
# Lecture 13

Produced by Dr. Worldwide

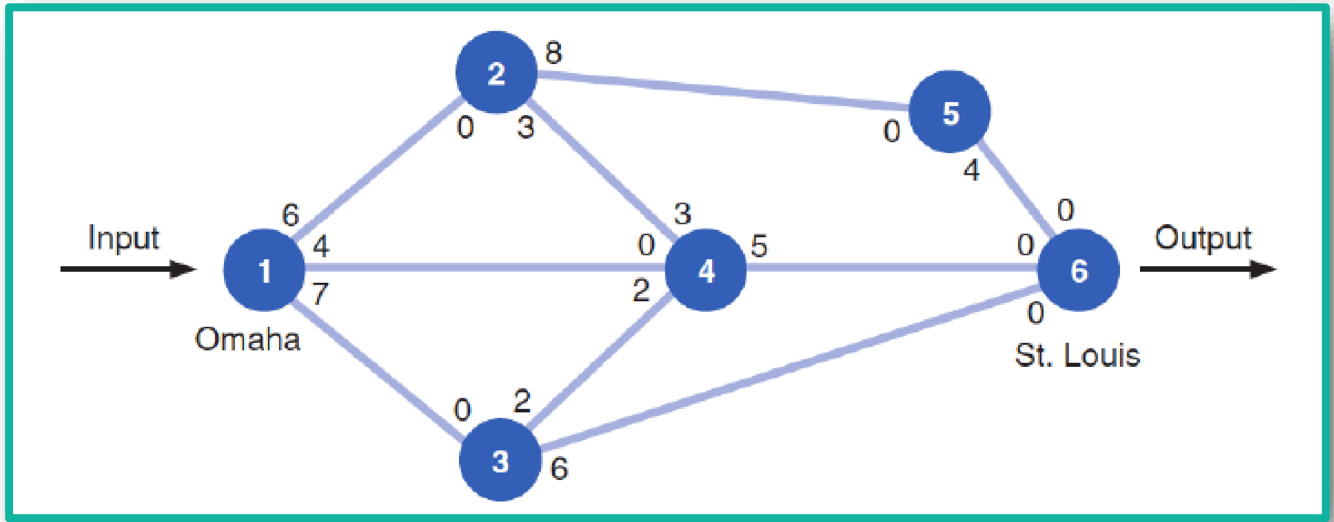
*Welcome to the 305*

# Maximal Flow Problem


- Sometimes the branches in a network have limitations on the capacity
- Suppose we are trying to move some resource (e.g. water, gas, oil) through a network of pipelines
- Pipelines are represented as edges in a graph (directed or undirected) and each has a finite capacity that determines how much can flow through them
- Source node produces the resource and a destination node receives it
- Q: What is the maximum amount of flow that can be moved through the network?

# Ex: Railway System

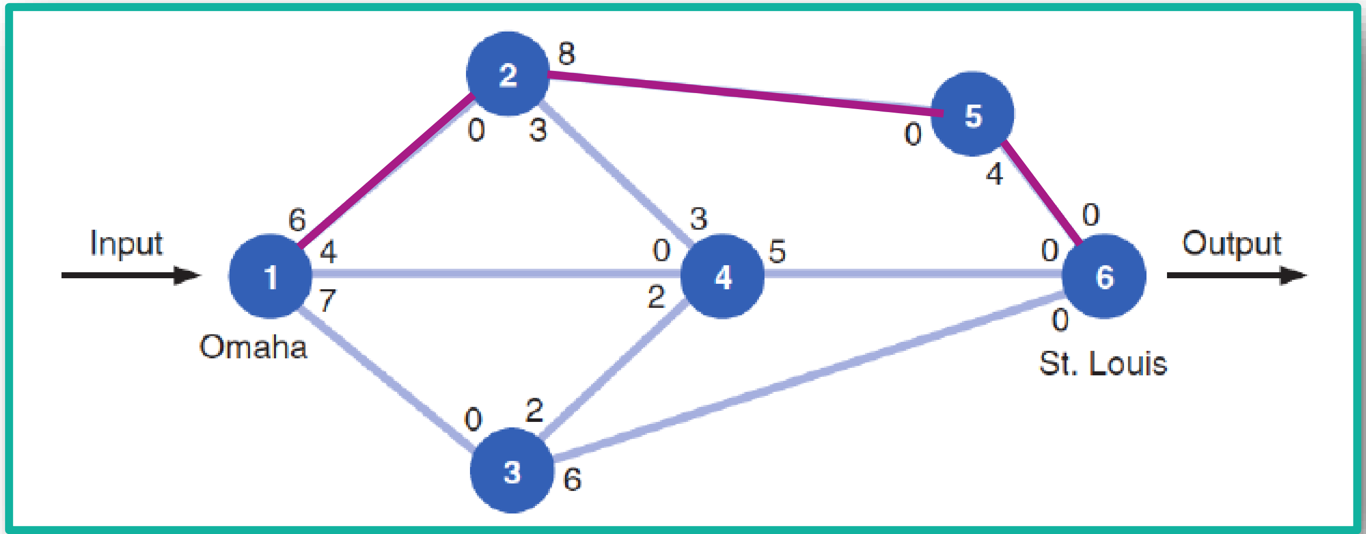
- Scott Tractor Company ships tractor parts from Omaha to St. Louis by railroad
- A contract limits the number of railroad cars available on each branch
- Graph of network showing the capacity (# of cars) leaving a node along an edge




- Q: Is this undirected or directed?

# Ex: Railway System

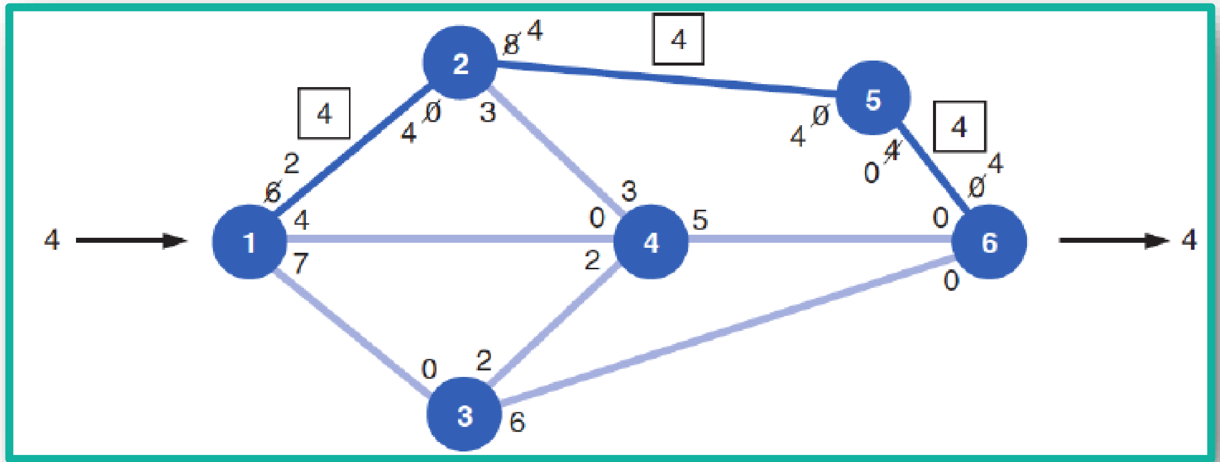
- We begin by choosing an arbitrary path from the origin to the destination
- A path can be defined by an ordering of nodes separated by hyphens
- For example, choose the path 1-2-5-6




- Q: What is the smallest capacity along this path?

# Ex: Railway System

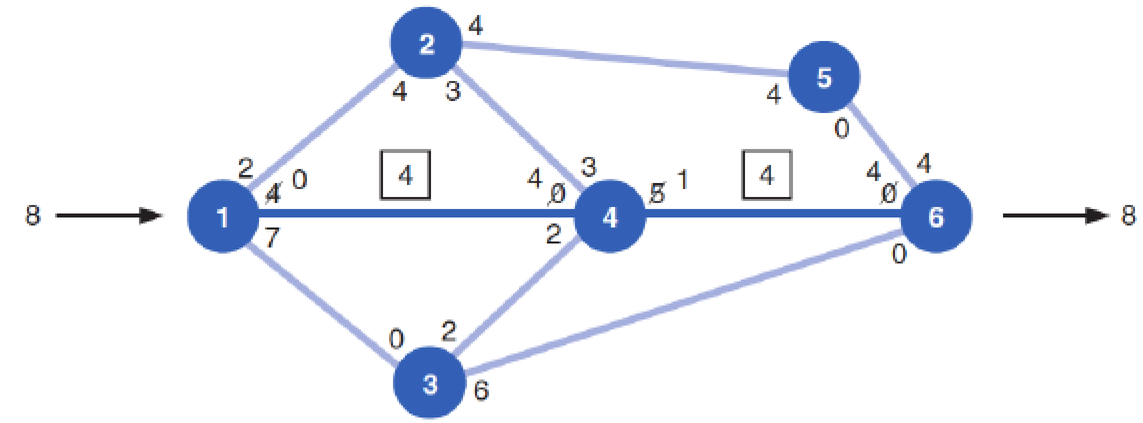
- The smallest capacity along the path is 4, corresponding to edge (5,6)
- A directed edge can be defined as an ordered pair of two nodes
- The capacity along the path 1-2-5-6 is 4
- Update by decreasing the capacities along the edges (1,2), (2,5), and (5,6) by 4 and increasing the capacities along the edges (6,5), (5,2), and (2,1) by 4



# Ex: Railway System

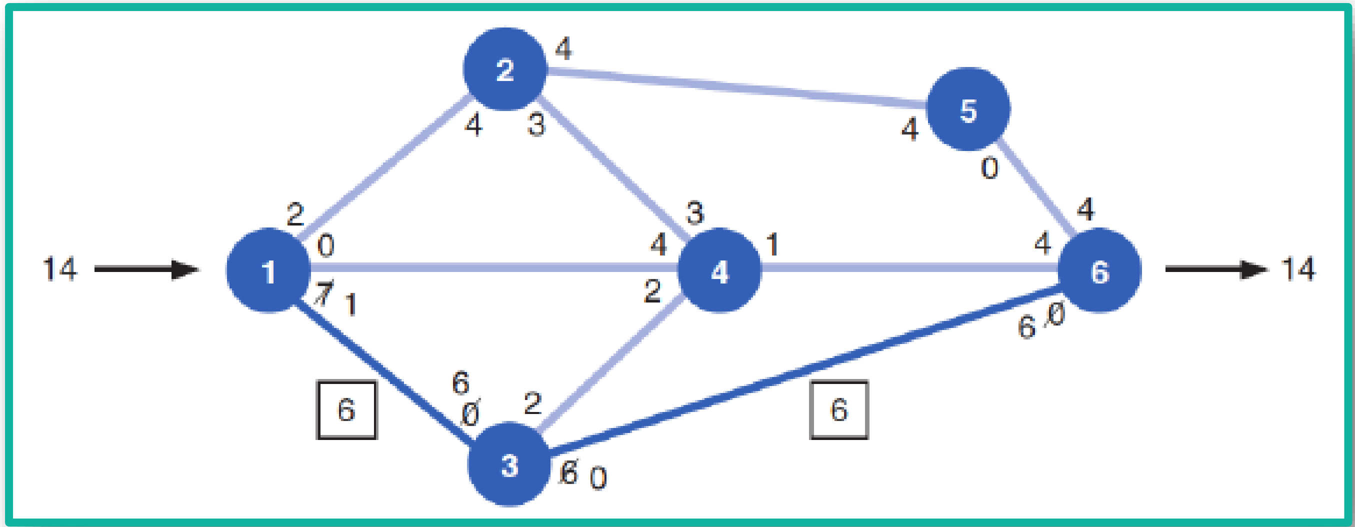

- Next, choose an arbitrary path from the updated graph
- For example, choose 1-4-6
- The smallest capacity is 4 because of (1,4); therefore, the capacity of 1-4-6 is 4
- Update the capacities in the same way as before by decreasing in the direction 1-4-6 and increasing in the direction 6-4-1
- Update maximum flow

  $4 + 4 = 8$



# Ex: Railway System

- Choose another arbitrary path from the updated graph like 1-3-6
- The smallest capacity is 6 corresponding to edge (3,6)
- Update the capacities according to the capacity of 1-3-6 which is 6



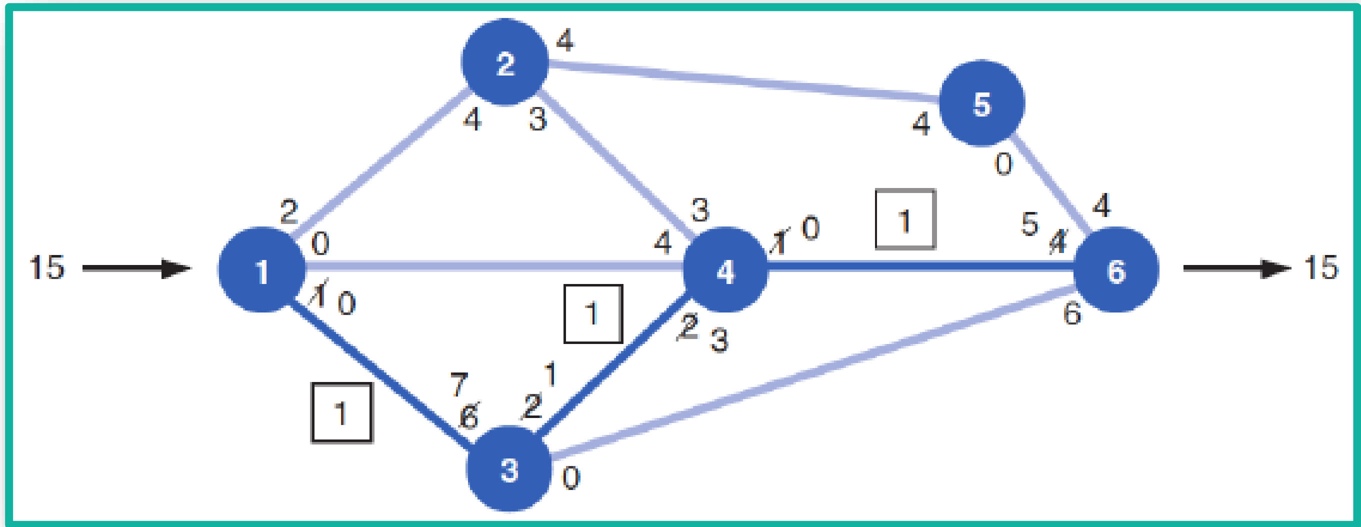

- The maximum flow now is $4 + 4 + 6 = 14$

# Ex: Railway System

- There are only two paths with available capacity from the node 1 to node 6
  - 1-2-4-6
  - 1-3-4-6

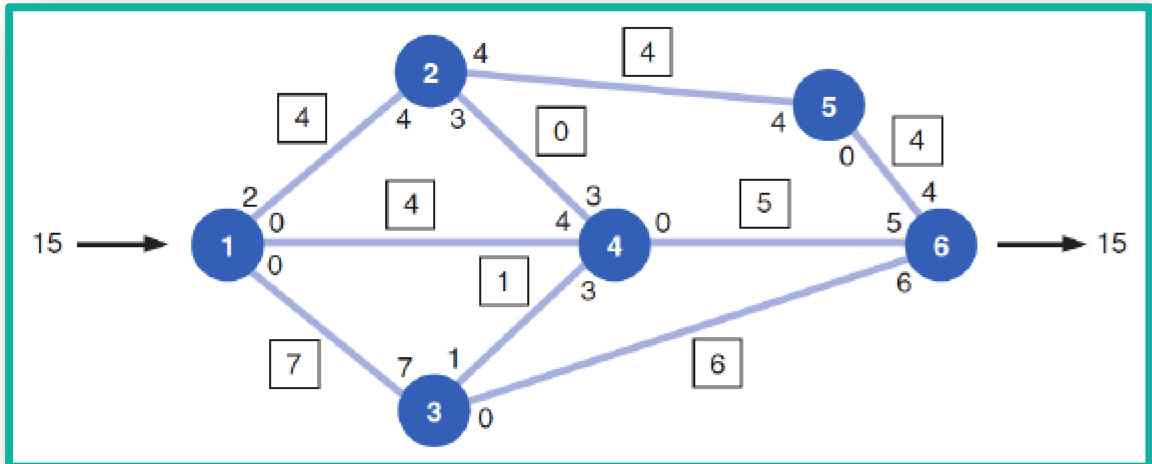
- We arbitrarily choose 1-3-4-6 with a capacity of 1 because of edge (4,6)

- Update graph and maximum flow is $4 + 4 + 6 + 1 = 15$



# Ex: Railway System

- No more paths to choose from in the updated graph



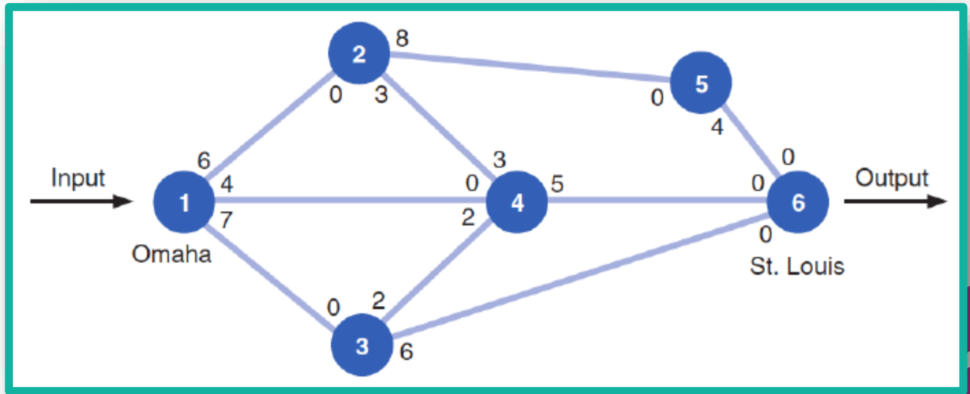

- Algorithm terminates at this point
- We say the maximum flow from node 1 to node 6 is 15

# Maximal Flow Problem

- Ford-Fulkerson's algorithm for identifying the maximal flow of a network
- Step 1: Arbitrarily select any path in the network from the origin to destination
- Step 2: Adjust the capacities at each node by subtracting the maximal flow for the path selected in the 1st step
- Step 3: Add the maximal flow along the path in the opposite direction
- Step 5: Repeat previous steps until there are no more paths with flow capacity

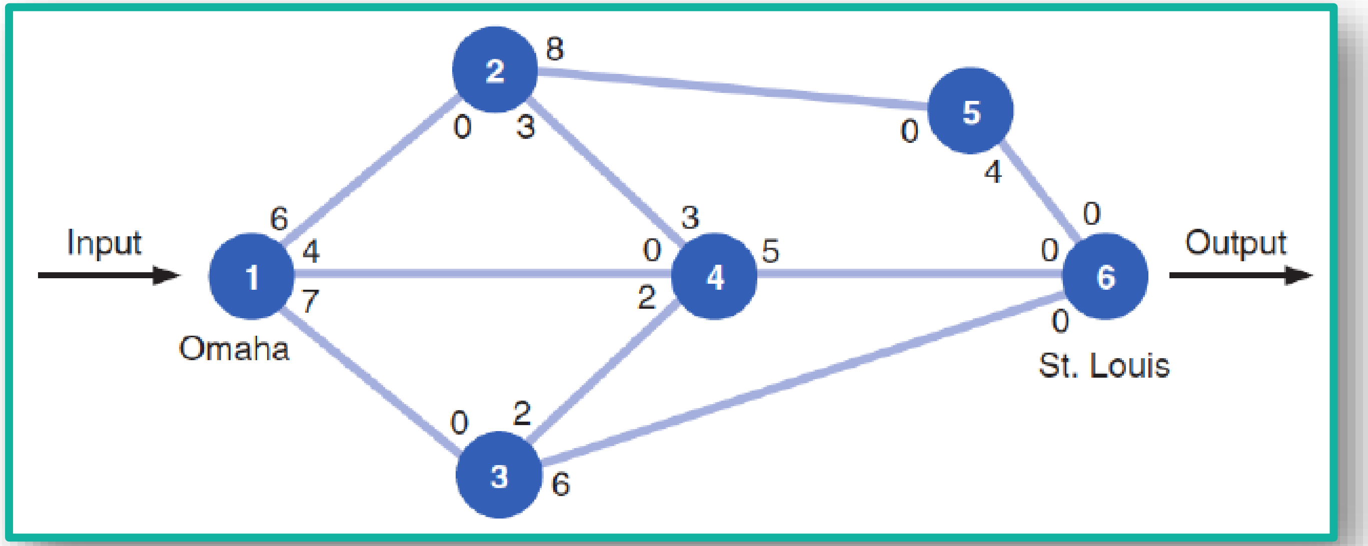
# Ex: Railway System

- Decision variables
  - $x_{ij} = amount\ of\ flow\ going\ through\ the\ edge\ (i,j)$
  - $i = \{1,2,3,\cdots,6\}$
  - $j = \{1,2,3,\cdots,6\}$
- Objective function
  - Want to maximize the flow from node 1 to node 6
  - The flow conservation principle says that these must be equal
  - $Z = x_{12} + x_{13} + x_{14}$
  - $Z = x_{56} + x_{46} + x_{36}$
  - Either one works



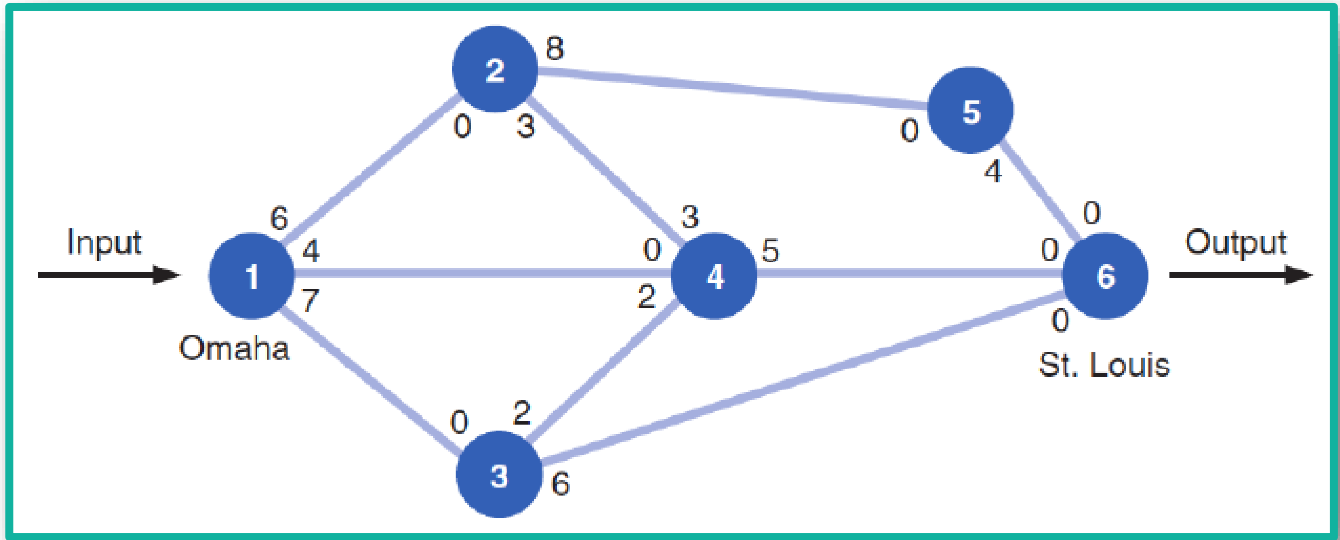
# Ex: Railway System

- Constraints to control flow
  - $x_{12} + x_{13} + x_{14} = x_{56} + x_{46} + x_{36}$ (Origin to destination)
  - $x_{12} + x_{42} = x_{24} + x_{25}$ (Through node 2)
  - $x_{13} + x_{43} = x_{34} + x_{36}$ (Through node 3)
  - $x_{14} + x_{24} + x_{34} = x_{42} + x_{43} + x_{46}$ (Through node 4)
  - $x_{25} = x_{56}$ (Through node 5)



# Ex: Railway System

- Constraints to control capacity
  - $x_{12} \leq 6$
  - $x_{13} \leq 7$
  - $x_{14} \leq 4$
  - $x_{24} \leq 3$
  - $x_{25} \leq 8$
  - $x_{34} \leq 2$
  - $x_{36} \leq 6$
  - $x_{42} \leq 3$
  - $x_{43} \leq 2$
  - $x_{46} \leq 5$
  - $x_{56} \leq 4$




- Constraints to control domain
  - $x_{ij} \in \{0,1,2,3,\cdots\}$

# Ex: Railway System


- Download MaximalFlow.xlsx from course website from link Sheet 1

**Maximal flow**

| Select branch | Edge (i,j) i | j | Capacity |
|---|---|---|---|
| 5 | 1 | 2 | 6 |
| 6 | 1 | 3 | 7 |
| 4 | 1 | 4 | 4 |
| 1 | 2 | 4 | 3 |
| 4 | 2 | 5 | 8 |
| 0 | 3 | 4 | 2 |
| 6 | 3 | 6 | 6 |
| 0 | 4 | 2 | 3 |
| 0 | 4 | 3 | 2 |
| 5 | 4 | 6 | 5 |
| 4 | 5 | 6 | 4 |

*Flow constraints:*

| Node | In | Out | Network Flow |
|---|---|---|---|
| 2 | 5 | 5 | 0 |
| 3 | 6 | 6 | 0 |
| 4 | 5 | 5 | 0 |
| 5 | 4 | 4 | 0 |
| Dest./Origin | 15 | 15 | 0 |

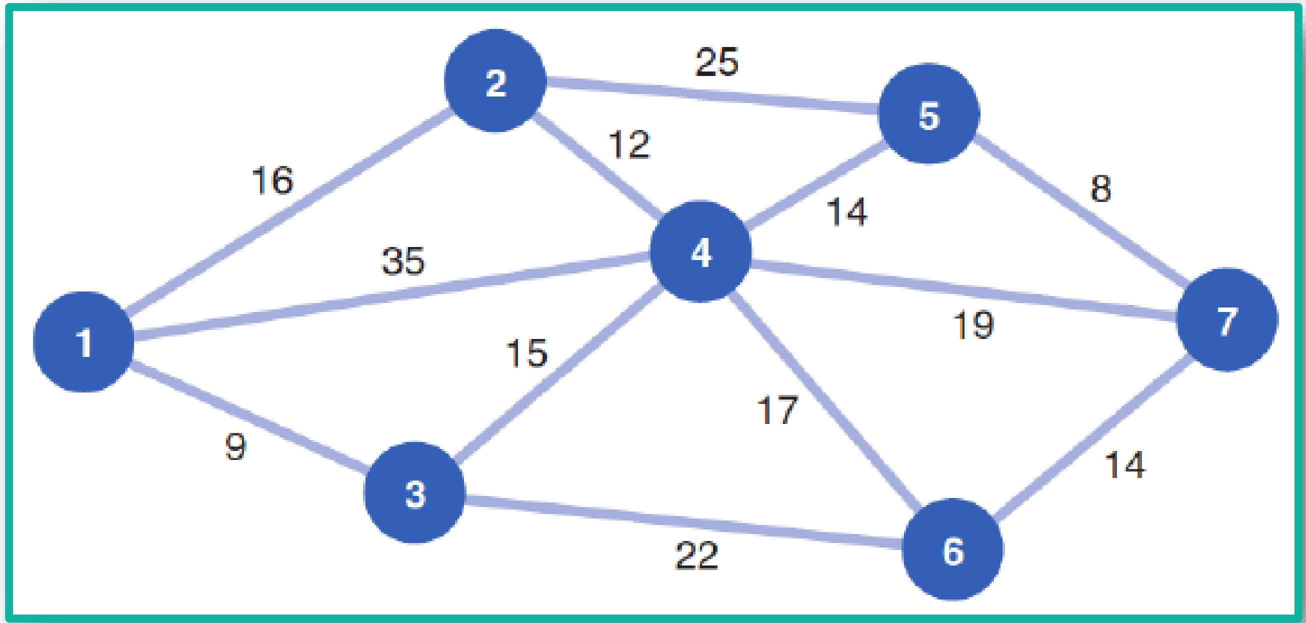
Maximal flow= 15 =A12+A15+A16

# Minimal Spanning Tree


- Like the shortest route problem where goal was to find the shortest path from a single origin to all other nodes in a network
- The minimal spanning tree is the set of edges having the minimum total length that connects all the nodes in the network
- A minimal spanning tree has no origin node

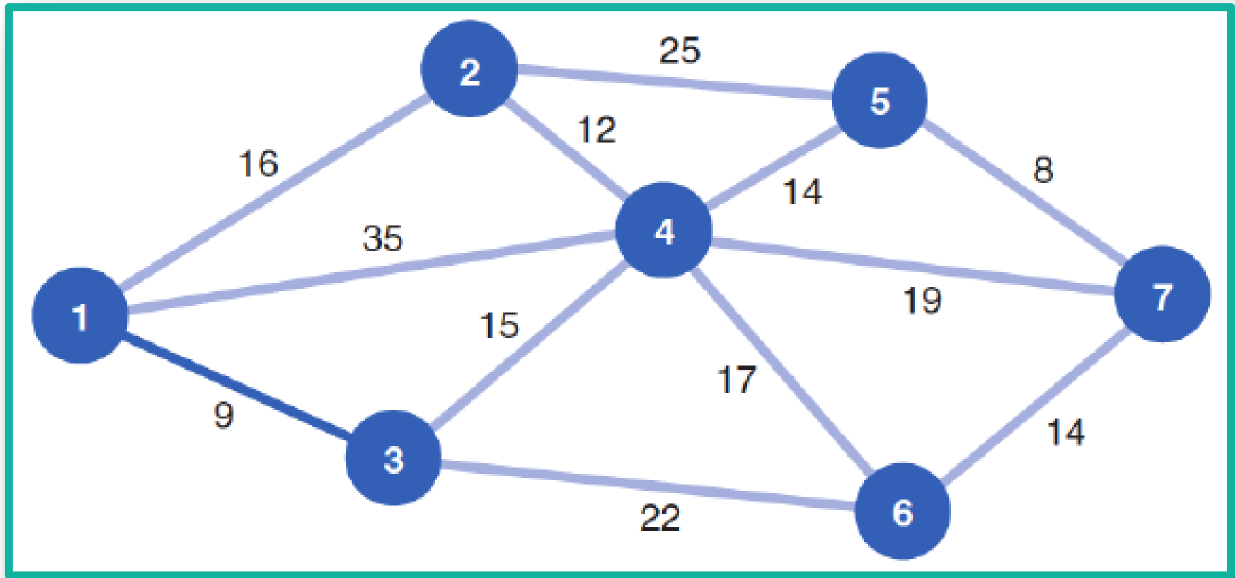
# Ex: Cable Company

- Cable company wants to connect all 7 houses in a neighborhood
- Network shows length of cable in feet required for each possible path
- Q: What is the shortest path that spans (connects) all the houses?



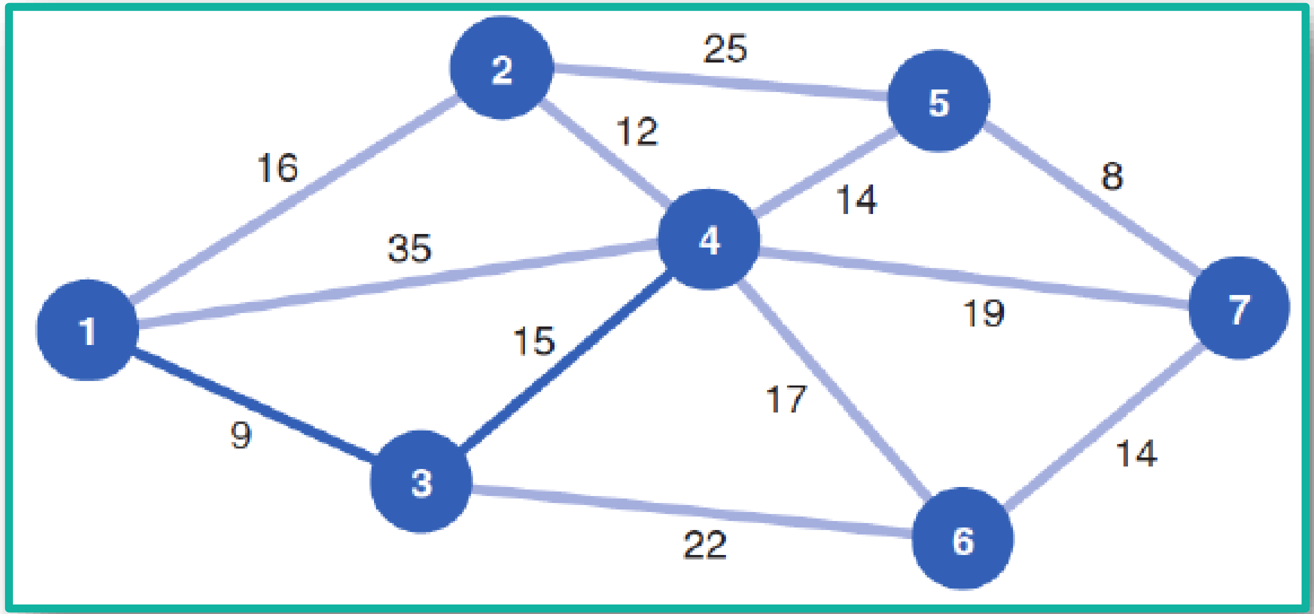
# Ex: Cable Company

- We can start from any node, but the convention is to start with node 1
- Define the set of nodes in the spanning tree as $S = \{1\}$
- Find the shortest edge from node 1, add that edge to the spanning tree, and the corresponding node to set $S$
- This would be edge $(1,3)$
- Update $S = \{1,3\}$



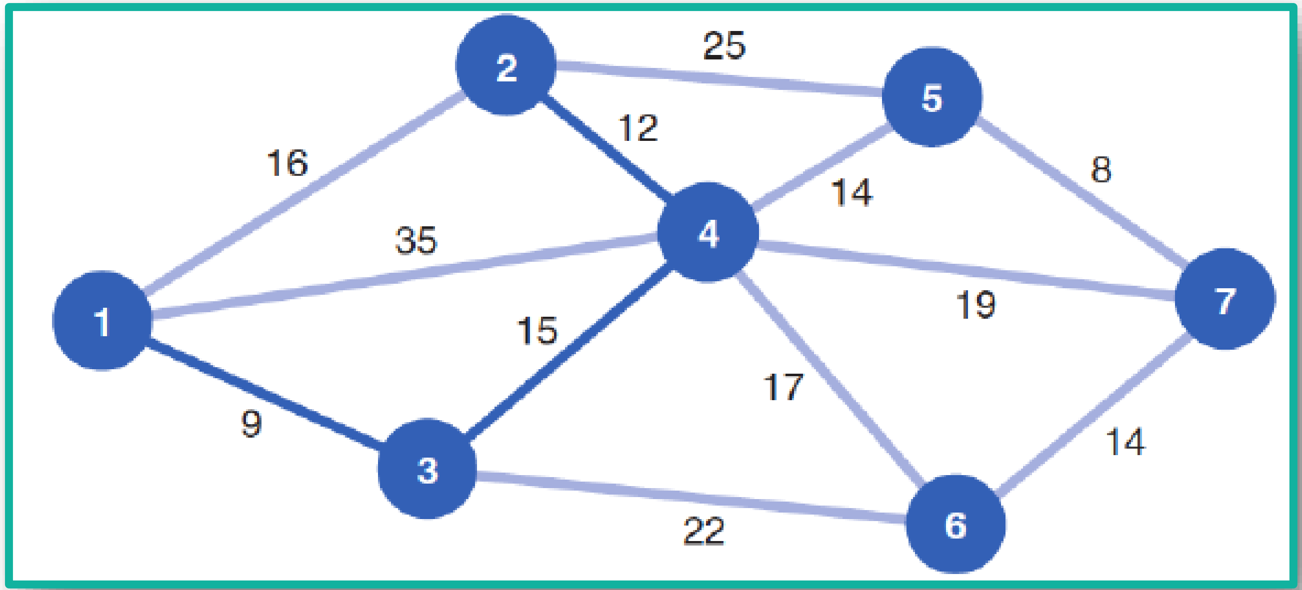
# Ex: Cable Company

- Choose shortest edge connecting nodes in $S = \{1,3\}$ to the other nodes
- This would be edge (3,4)
- Update $S = \{1,3,4\}$



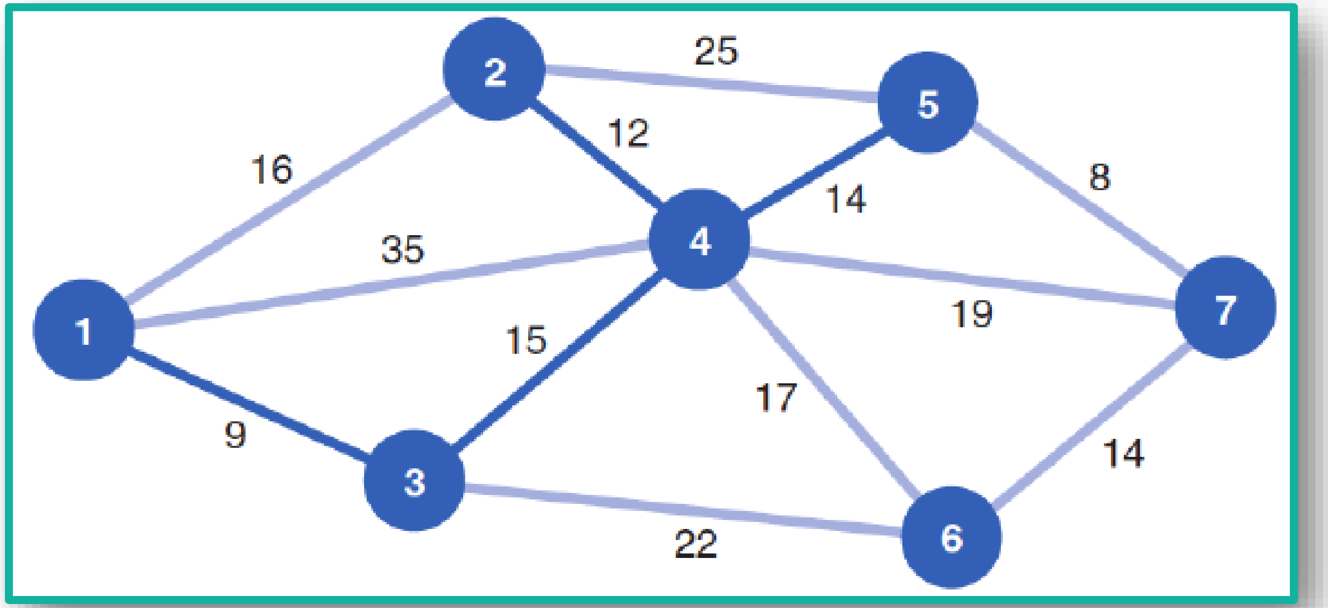
# Ex: Cable Company

- Choose shortest edge connecting nodes in $S = \{1,3,4\}$ to the other nodes
- This would be edge $(4,2)$
- Update $S = \{1,3,4,2\}$



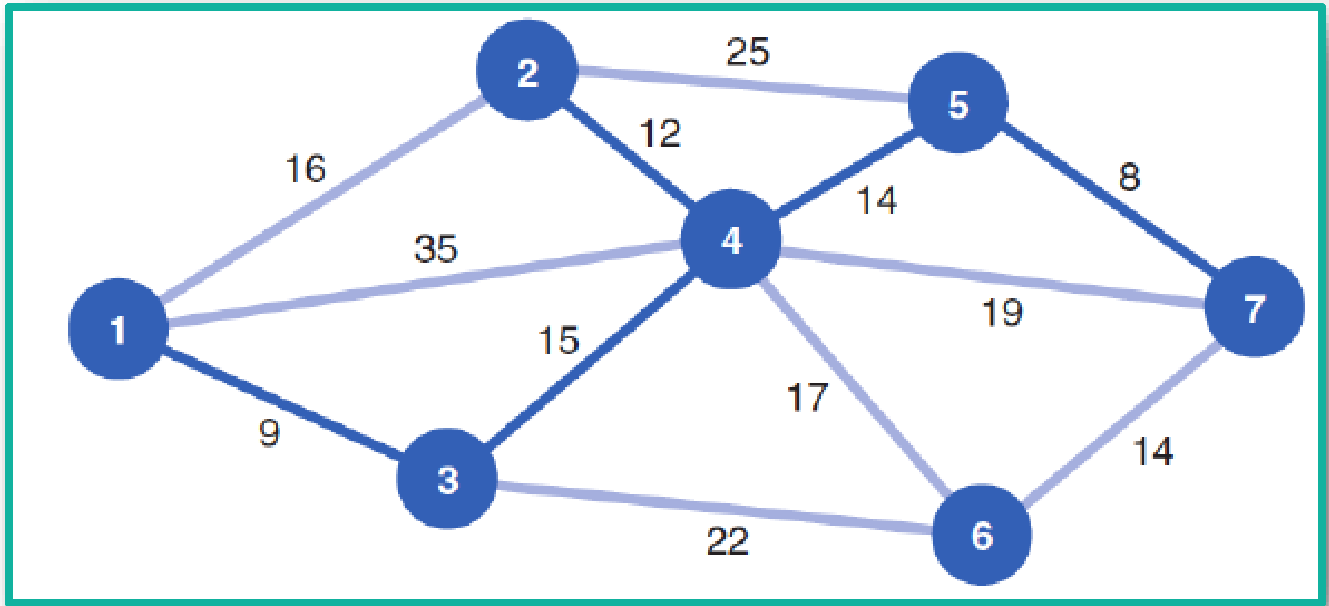
# Ex: Cable Company

- Choose shortest edge connecting nodes in $S = \{1,3,4,2\}$ to the other nodes
- This would be edge (4,5)
- Update $S = \{1,3,4,2,5\}$



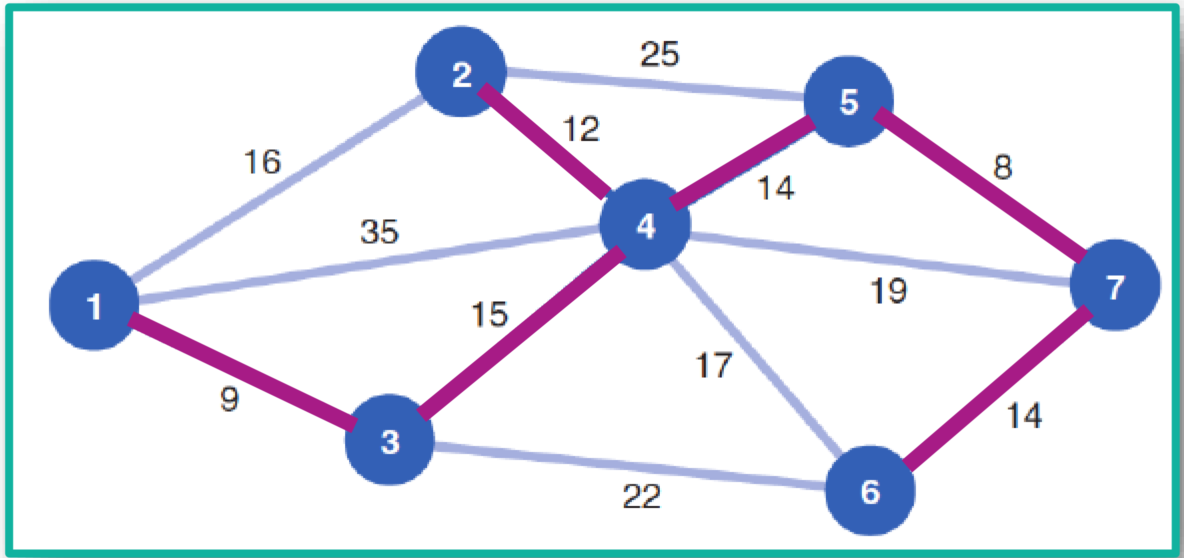
# Ex: Cable Company

- Choose shortest edge connecting nodes in $S = \{1,3,4,2,5\}$ to the other nodes
- This would be edge (5,7)
- Update $S = \{1,3,4,2,5,7\}$



# Ex: Cable Company

- Choose shortest edge connecting nodes in $S = \{1,3,4,2,5,7\}$ to the last node 6
- This would be edge $(7,6)$
- Length of the minimal spanning tree is the sum of lengths of chosen edges
  $9 + 15 + 12 + 14 + 8 + 14 = 72$



# Minimal Spanning Tree


- Prim algorithm for identifying the minimal spanning tree
- Step 1: Select an arbitrary node to start (usually node 1)
- Step 2: Select the node closest to the starting node to join the spanning tree
- Step 3: Select the closest node not currently in the spanning tree
- Step 4: Repeat the 3rd step until all nodes have joined the spanning tree


# The End

# Dale