Lecture 12

Produced by Dr. Worldwide

Welcome to the 305

Shortest Route Problem

- Dijkstra's algorithm for identifying the shortest route
- Step 1: Select the node with the shortest route from the origin
- Step 2: Create a permanent set that includes the origin and the node chosen in the 1st step
- Step 3: Identify all nodes that are adjacent to the nodes in the permanent set
- Step 4: Select the node with the shortest route from the group of nodes adjacent to the nodes in the permanent set. Add the chosen node to the permanent set
- Step 5: Repeat the 3rd and 4th steps until all nodes are in the permanent set

Shortest Route Problem

- We want to reformulate the problem as a linear program
- The origin in the shortest route problem can be thought of as a single supply node
- The other nodes can be thought of as demand nodes
- Source has supply equal to the number of nodes in the graph minus one (itself)
- Each demand node requires a single unit
- Distance between nodes corresponds to transportation cost of that edge
- To reduce the number of variables, we assume units only flow in the direction of a higher node number

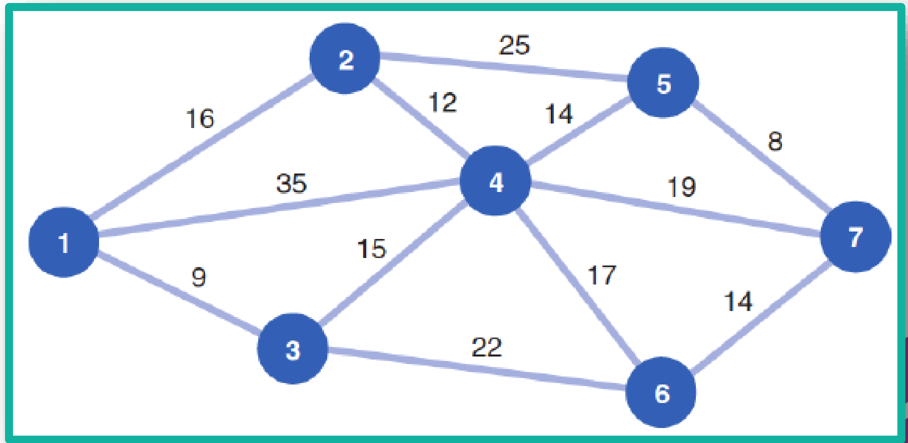
Ignore x_{ij} if $i \geq j$

Ex: Shortest Shipping Route

- Decision variables
  - $x_{ij} = number\ of\ trucks\ transported\ along\ edge\ (i,j)$
  - $i = \{1,2,3,\cdots,7\}$
  - $j = \{1,2,3,\cdots,7\}$
  - $i < j$

- Objective function

$$Z = 16x_{12} + 35x_{14} + 9x_{13} + 12x_{24} \\ +15x_{34} + 25x_{25} + 14x_{45} + 17x_{46} \\ +22x_{36} + 8x_{57} + 19x_{47} + 14x_{67}$$



Ex: Shortest Shipping Route

- Whatever gets into a node leaves the node is known as flow conservation

- Origin produces 1 unit of flow and the node with largest index must get 1 unit

- Constraints
  - $x_{12} + x_{13} + x_{14} = 6$ (Out of node 1)
  - $x_{12} = x_{24} + x_{25} + 1$ (Through node 2)
  - $x_{13} = x_{34} + x_{36} + 1$ (Through node 3)
  - $x_{14} + x_{24} + x_{34} = x_{45} + x_{46} + x_{47} + 1$ (Through node 4)
  - $x_{25} + x_{45} = x_{57} + 1$ (Through node 5)
  - $x_{36} + x_{46} = x_{67} + 1$ (Through node 6)
  - $x_{47} + x_{57} + x_{67} = 1$ (Into node 7)
  - x_{ij} is an integer

- Q: Why the +1 in the constraints?

Ex: Shortest Shipping Route

- Download ShortestRoute.xlsx from course website from link Sheet 1
- Look at tab titled One-to-All

Units shipped	Node	City	Node	City	Distance (hours)
1	1	Los Angeles	2	Salt Lake City	16
5	1	Los Angeles	3	Phoenix	9
0	1	Los Angeles	4	Denver	35
0	2	Salt Lake City	4	Denver	12
0	2	Salt Lake City	5	Des Moines	25
3	3	Phoenix	4	Denver	15
1	3	Phoenix	6	Dallas	22
1	4	Denver	5	Des Moines	14
0	4	Denver	6	Dallas	17
1	4	Denver	7	St. Louis	19
0	5	Des Moines	7	St. Louis	8
0	6	Dallas	7	St. Louis	14
				Total	161

Ex: Shortest Shipping Route

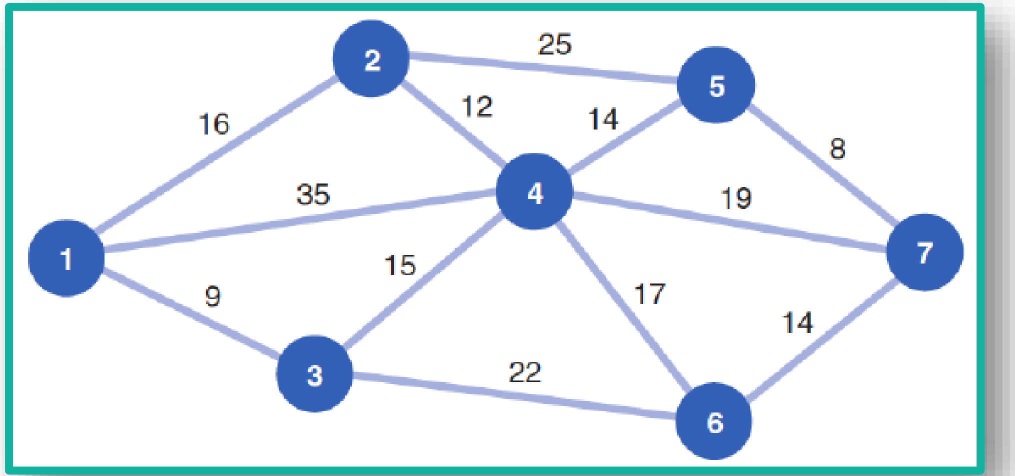
- Look at tab titled One-to-One

Arcs used	Node	City	Node	City	Distance (hours)
0	1	Los Angeles	2	Salt Lake City	16
1	1	Los Angeles	3	Phoenix	9
0	1	Los Angeles	4	Denver	35
0	2	Salt Lake City	4	Denver	12
0	2	Salt Lake City	5	Des Moines	25
1	3	Phoenix	4	Denver	15
0	3	Phoenix	6	Dallas	22
0	4	Denver	5	Des Moines	14
0	4	Denver	6	Dallas	17
1	4	Denver	7	St. Louis	19
0	5	Des Moines	7	St. Louis	8
0	6	Dallas	7	St. Louis	14
				Total	43

- Q:What is the difference between One-to-All and One-to-One?

Shortest Route Problem

- Assumed that we had supply at origin (node 1) to fulfill demand at all other nodes

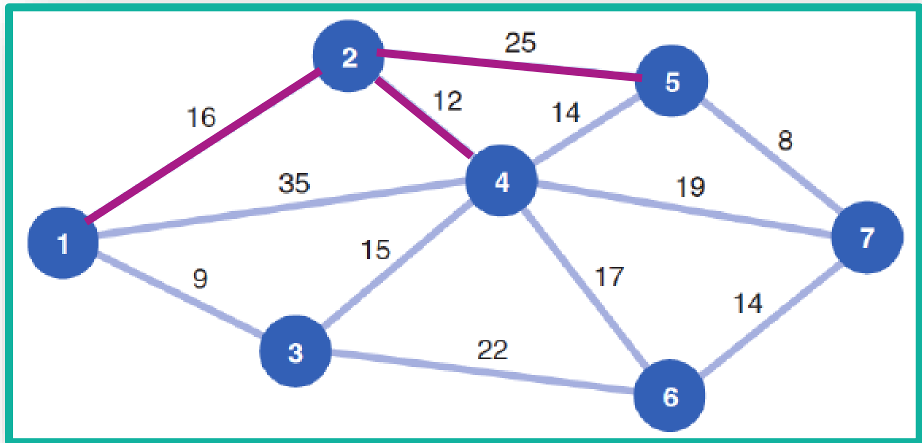


- Suppose we only wanted to get one unit from the origin to one destination node
- We must modify the constraints to reflect a supply and demand of one

Ex: Shortest Shipping Route

- Modification of constraints
  - One origin to many destinations

Flow constraints:			
Node	Network Flo	Constraint	Value
1	1	=	1
2	0	=	0
3	0	=	0
4	0	=	0
5	0	=	0
6	0	=	0
7	1	=	1



  - One origin to one destination

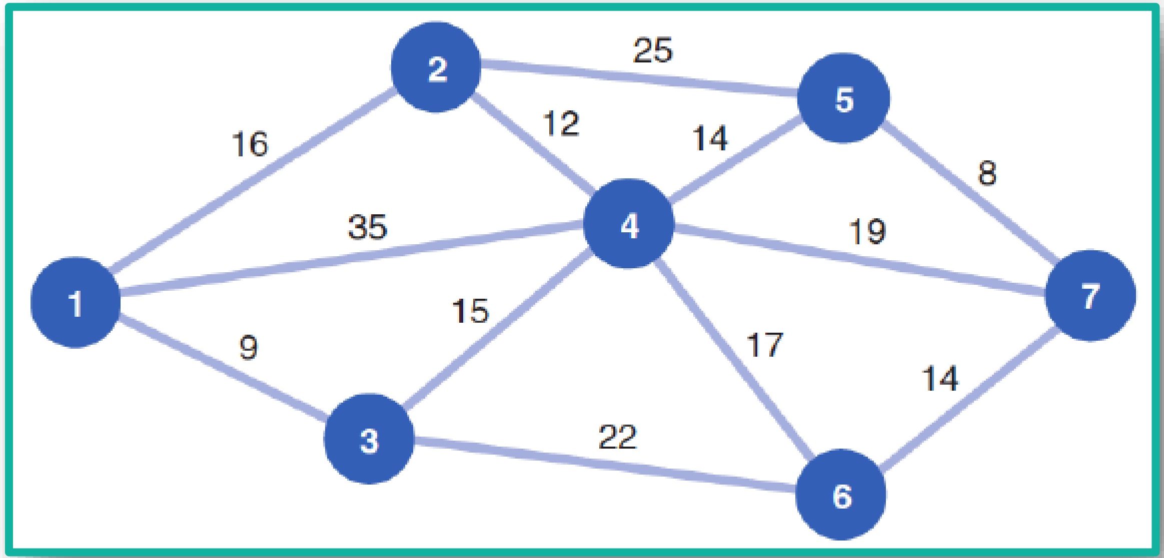
Flow constraints:			
Node	Network Flow	Constraint	Value
1	6	=	6
2	1	=	1
3	1	=	1
4	1	=	1
5	1	=	1
6	1	=	1
7	1	=	1

Difference Between In and Out

$$x_{12} - x_{24} - x_{25}$$

Shortest Route Problem

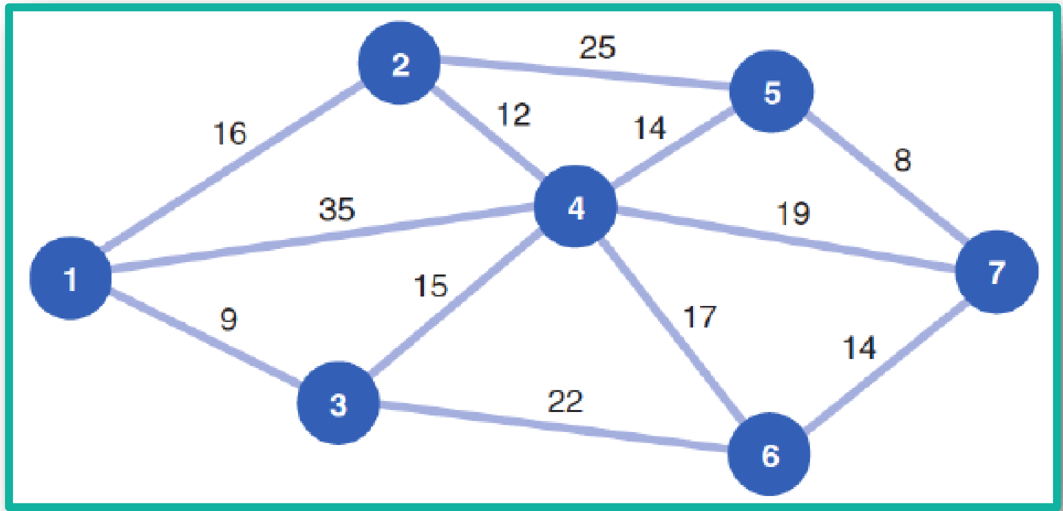
- Assumed that edges had direction from West (node 1) to East (nodes 2-7)



- Suppose direction doesn't matter
- We must consider both directions for each branch

Ex: Shortest Shipping Route

- Decision variables for undirected network
  - $x_{ij} = number\ of\ trucks\ transported\ along\ edge\ (i, j)$
  - $i = \{1,2,3, \cdots, 7\}$
  - $j = \{1,2,3, \cdots, 7\}$
  - $i < j$ or $i > j$

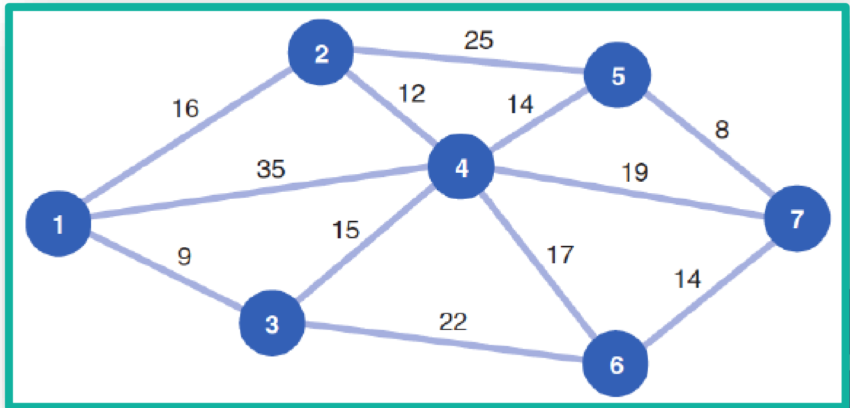


- Objective function for undirected network

$$Z = 16(x_{12}+x_{21}) + 35(x_{14} + x_{41}) + 9(x_{13}+x_{31}) + 12(x_{24} + x_{42}) \\ + 15(x_{34}+x_{43}) + 25(x_{25} + x_{52}) + 14(x_{45} + x_{54}) + 17(x_{46}+x_{64}) \\ + 22(x_{36} + x_{63}) + 8(x_{57} + x_{75}) + 19(x_{47} + x_{74}) + 14(x_{67} + x_{76})$$

Ex: Shortest Shipping Route

- Constraints for undirected network
  - $x_{21} + x_{31} + x_{41} - x_{12} - x_{13} - x_{14} = -6$ (Node 1)
  - $x_{12} + x_{52} + x_{42} - x_{21} - x_{24} - x_{25} = 1$ (Node 2)
  - $x_{13} + x_{43} + x_{63} - x_{31} - x_{34} - x_{36} = 1$ (Node 3)
  - $x_{14} + x_{24} + x_{34} + x_{54} + x_{64} + x_{74}$
    $-x_{41} - x_{42} - x_{43} - x_{45} - x_{46} - x_{47} = 1$ (Node 4)
  - $x_{25} + x_{45} + x_{75} - x_{52} - x_{54} - x_{57} = 1$ (Node 5)
  - $x_{36} + x_{46} + x_{76} - x_{63} - x_{64} - x_{67} = 1$ (Node 6)
  - $x_{47} + x_{57} + x_{67} - x_{74} - x_{75} - x_{76} = 1$ (Node 7)
  - x_{ij} is an integer



Ex: Shortest Shipping Route

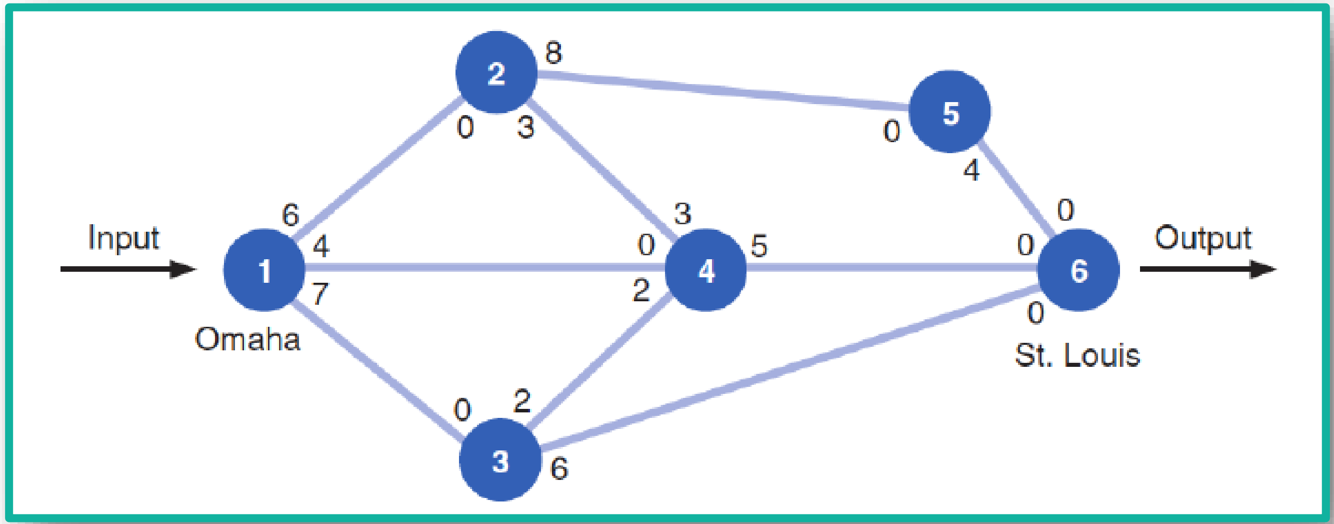
- Download ShortestRoute-1.xlsx from course website from link Sheet 2
- The tabs One-to-All-Directed and One-to-One-Directed contain previous results
- Closely examine the tabs One-to-All-Undirected and One-to-One-Undirected
- Q: What do you notice is the same between directed and undirected problems?
- Q: What do you notice is different between directed and undirected problems?
- Q: What is the purpose of the number 1,000 in the distance matrix?

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

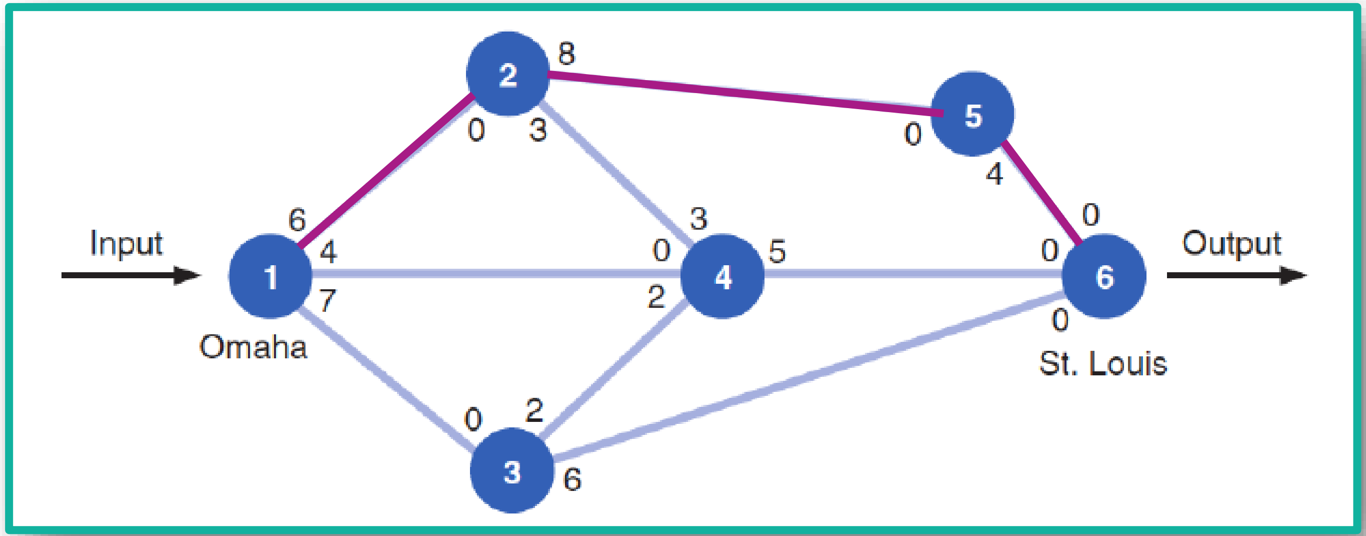
- Scott Tractor Company ships tractor parts from Omaha to St. Louis by railroad
- A contract limits the number of railroad cars available on each branch
- Graph of network showing the capacity (# of cars) leaving a node along an edge



- Q: Is this undirected or directed?

Ex: Railway System

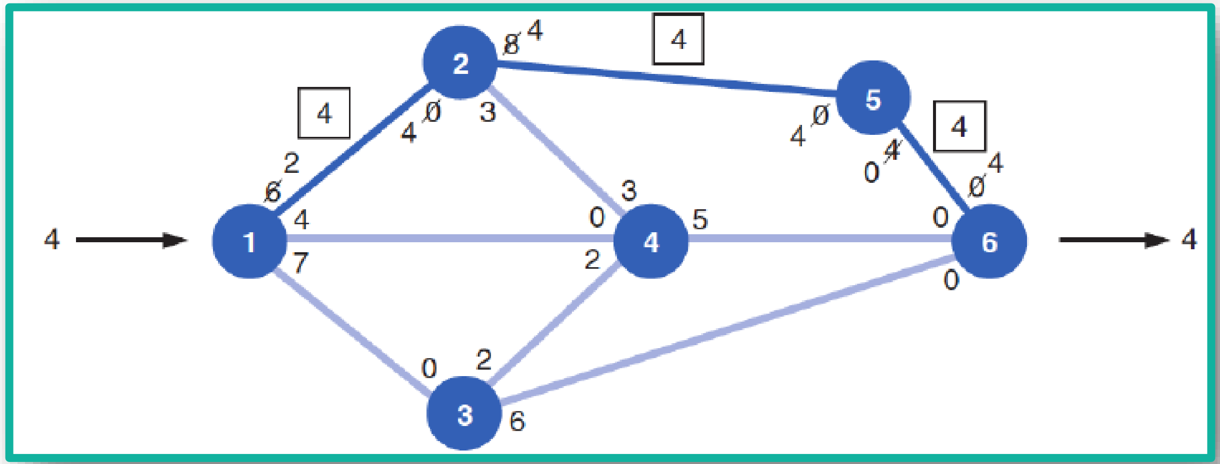
- We begin by choosing an arbitrary path from the origin to the destination
- A path can be defined by an ordering of nodes separated by hyphens
- For example, choose the path 1-2-5-6



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

Ex: Railway System

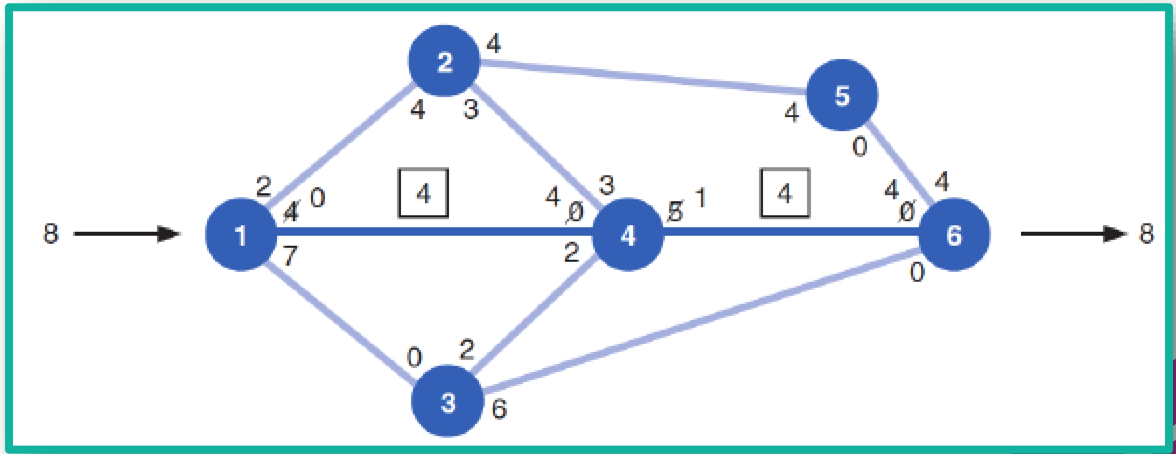
- The smallest capacity along the path is 4, corresponding to edge (5,6)
- A directed edge can be defined as an ordered pair of two nodes
- The capacity along the path 1-2-5-6 is 4
- Update by decreasing the capacities along the edges (1,2), (2,5), and (5,6) by 4 and increasing the capacities along the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



Ex: Railway System

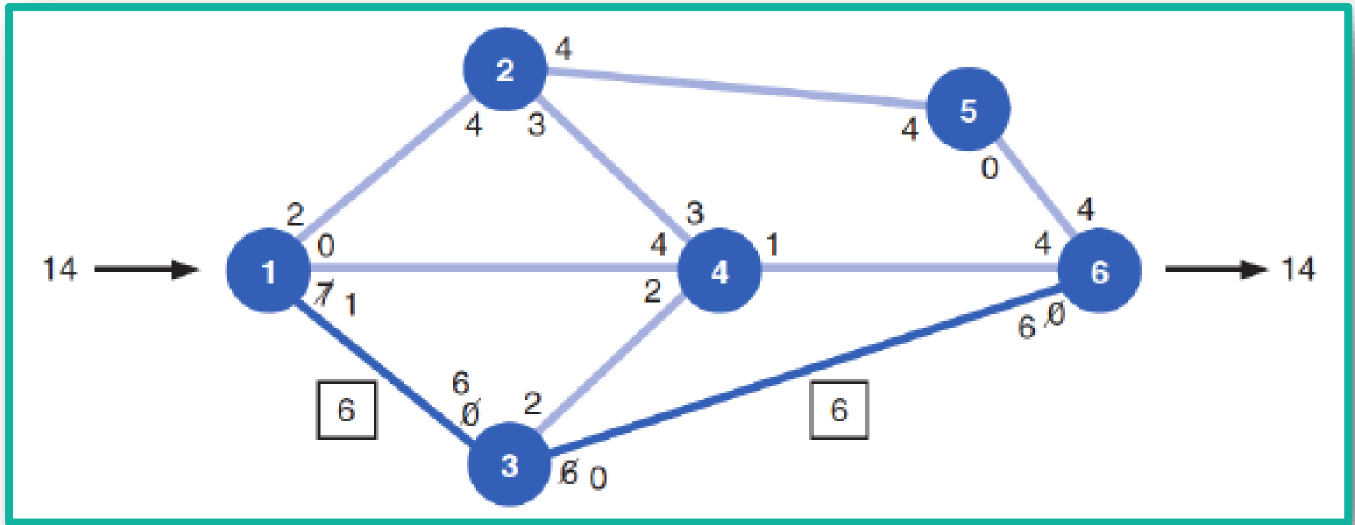
- Next, choose an arbitrary path from the updated graph
- For example, choose 1-4-6
- The smallest capacity is 4 because of (1,4); therefore, the capacity of 1-4-6 is 4
- Update the capacities in the same way as before by decreasing in the direction 1-4-6 and increasing in the direction 6-4-1
- Update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

  $4 + 4 = 8$



Ex: Railway System

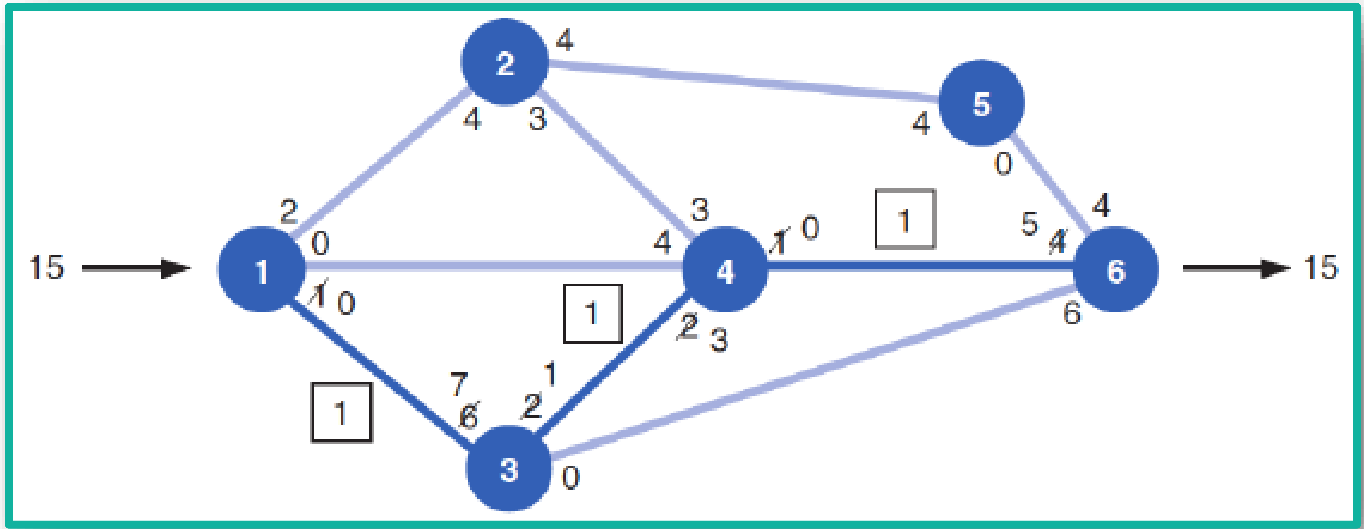
- Choose another arbitrary path from the updated graph like 1-3-6
- The smallest capacity is 6 corresponding to edge (3,6)
- Update the capacities according to the capacity of 1-3-6 which is 6



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

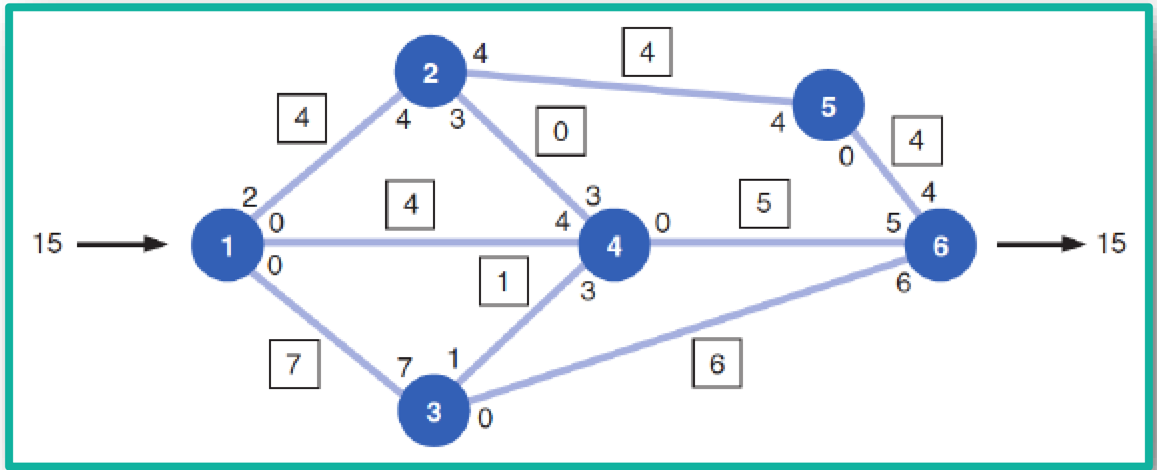
Ex: Railway System

- 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

Shortest Route 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

305 RESILIENT

The End

Dale