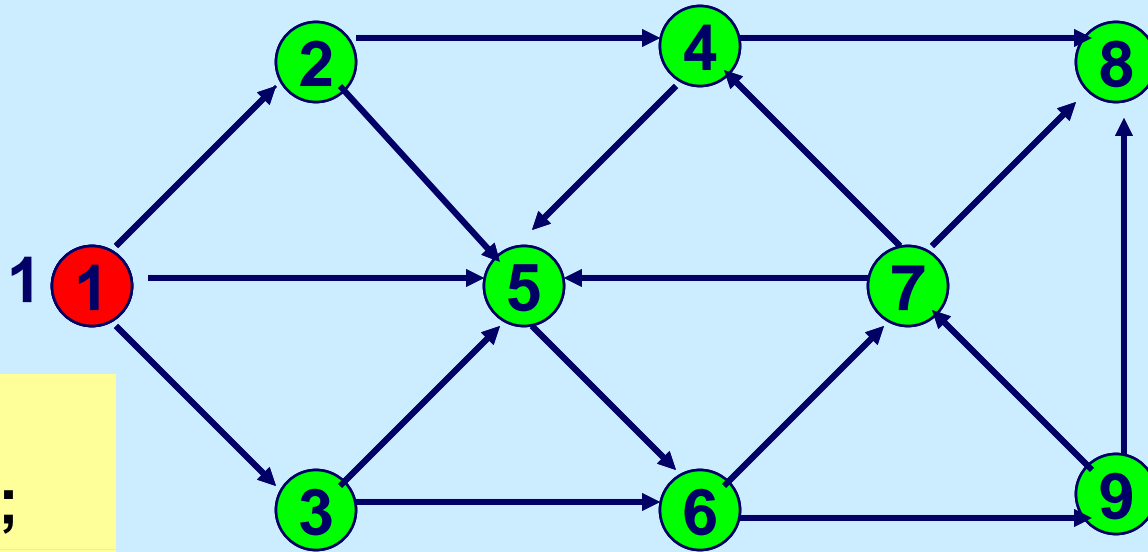
# **Network Optimization**

## **Breadth First Search**

---

Breadth first search animation

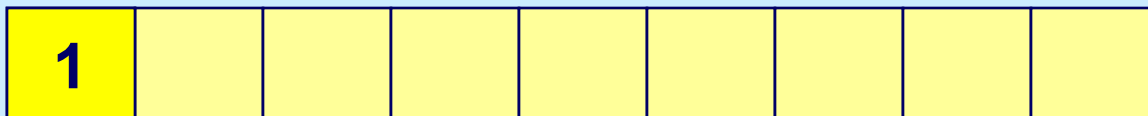
# Initialize



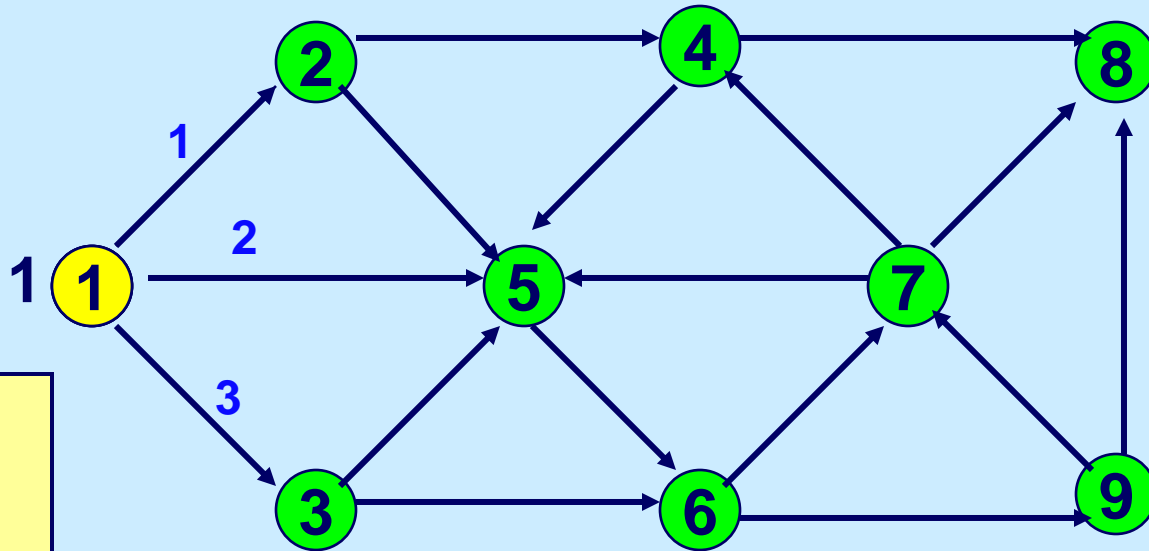
Unmark all nodes in N;

Mark node 1 and add it to LIST.

LIST



# Select node 1



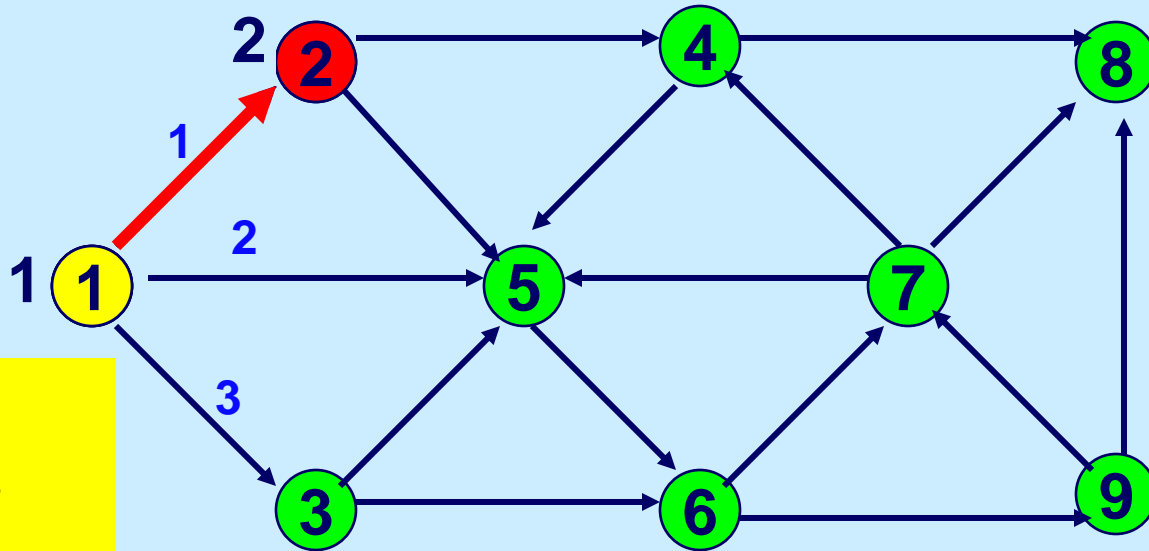
Select the first node in LIST.

Arc numbers for arcs in  $A(i)$  refer to the order in which arcs of  $A(i)$  are scanned.

LIST

1									
---	--	--	--	--	--	--	--	--	--

# Scan arcs (1,2)

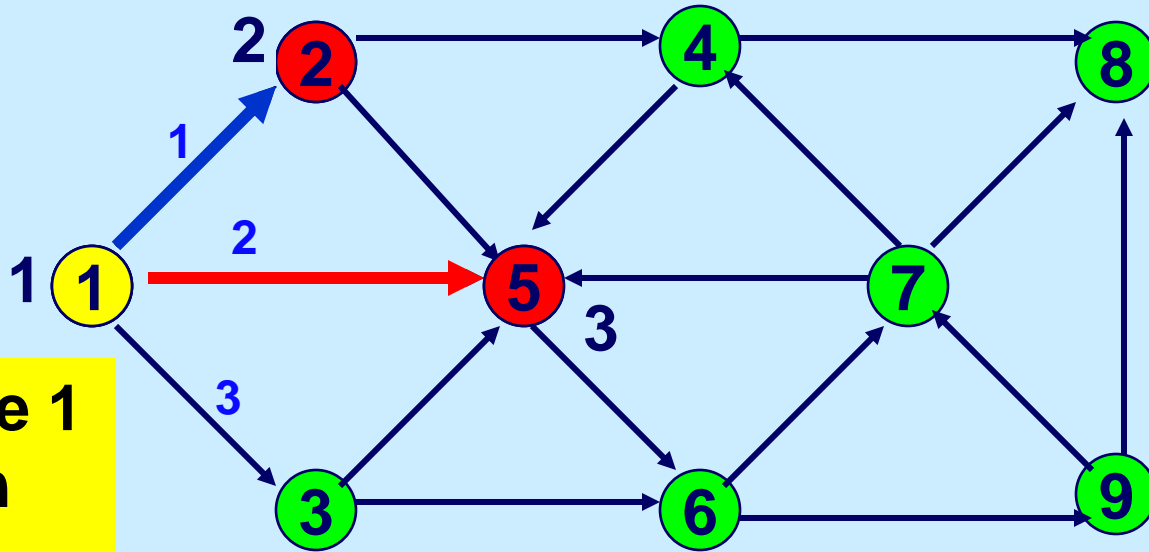


Find an  
admissible  
arc

LIST

1	2								
---	---	--	--	--	--	--	--	--	--

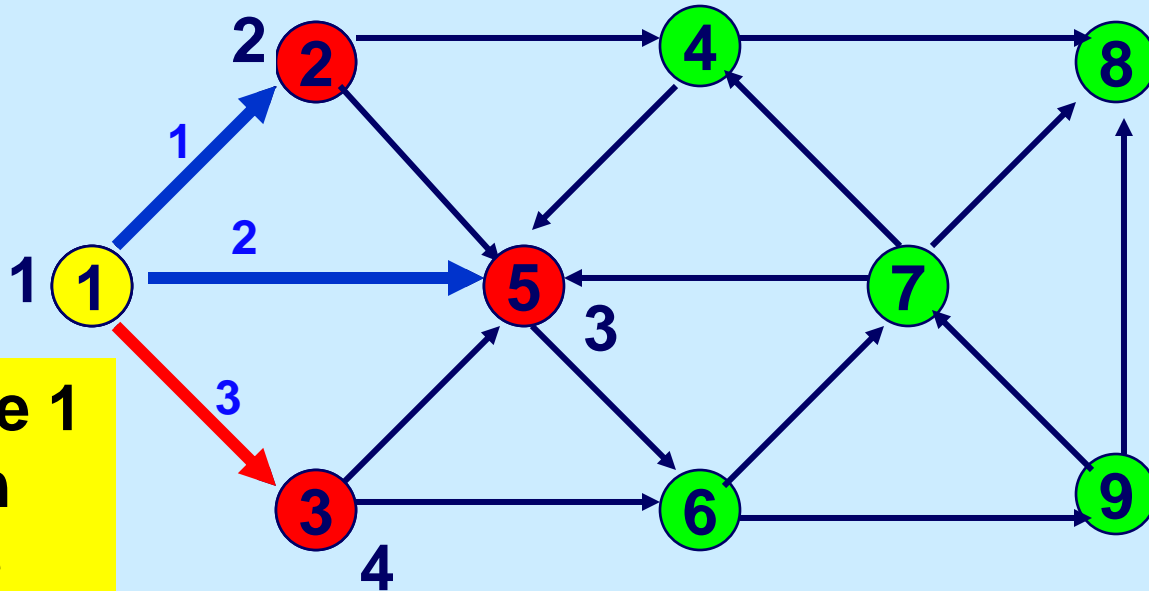
# Scan arc (1,5)



Select node 1 and find an admissible arc.

LIST	1	2	5						
------	---	---	---	--	--	--	--	--	--

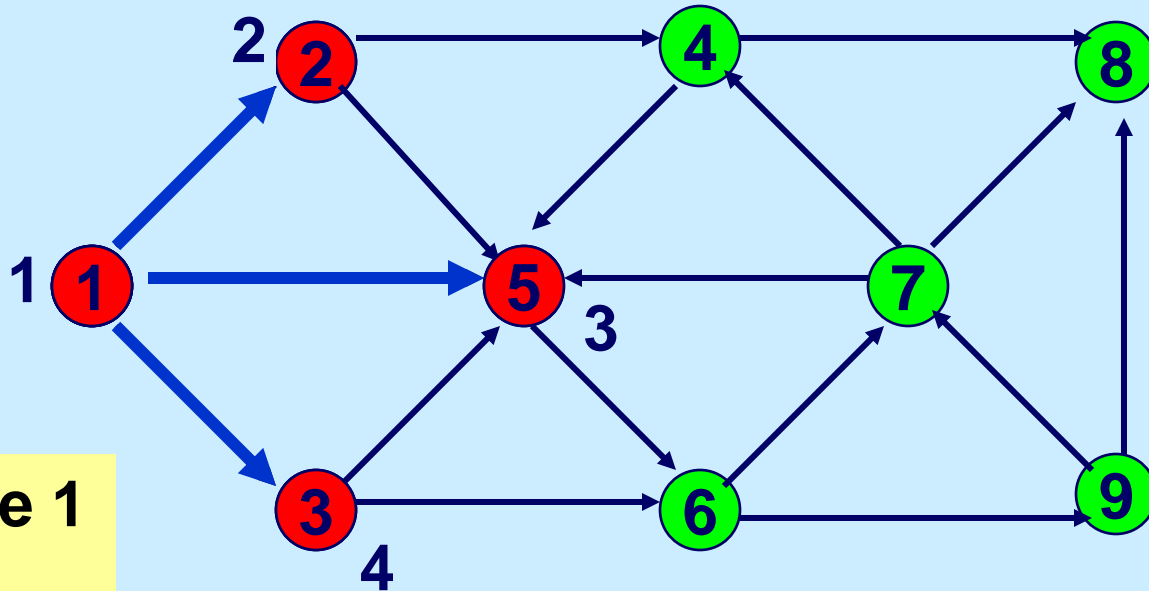
# Scan arc (1,3)



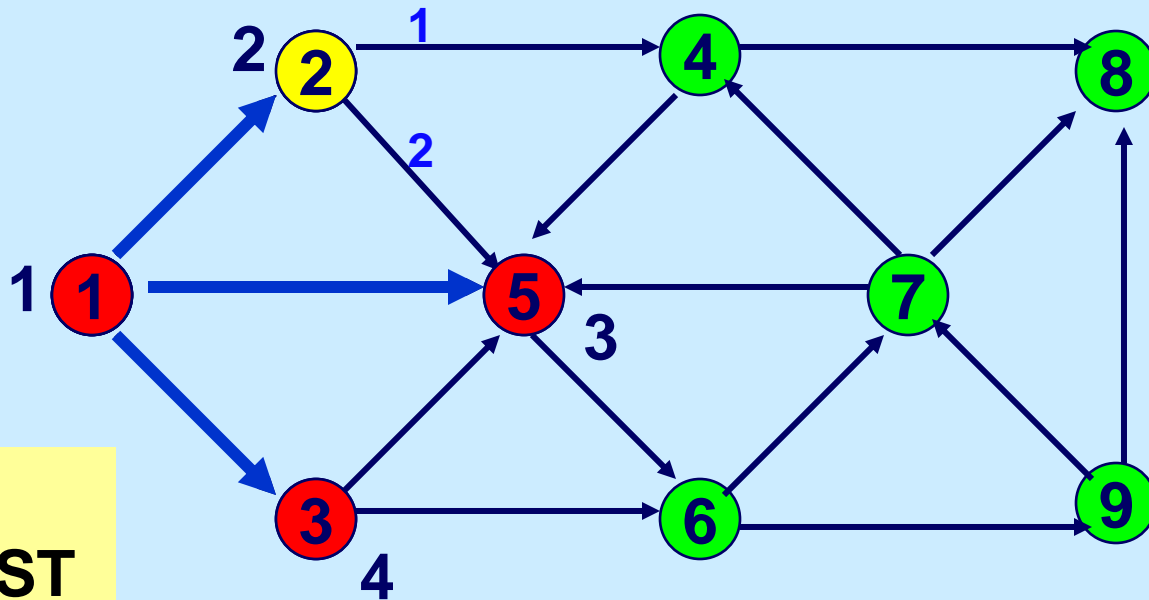
Select node 1 and find an admissible arc.

LIST	1	2	5	3					
------	---	---	---	---	--	--	--	--	--

# Delete Node 1 from LIST



# Select Node 2



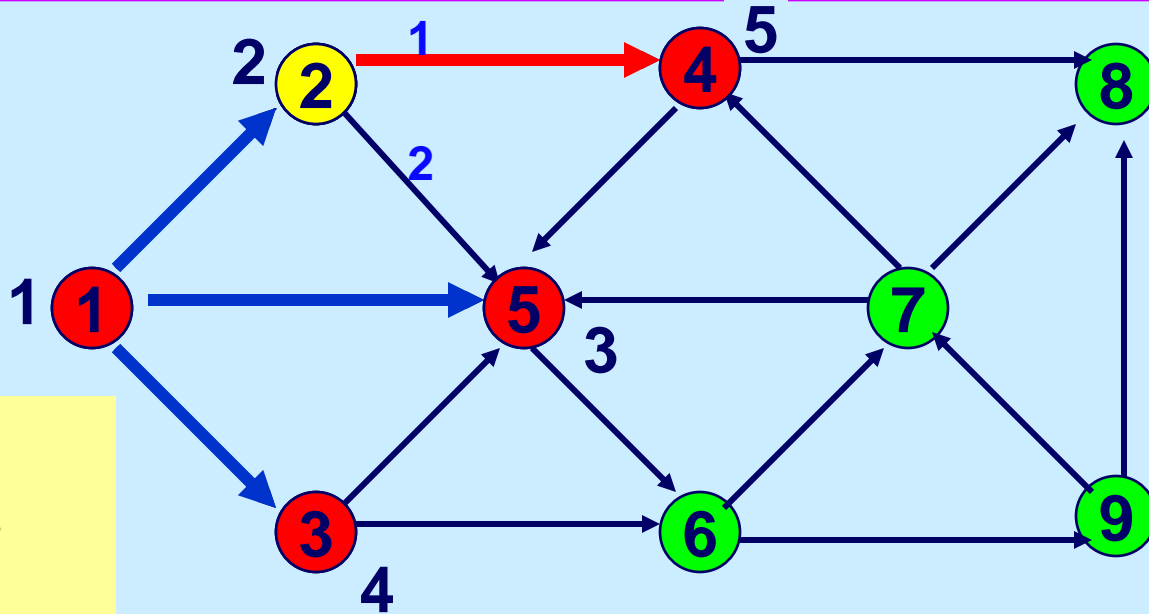
Select first node on LIST

LIST

2	5	3						
---	---	---	--	--	--	--	--	--



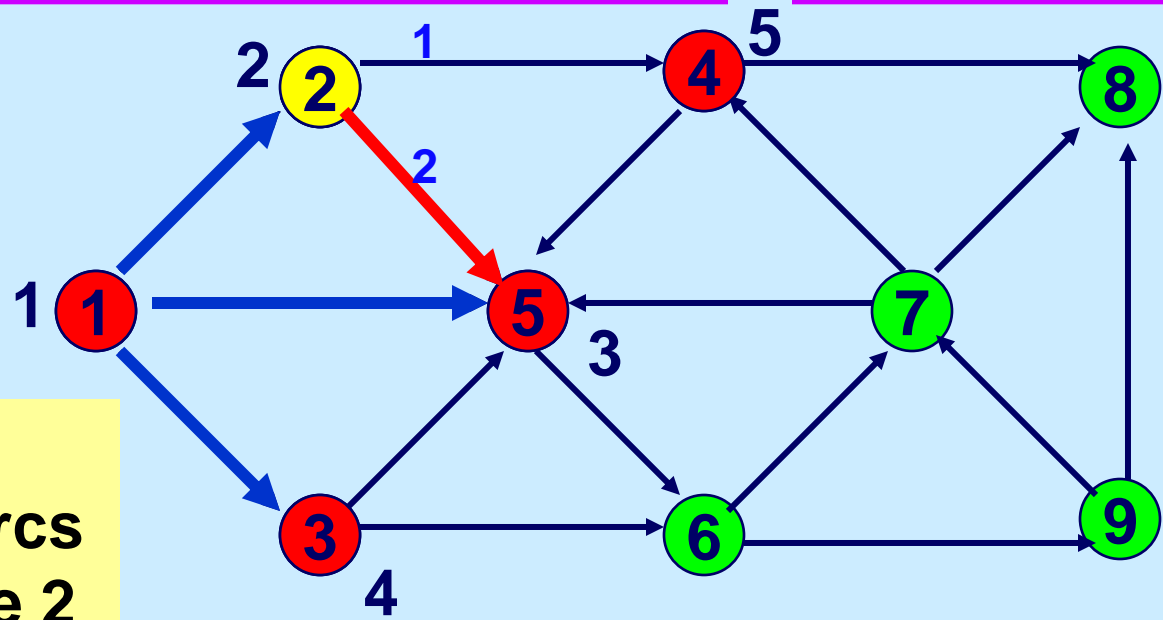
# Scan arc (2, 4)



LIST

2	5	3	4				
---	---	---	---	--	--	--	--

# Scan arc (2, 5)

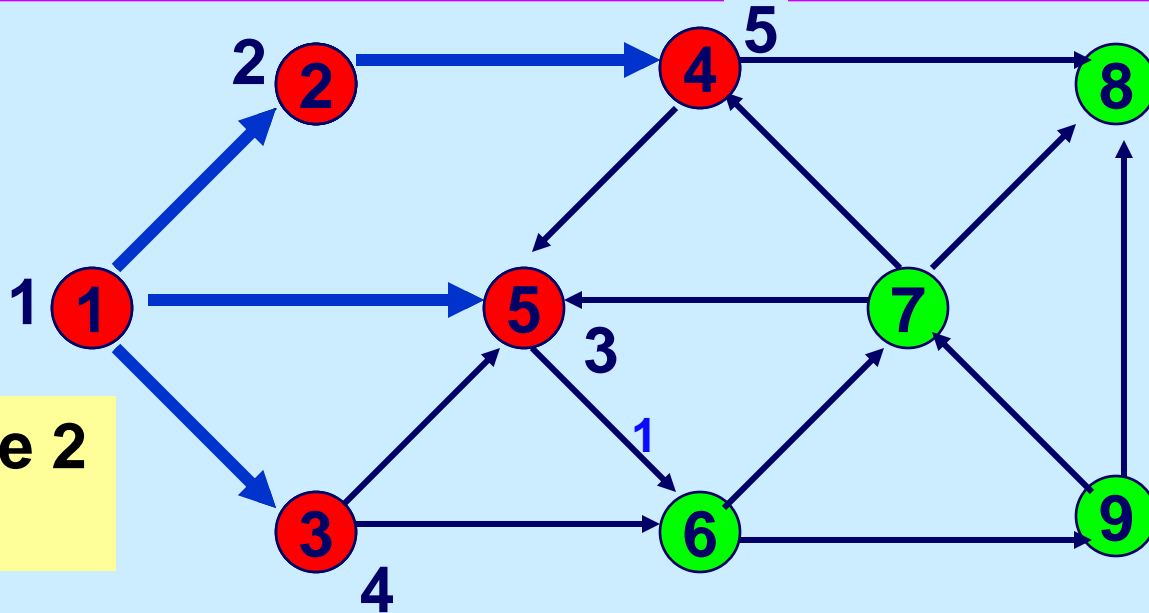


Ignore any scanned arcs out of node 2 that are not admissible.

LIST

2	5	3	4				
---	---	---	---	--	--	--	--

# Delete node 2

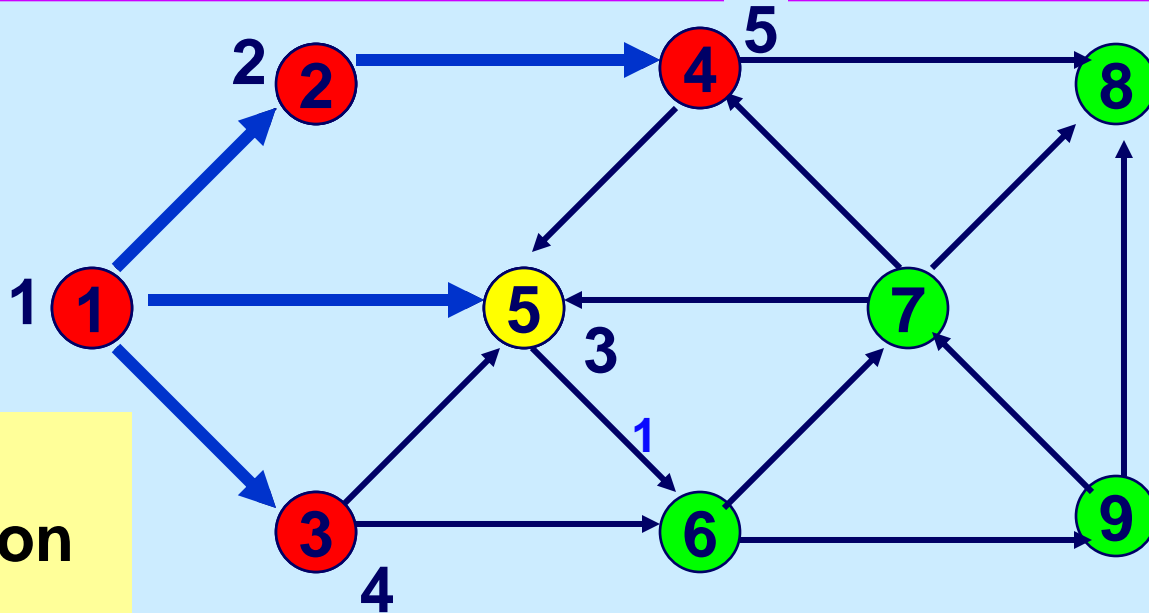


Delete node 2  
from LIST

LIST

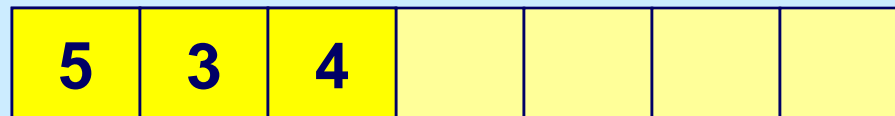
2	5	3	4				
---	---	---	---	--	--	--	--

# Select node 5

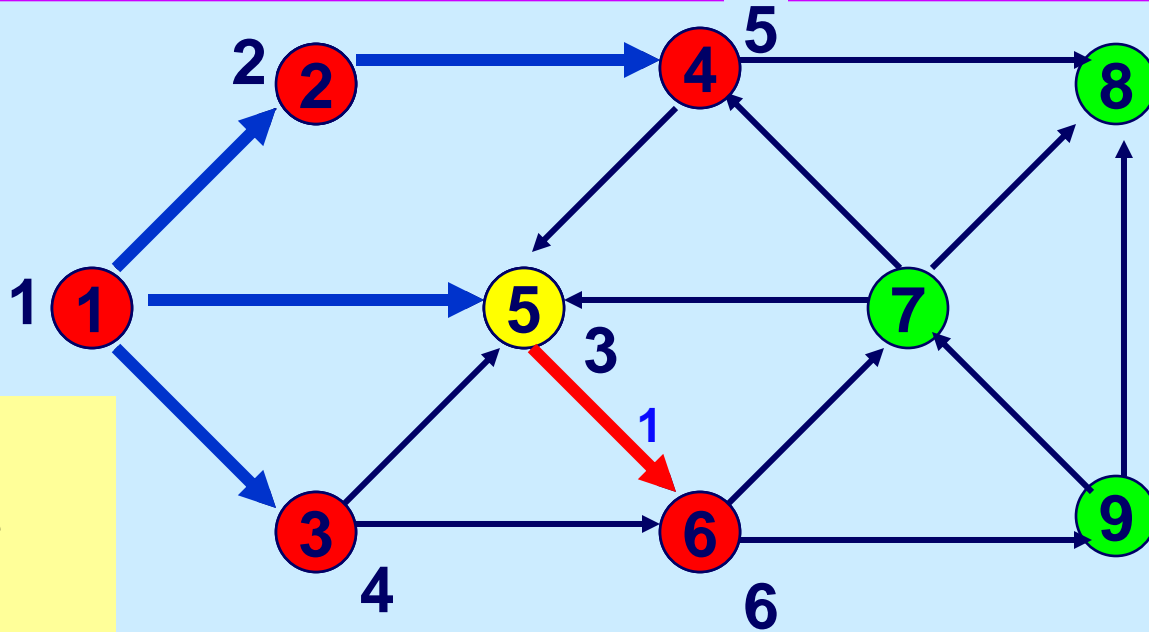


Select the first node on LIST

LIST



# Scan arc (5,6)

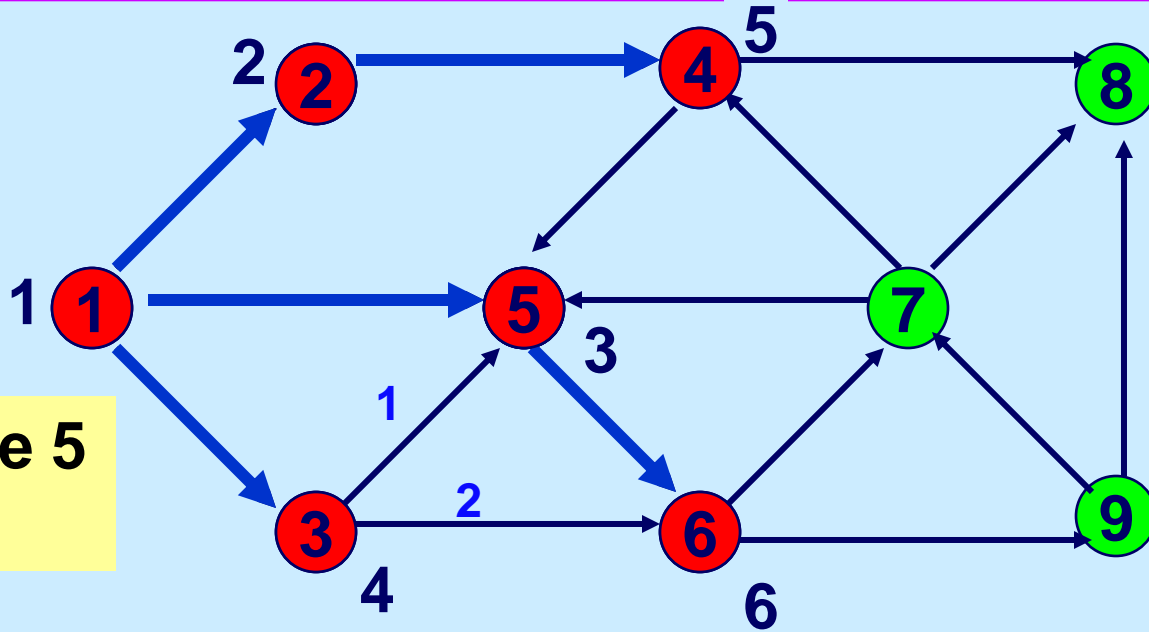


Find an  
admissible  
arc

LIST

5	3	4	6			
---	---	---	---	--	--	--

# Delete node 5

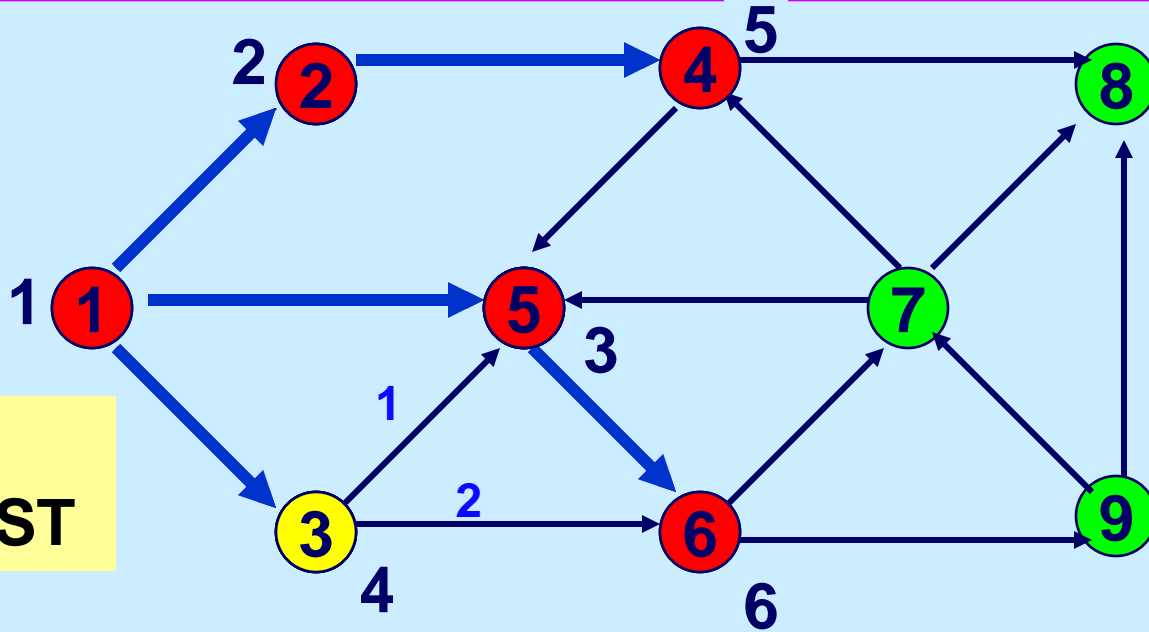


Delete node 5 from LIST

LIST

5	3	4	6			
---	---	---	---	--	--	--

# Select node 3

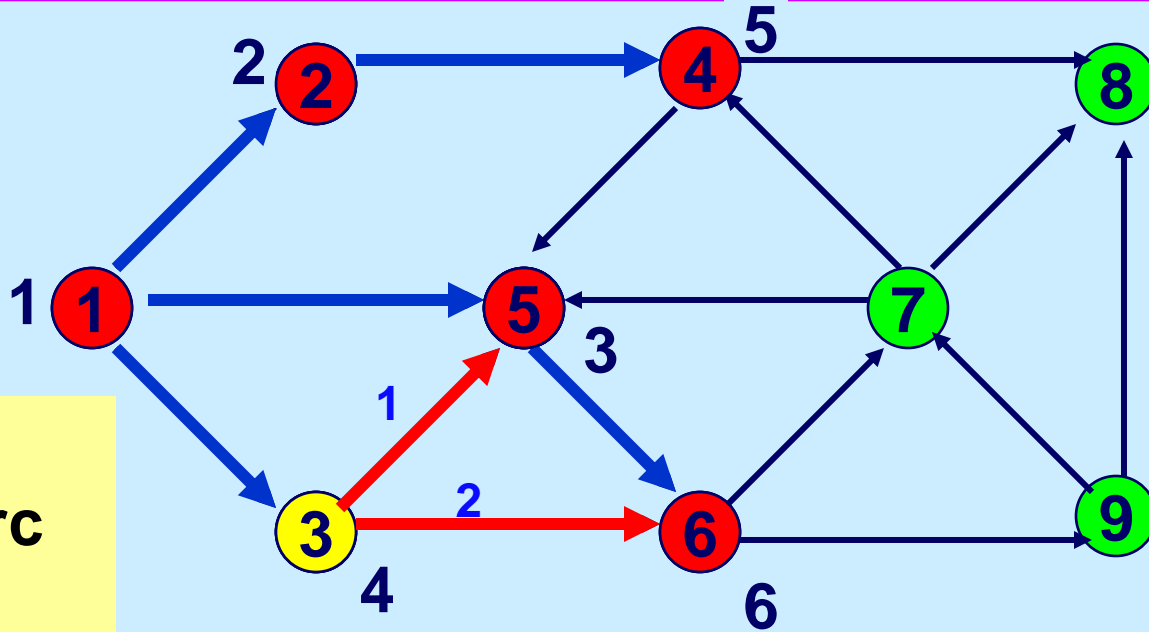


Select first node on LIST

LIST

3	4	6			
---	---	---	--	--	--

# Scan arcs out of node 3



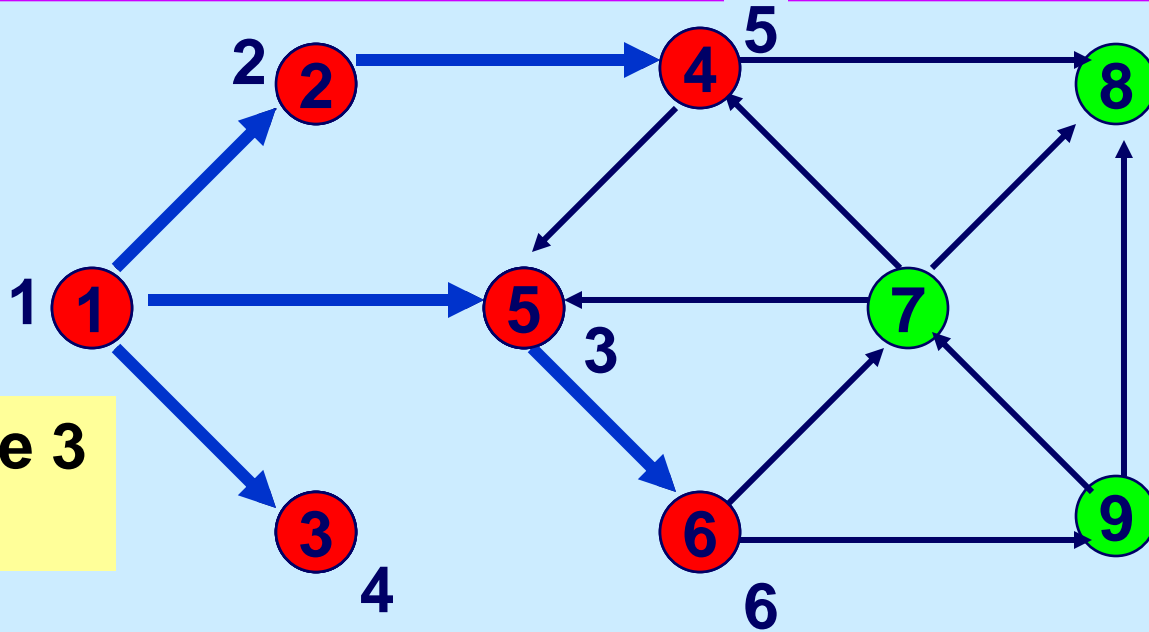
Ignore any scanned arc that is not admissible.

LIST

3	4	6			
---	---	---	--	--	--



# Delete node 3

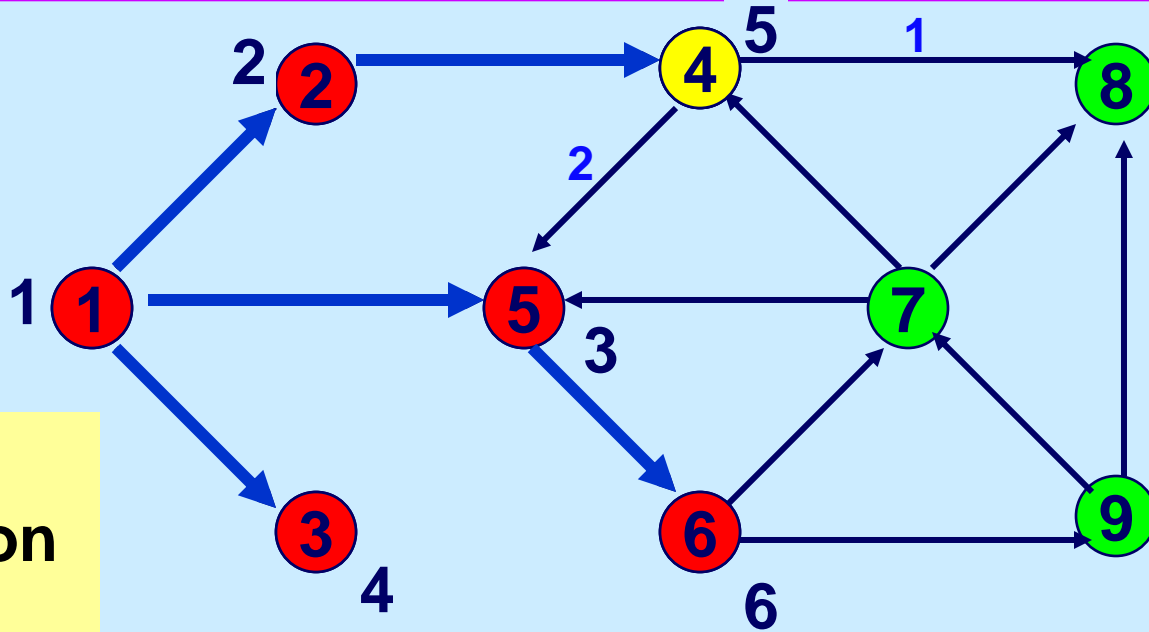


Delete node 3 from LIST

LIST

3	4	6			
---	---	---	--	--	--

# Select node 4

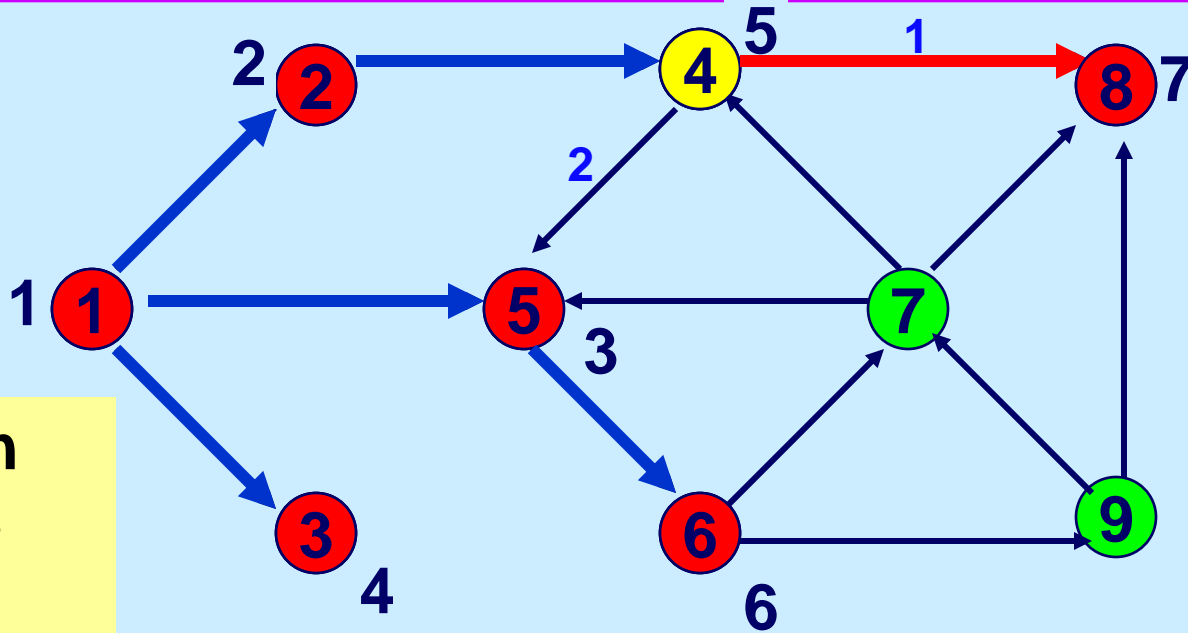


Select the first node on LIST

LIST

4	6			
---	---	--	--	--

# Scan arc (4, 8)

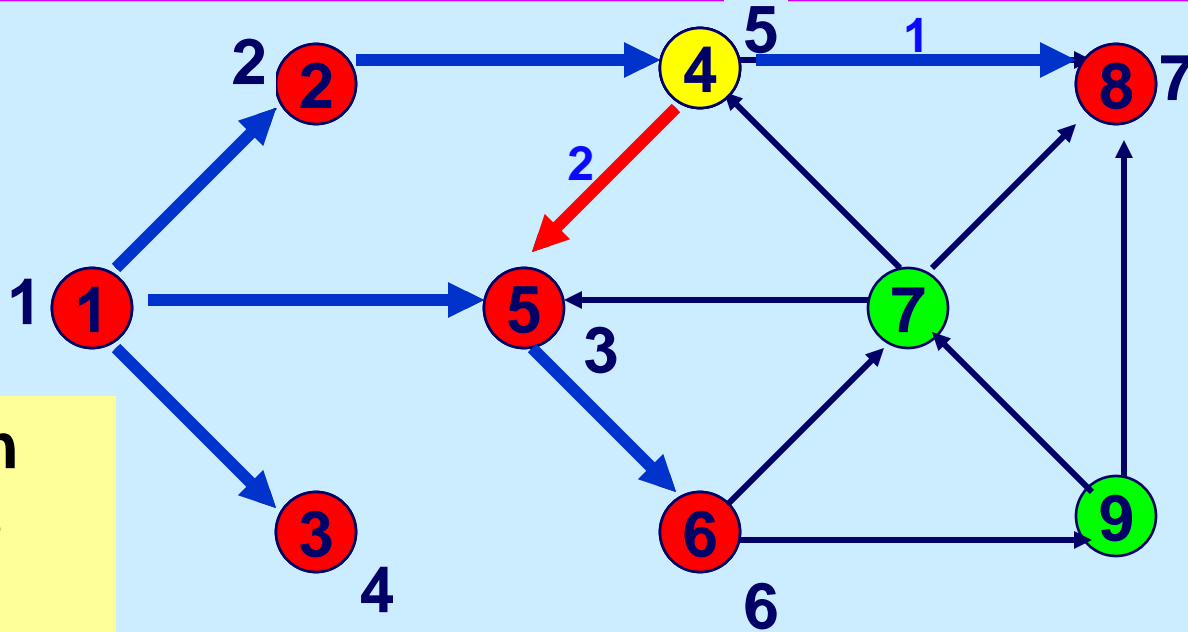


Look for an admissible arc.

LIST

4	6	8		
---	---	---	--	--

# Scan arc (4, 5)

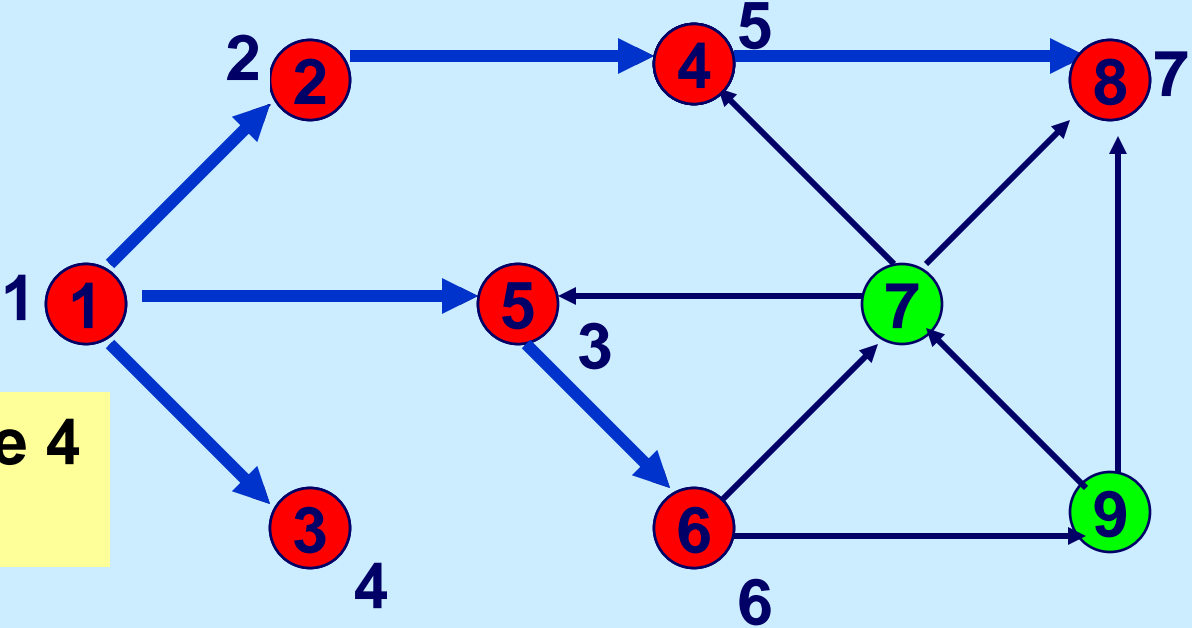


Look for an  
admissible  
arc

LIST

4	6	8		
---	---	---	--	--

# Delete Node 4

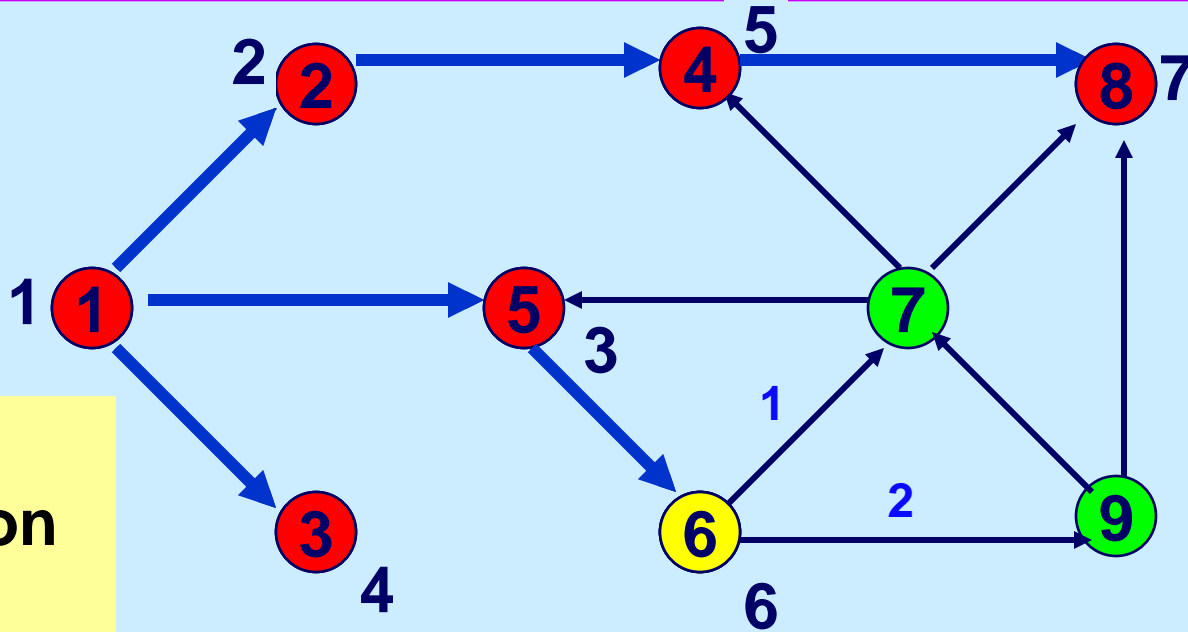


Delete node 4 from LIST

LIST

4	6	8		
---	---	---	--	--

# Select node 6

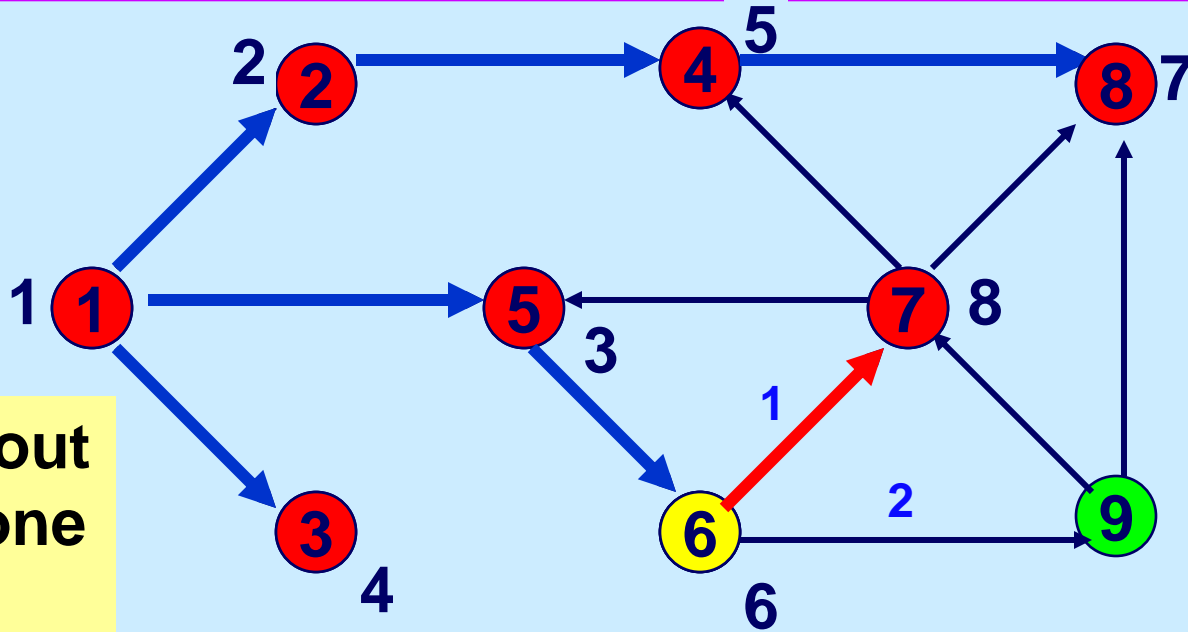


Select the first node on LIST

LIST

6	8		
---	---	--	--

# Scan arc (6,7)

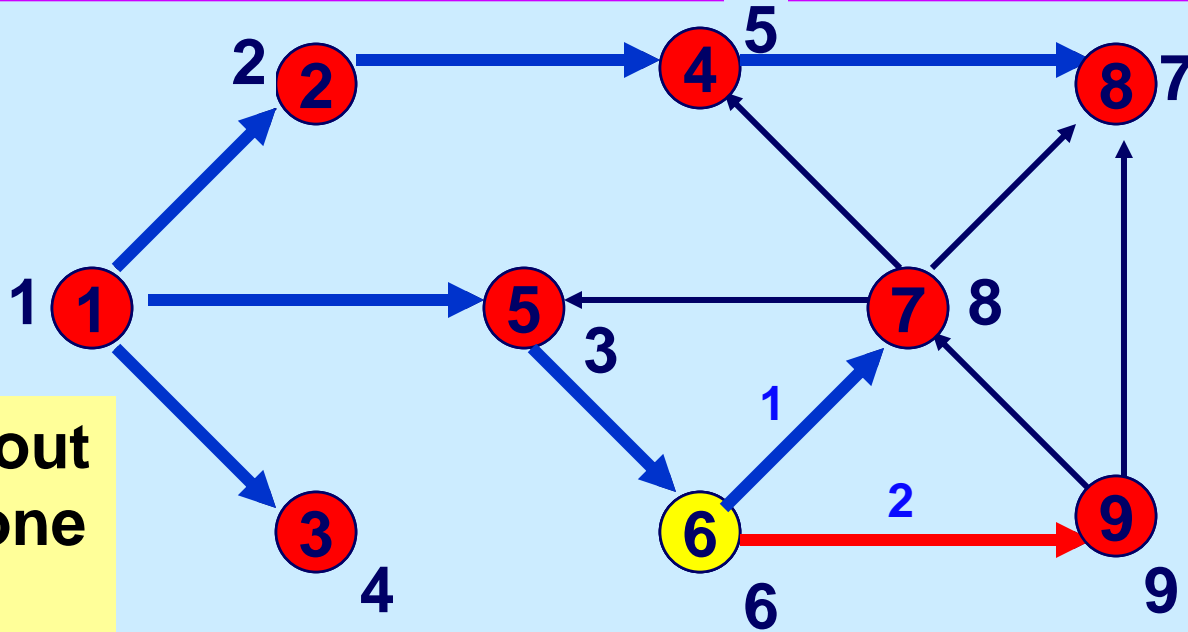


Scan arcs out of node 6 one at a time.

LIST

6	8	7	
---	---	---	--

# Scan arc (6,9)

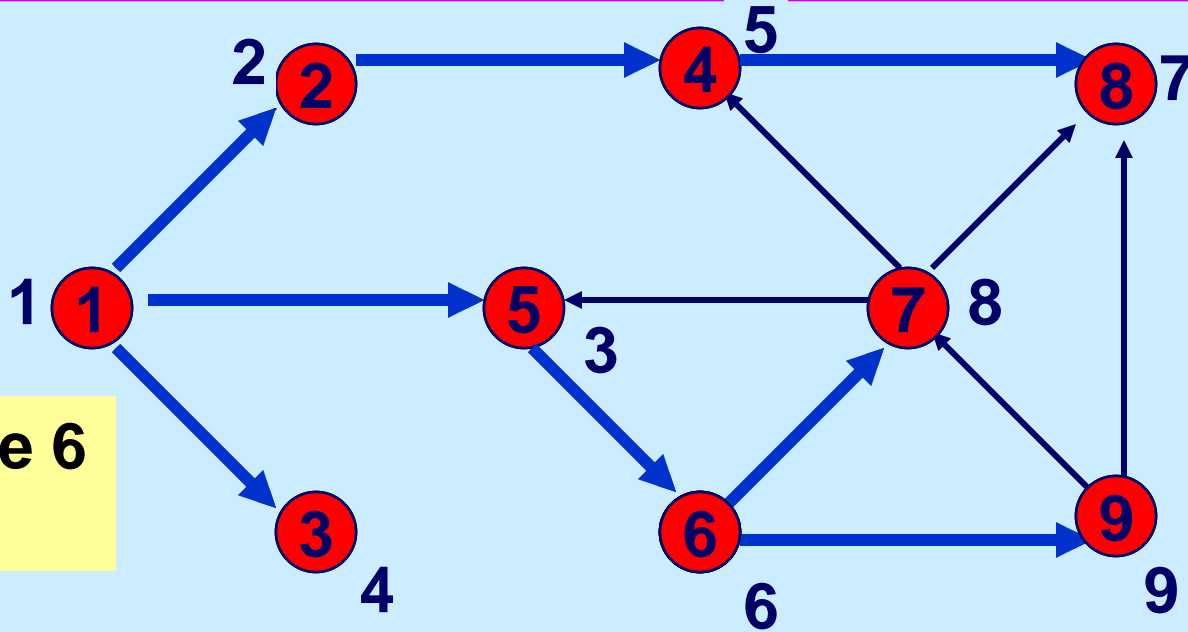


LIST





# Delete node 6

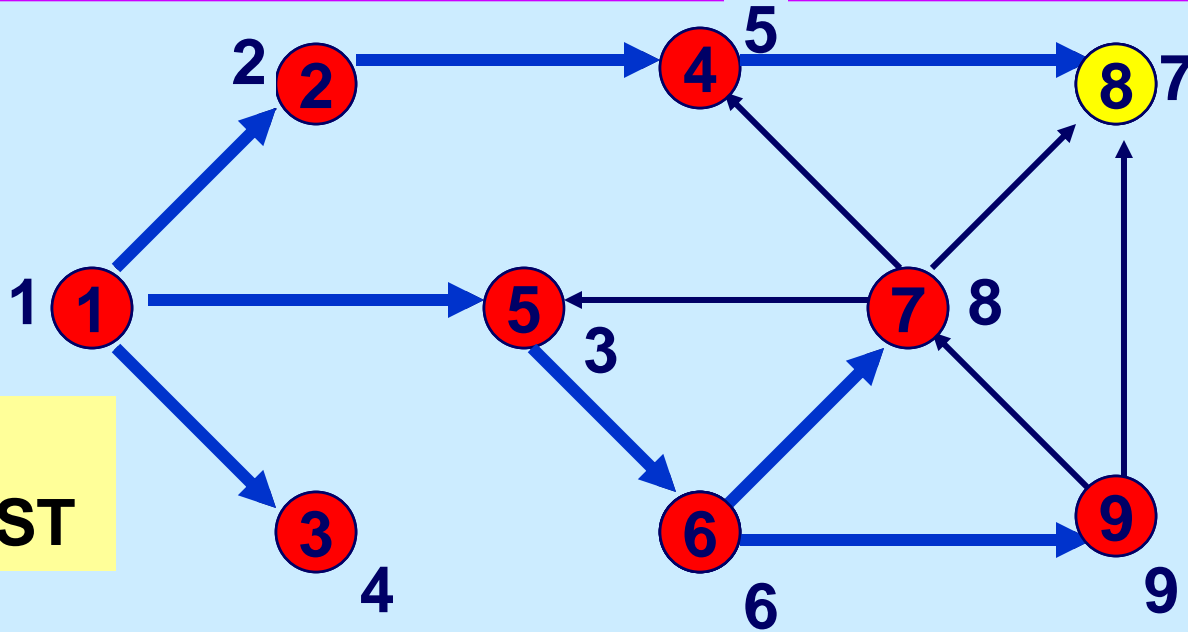


Delete node 6 from LIST

LIST

6	8	7	9
---	---	---	---

# Select node 8

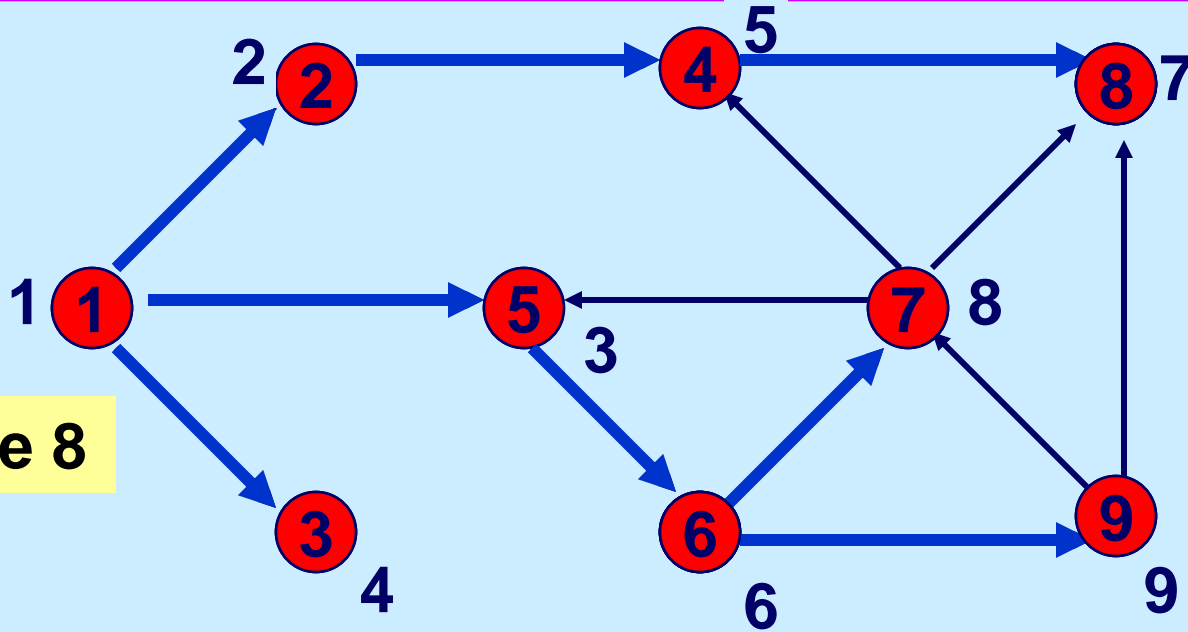


Select first node on LIST

LIST

8	7	9
---	---	---

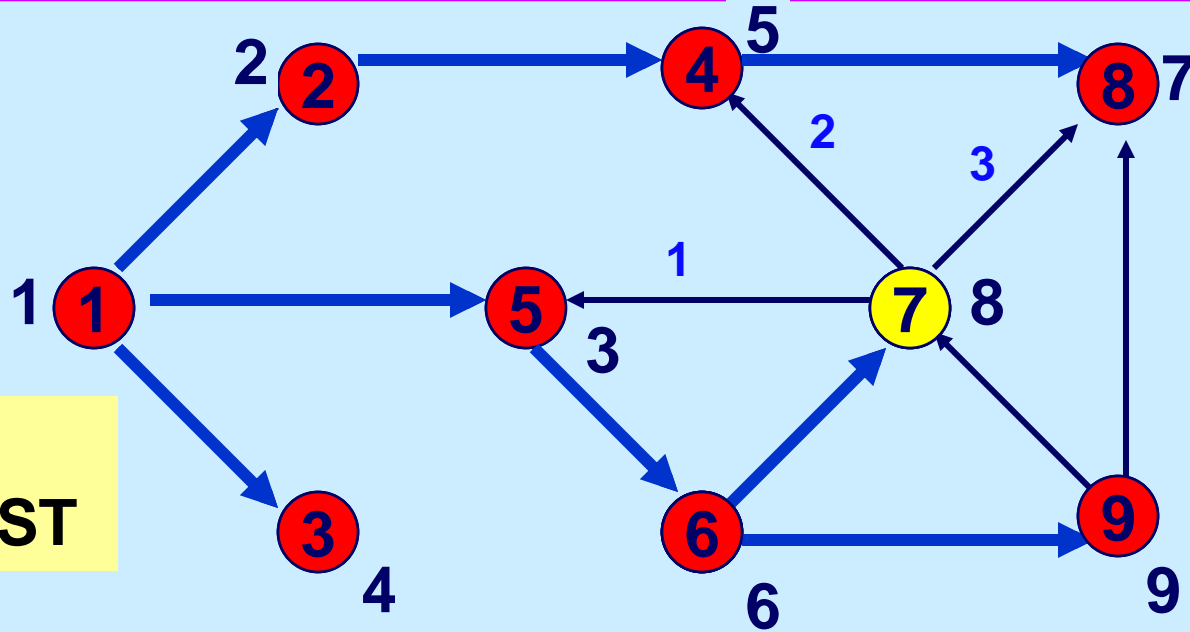
# Delete node 8



LIST

8	7	9
---	---	---

# Select node 7

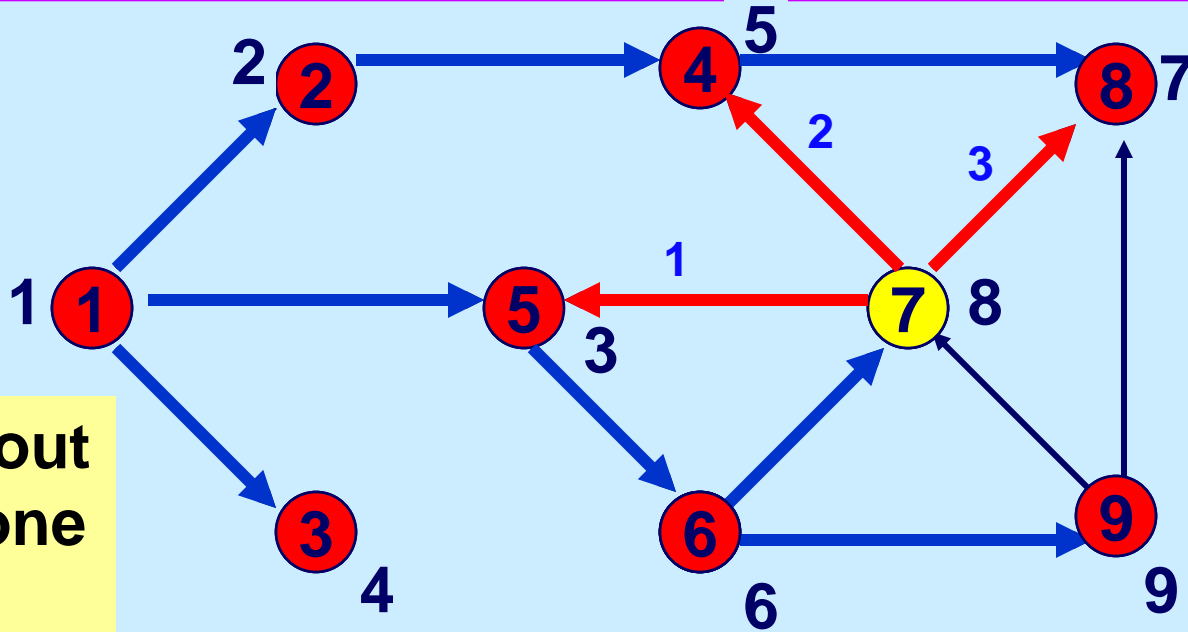


Select first node on LIST

LIST

7	9
---	---

# Scan arcs out of node 7

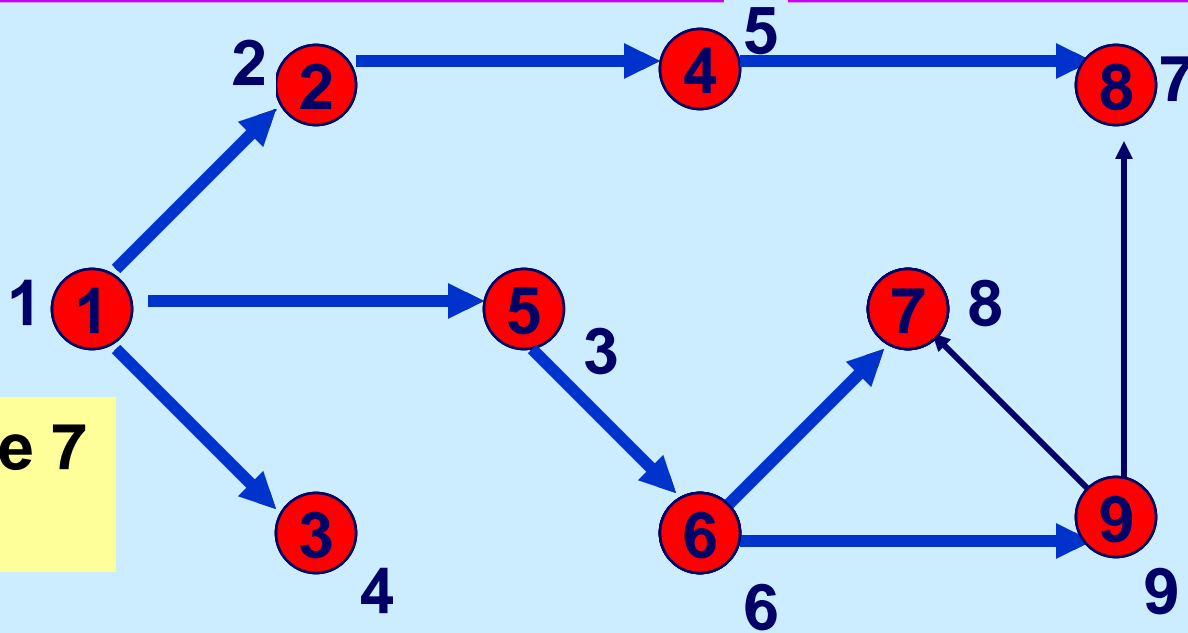


Scan arcs out of node 7 one at a time.

LIST

7	9
---	---

# Delete node 7

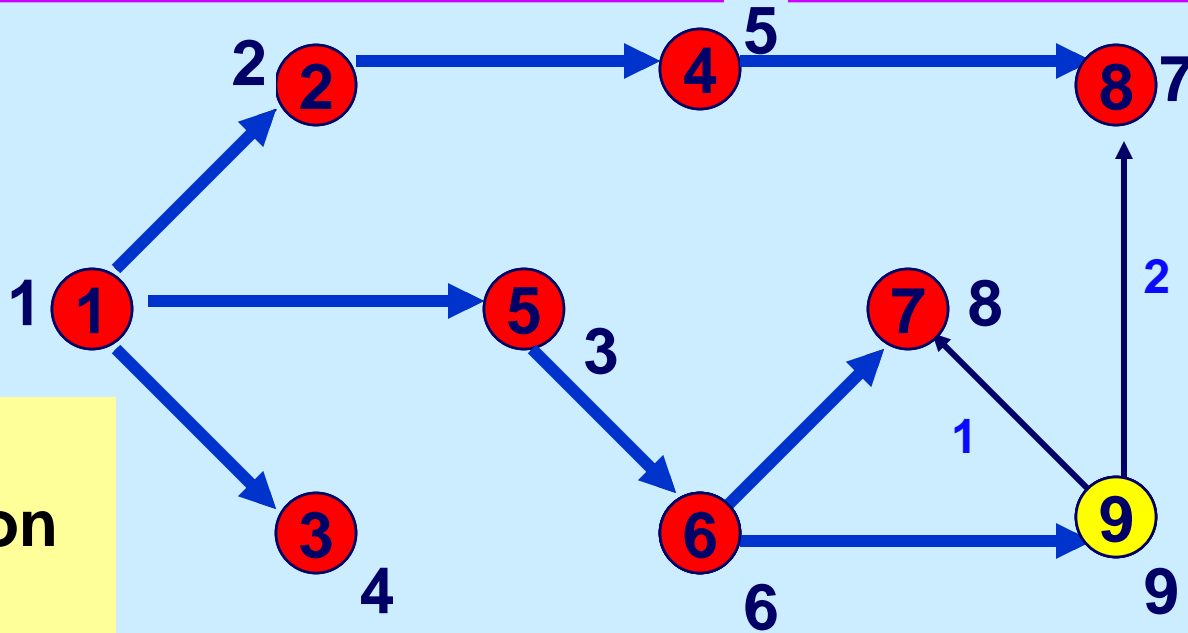


Delete node 7  
from LIST

LIST

7	9
---	---

# Select node 9

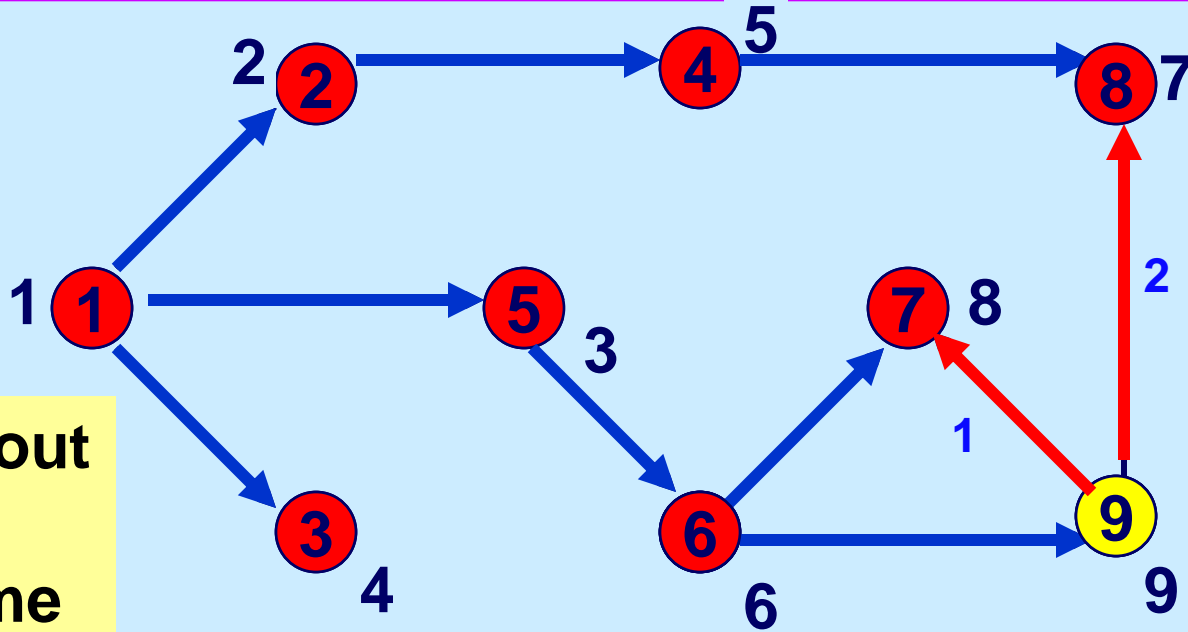


Select the first node on LIST

LIST

9

# Scan arcs out of node 9



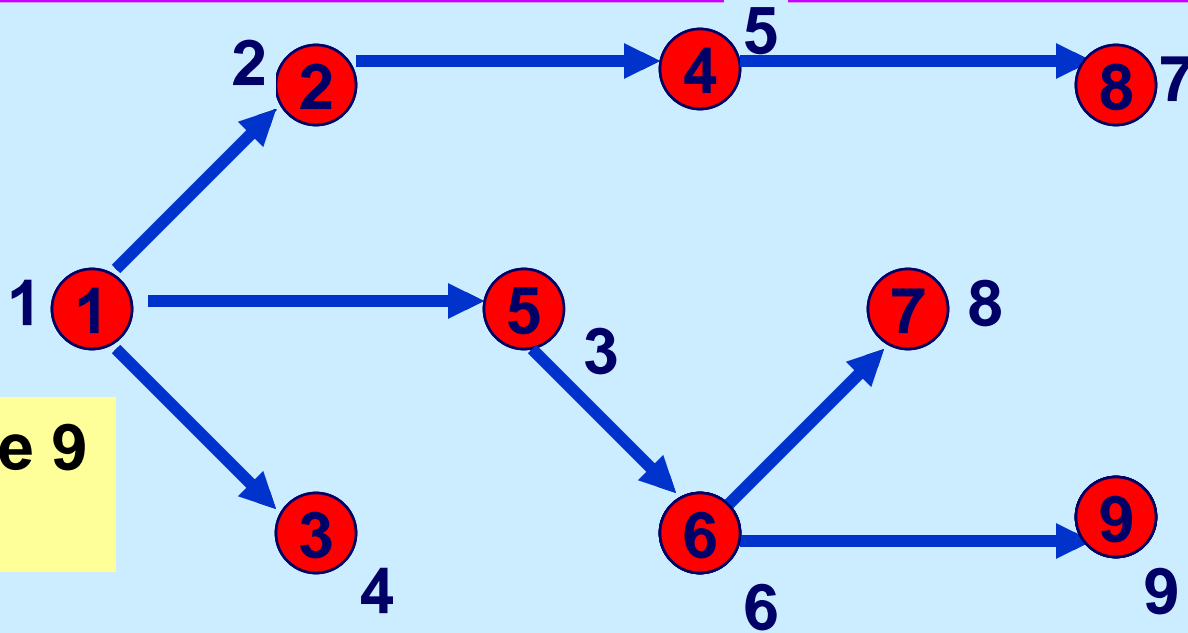
Scan arcs out of node 9, one at a time

LIST





# Delete node 9



Delete node 9 from LIST

LIST

9

# THE END

---

MIT OpenCourseWare  
<http://ocw.mit.edu>

15.082J / 6.855J / ESD.78J Network Optimization  
Fall 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.