Management Science

Network Models

Dan Brandon, Ph.D., PMP

Session Objectives

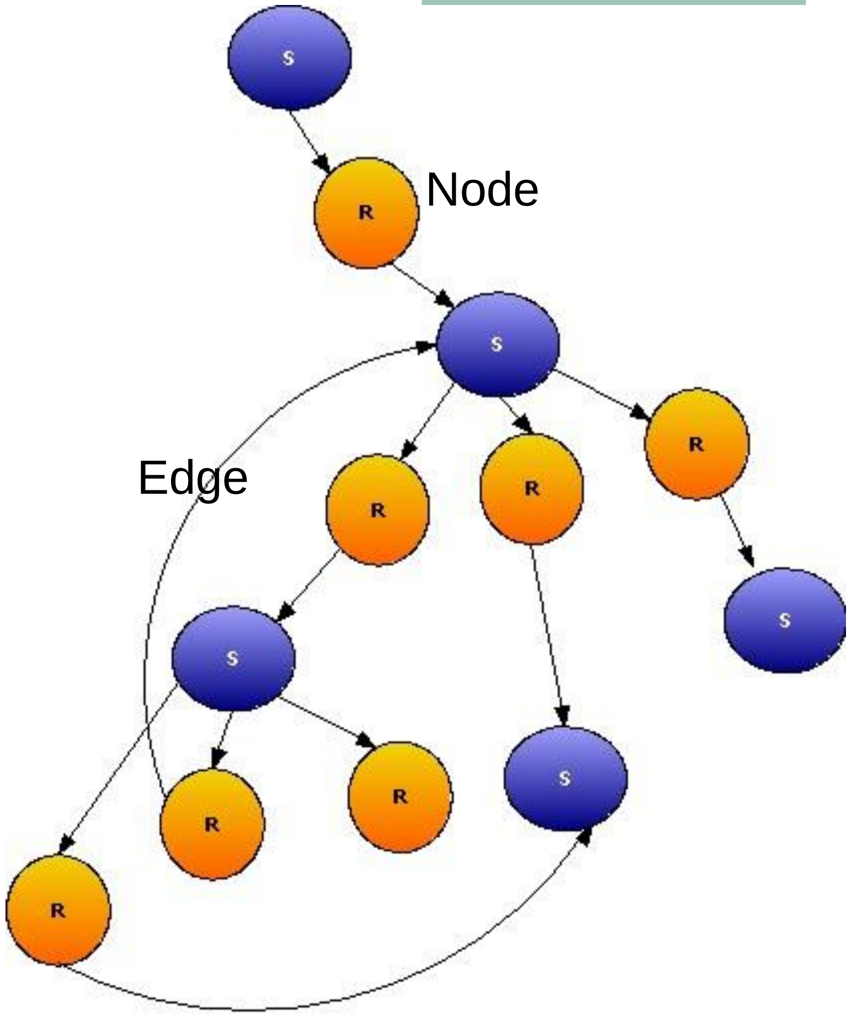
- Find the least number of hops (stops) thru a network
- Find the shortest or cheapest path through a network using the shortest-route technique
- Connect all points of a network while minimizing total distance using the minimal-spanning tree technique
- Determine the maximum flow through a network using the maximal-flow technique
- Understand how to use LP and matrix solution methods to network problems

Network Analysis & Graph Theory

- Problems lending themselves to “network analysis” have some type of structural relationship between entities which needs to be “optimized”
  - Communication between people, computers, departments, etc.
  - Flow of material between locations
  - Tasks in a projects (precedence)
  - The layout of workspaces, offices, manufacturing plant layout, etc.

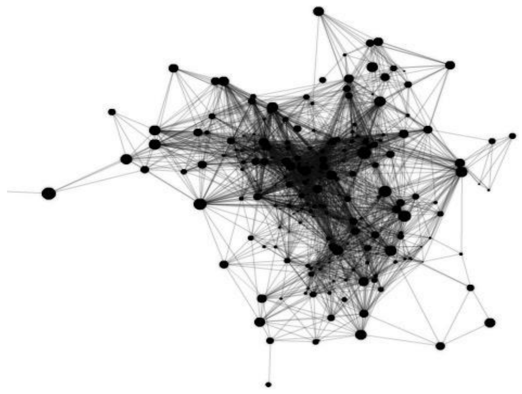
Network Graphs

- Network types of problems can be set up as mathematical “graphs” (the area of mathematics called “topology”)
- The graph has two kinds of elements:
  - Nodes
  - Edges (or arcs)



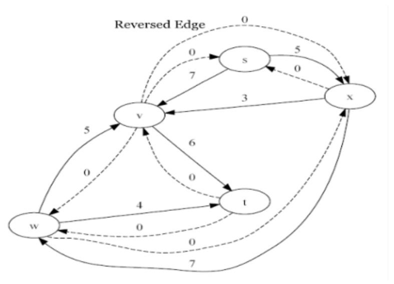
Nodes

- The nodes correspond to the entities needing to be structurally organized, such as:
  - People in an organization
  - Stops along a material flow
  - Tasks in a project
  - Computers in a network
- In general, the nodes are "states" in a system
- The nodes are generally represented as dots or circles



Edges

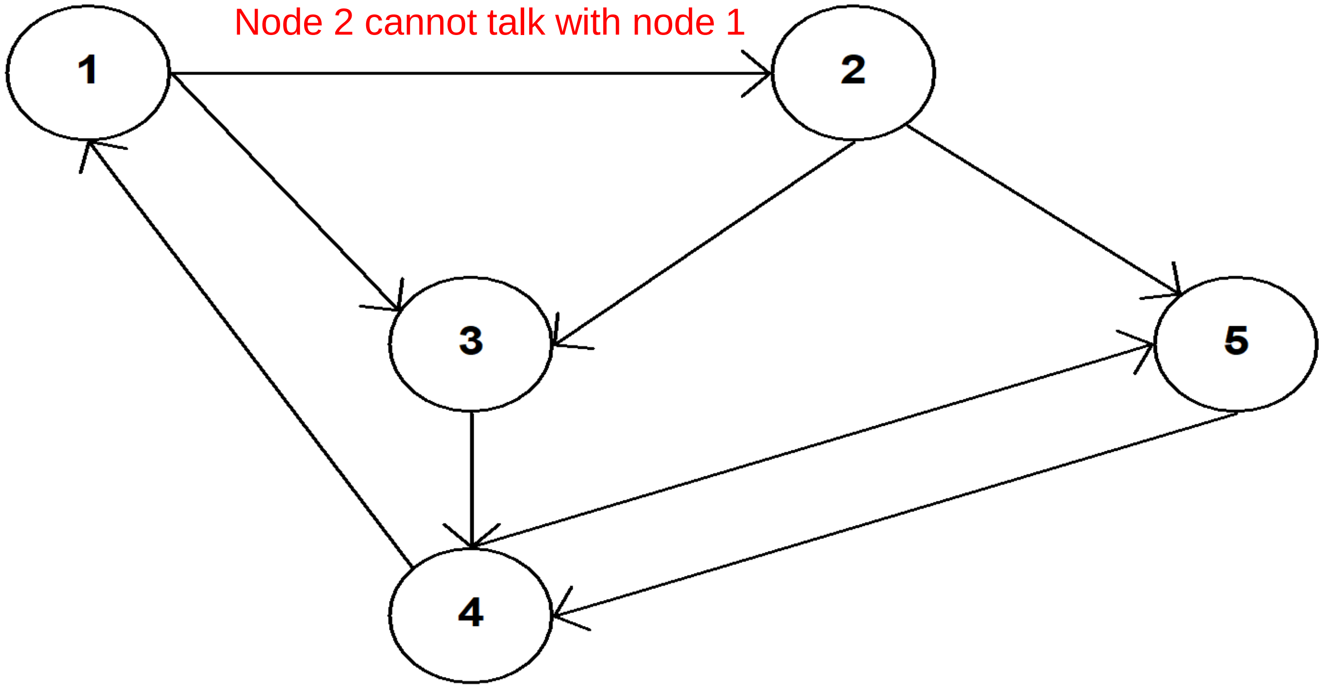
- The edges (or arcs) are line segments connecting the nodes
- Logically they may be:
  - Communication channels
  - Traffic lanes
  - Precedence relationships



- The edges can be directed, and if so, they begin at one node and end at another node; an arrow is shown at the ending node
  - For two way communication, two edges (each with arrows) are shown (if all edges are "two-way", arrows may be omitted)
- A "cost" may be associated with each edge, such as a time, distance, dollar cost, etc.

Example Network (adjacency) Graph

Node 1 can talk to node2
Node 2 cannot talk with node 1

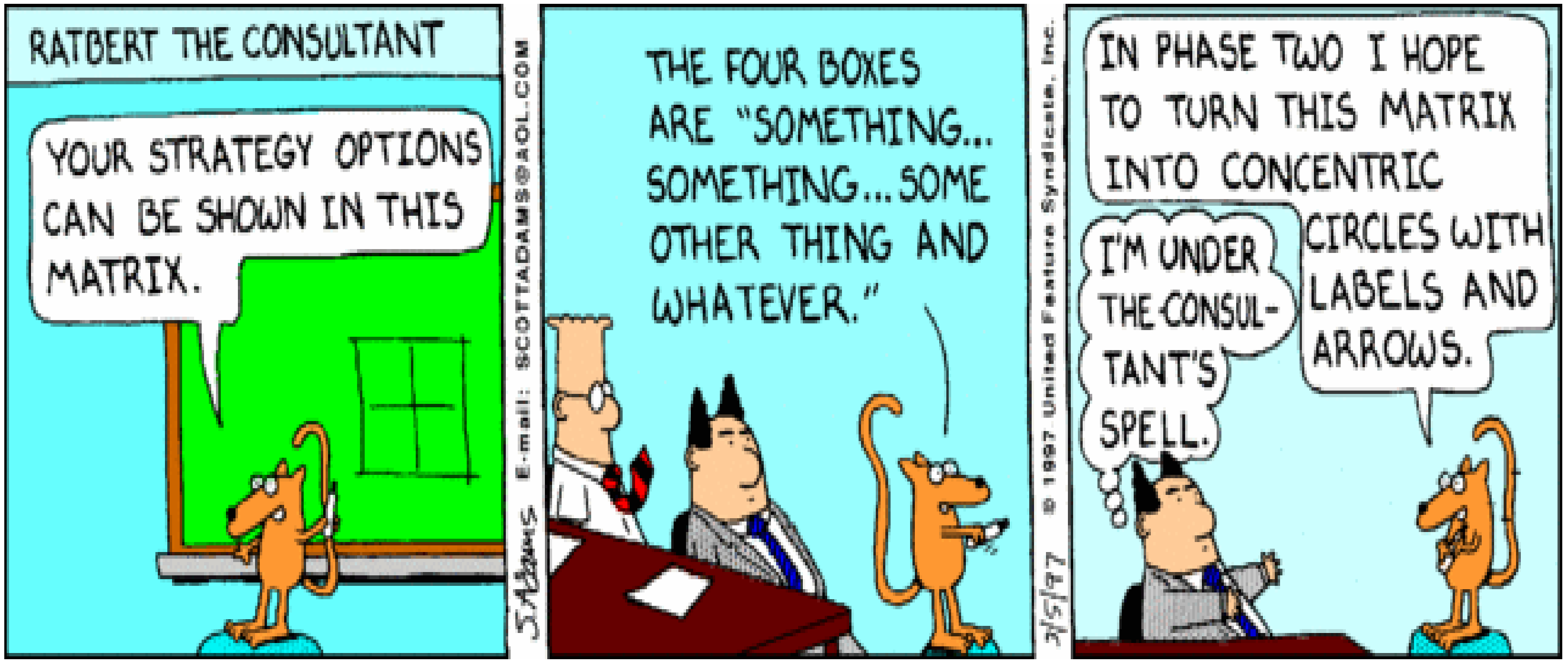


If node 5 wants to talk with node 3, what is the path to use with the least connections ?

Matrix Representation

Who doesn't love matrices ?

- The graph can be represented as a square matrix A
- If node i is directed to node j, then:
  - A[i,j] = 1 (else it is zero [or empty])
- If node i and j can both communicate with each other then:
  - A[i,j] = 1 and A[j,i]=1
- For a graph where the edges have costs, <u>the value of the cost is used instead of a one</u>


RATBERT THE CONSULTANT
YOUR STRATEGY OPTIONS CAN BE SHOWN IN THIS MATRIX.
S. Adams E-mail: SCOTTADAMS@AOL.COM
THE FOUR BOXES ARE "SOMETHING... SOMETHING... SOME OTHER THING AND WHATEVER."
3/5/97 © 1997 United Feature Syndicate, Inc.
IN PHASE TWO I HOPE TO TURN THIS MATRIX INTO CONCENTRIC CIRCLES WITH LABELS AND ARROWS.
I'M UNDER THE CONSULTANT'S SPELL.

Matrix for Previous Example Graph

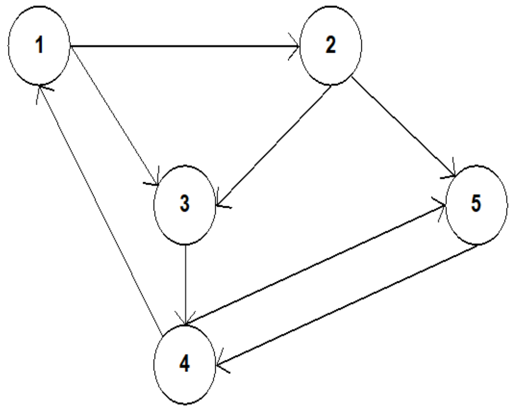
[this shows which nodes can talk directly with which other nodes, the rows are "from" and the columns are "to"]

To

From

	1	1		
		1		1
			1	
1				1
			1	

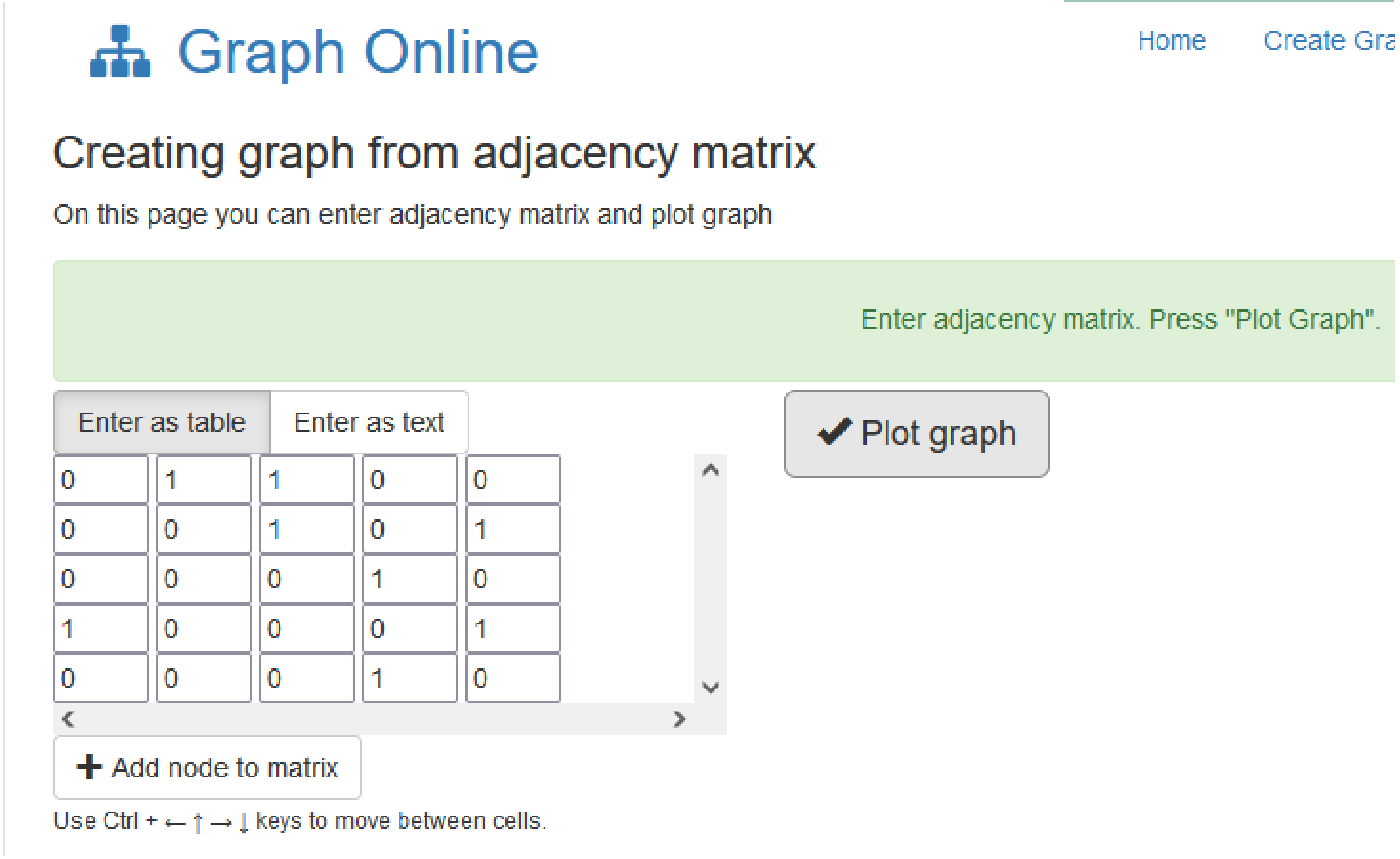
8 non-zeros in matrix, corresponding to 8 paths in graph



From 5 to 4

Draw Graph From Matrix

[https://graphonline.ru/en/create_graph_by_matrix]



Graph Online

Home Create Gra

Creating graph from adjacency matrix

On this page you can enter adjacency matrix and plot graph

Enter adjacency matrix. Press "Plot Graph".

Enter as table | Enter as text

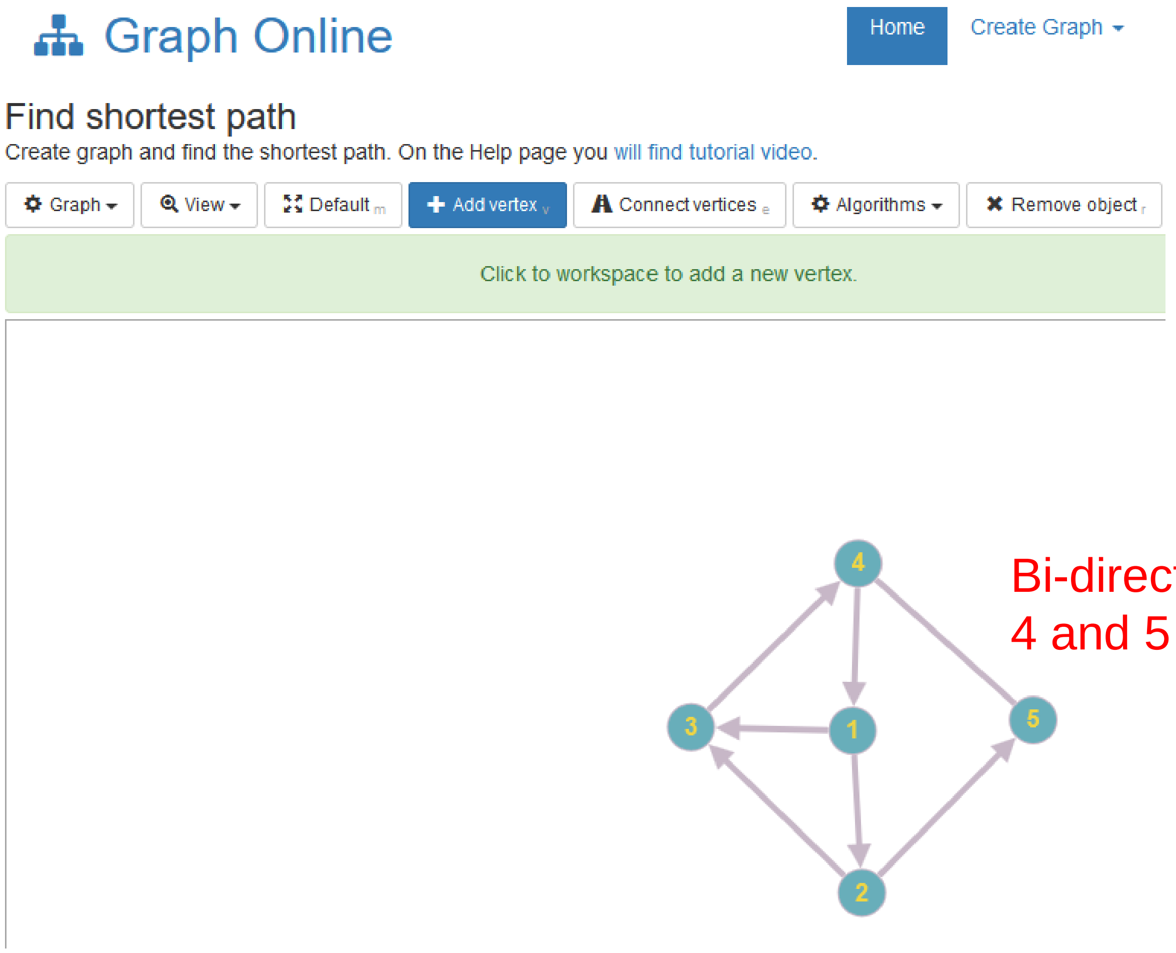
0	1	1	0	0
0	0	1	0	1
0	0	0	1	0
1	0	0	0	1
0	0	0	1	0

✔ Plot graph

+ Add node to matrix

Use Ctrl + ← ↑ → ↓ keys to move between cells.

Draw Graph From Matrix (con't)



Find shortest path

Create graph and find the shortest path. On the Help page you will find tutorial video.

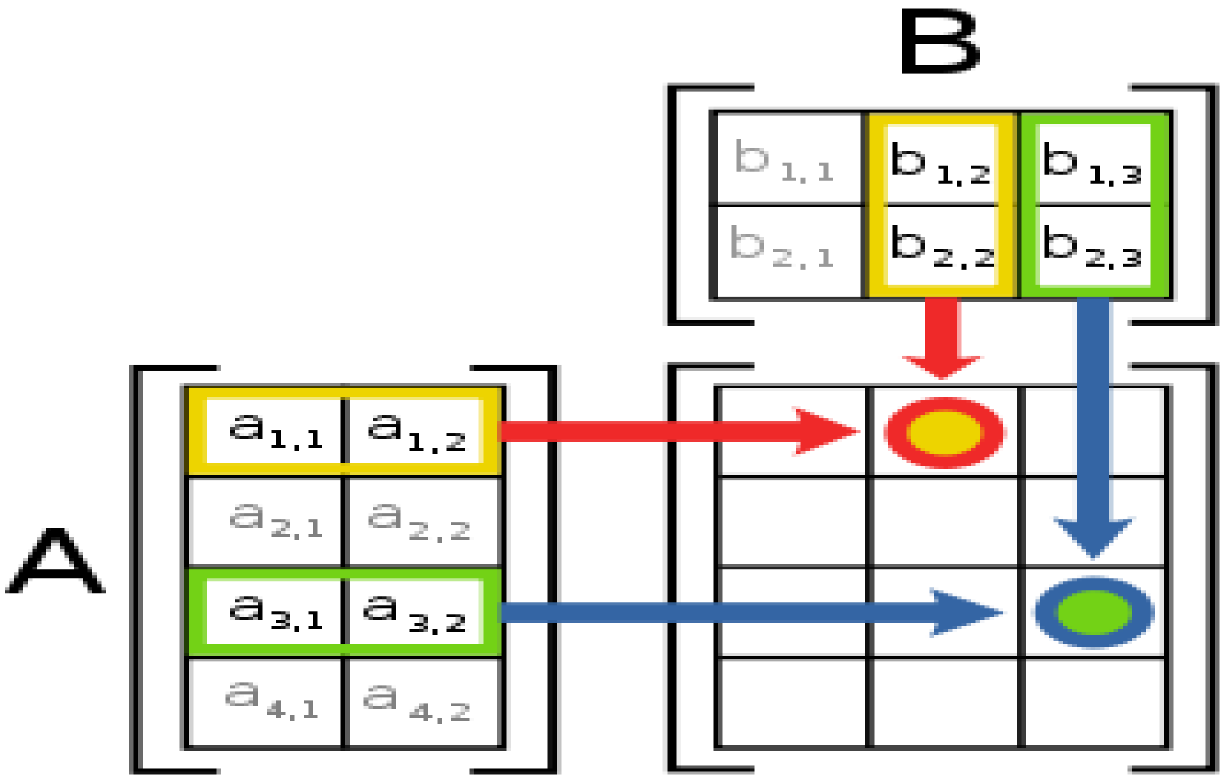
Click to workspace to add a new vertex.

Bi-directional between 4 and 5

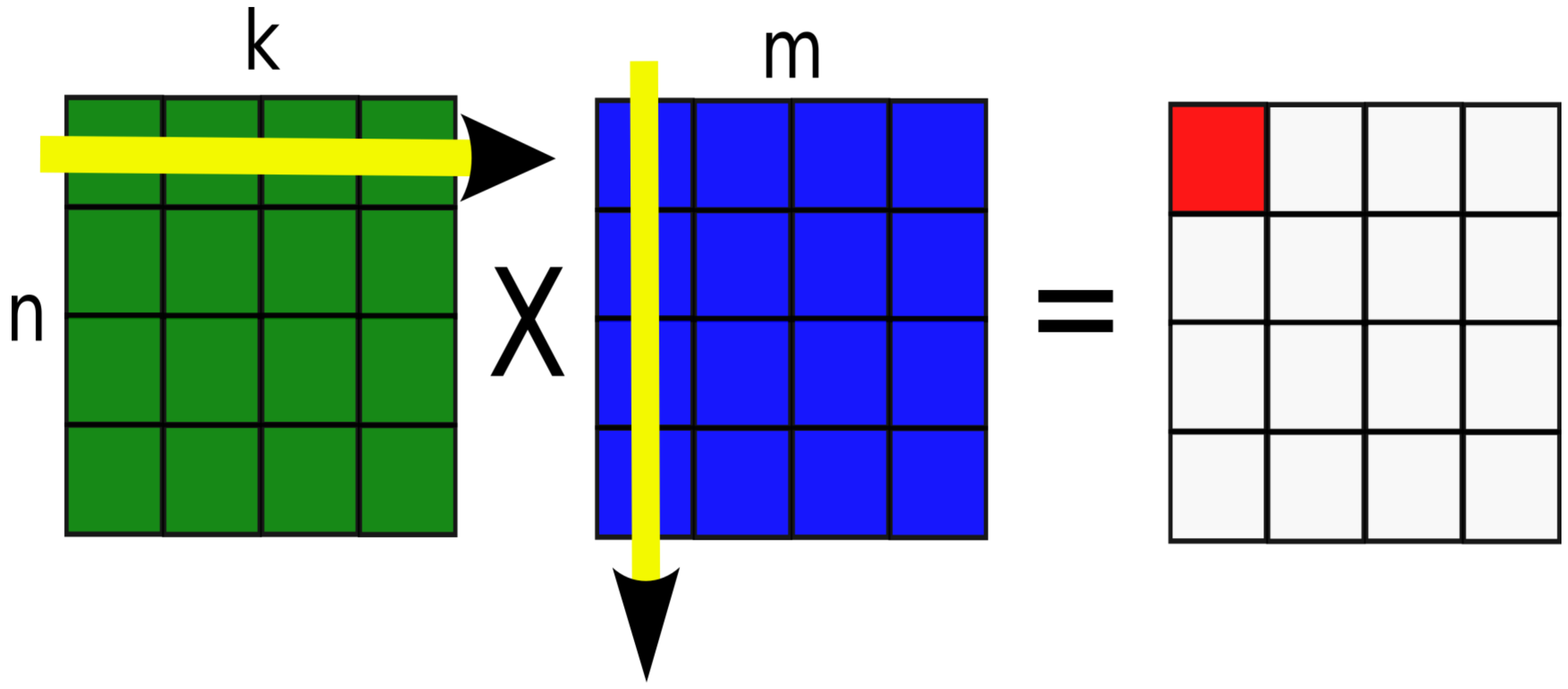
Fewest Steps thru a Network

- If we square the matrix A, we get another square matrix (B) which shows which nodes can talk to each other in two steps (by going thru another intermediate node)
- Since:
  - B[i,j] = SUM (A[i,k]*A[k,j]) (where the sum index is k)
- The number in each cell in the B matrix represents the number of two step links
  - Details on next slide →

Matrix Multiplication: A * B



Square of a Matrix: $B = A^2$

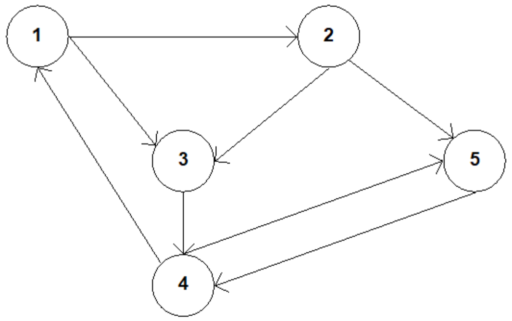


$$b_{11} = a_{11}*a_{11} + a_{12}*a_{21} + a_{13}*a_{31} + a_{14}*a_{41}$$

Meaning of Matrix Multiplication

[try all the ways to go from one point to another !]

- Matrix multiplication examines all combinations of one node related to another node
- Example, from 2 to 4:
  - B[2,4] = Row 2 times Column 4
  - B[2,4] = ∑ A[2,k]*A[k,4]
  - B[2,4] = 0*0 +   {thru 1}
  -                 0*0 +   {thru 2}
  -                 1*1 +   {thru 3}
  -                 0*0 +   {thru 4}
  -                 1*1   {thru 5}
  - B[2,4] = 2 [2 ways to go from 2 to 4]
  -                 in 2 steps



To 4

From 2

	1	1		
		1		1
			1	
1				1
			1	

Calculate the Square of this Matrix by Hand ! Hint →

	1	1		
		1		1
			1	
1				1
			1	

B[i,j] = row i times column j

	1	1		
		1		1
			1	
1				1
			1	

X

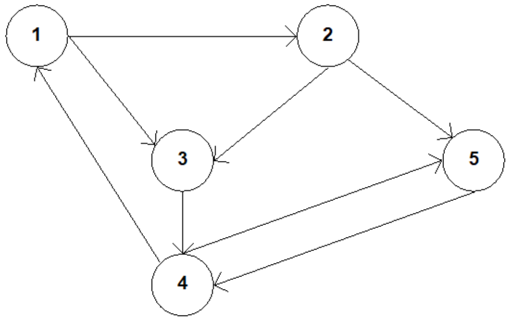
	1	1		
		1		1
			1	
1				1
			1	

- Do not look ahead !

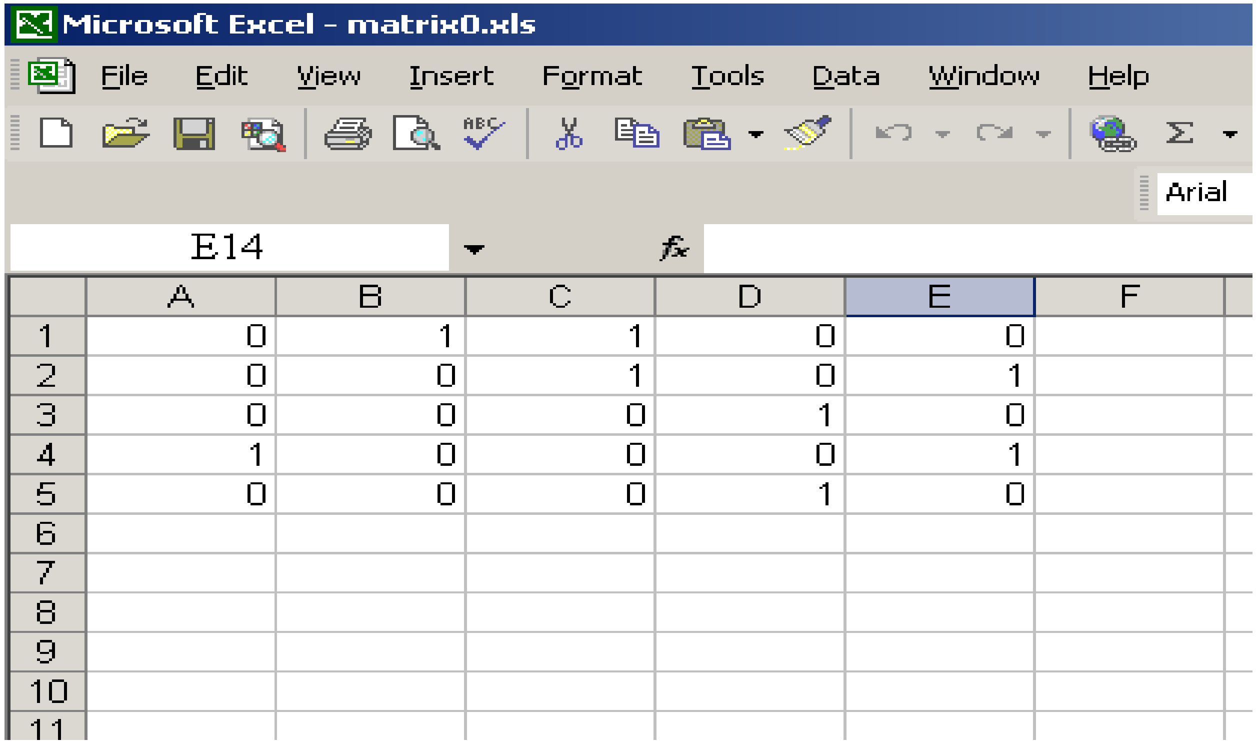
Square of Network Matrix for Example Graph

[this shows which nodes can talk with each other in 2 steps, note that there are 2 ways node 2 can talk to node 4 <u>in 2 steps</u>]

		1	1	1
			2	
1				1
	1	1	1	
1				1



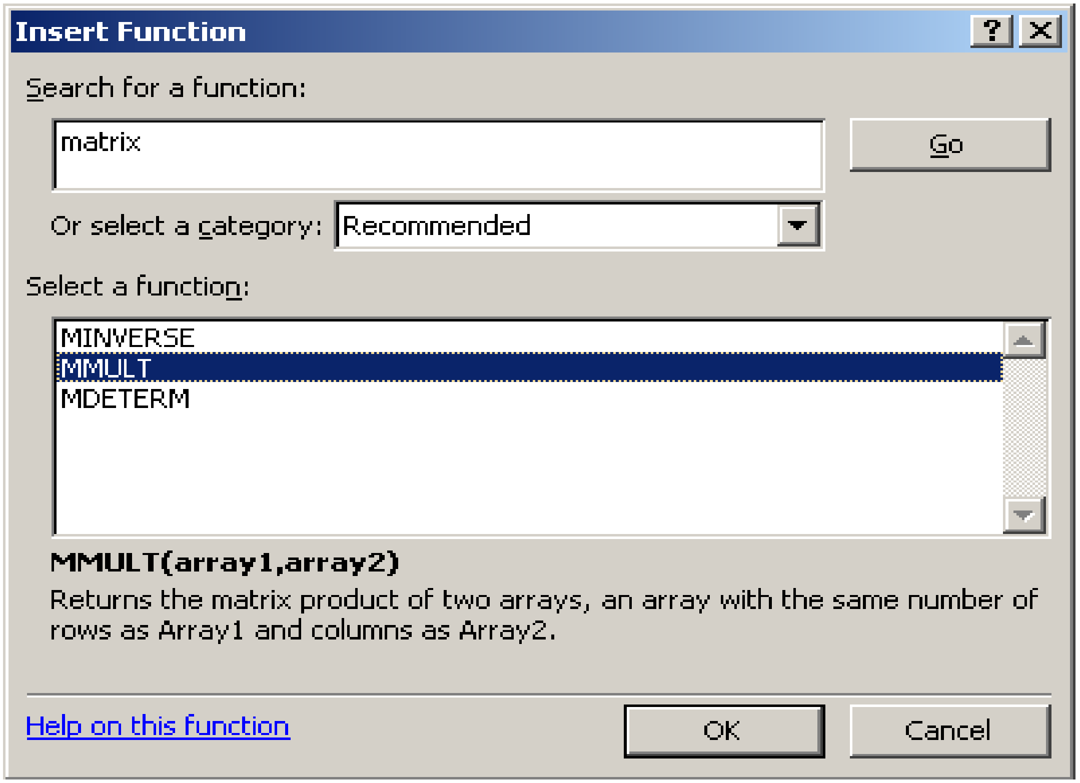
Using Excel for Matrix Multiplication


Microsoft Excel - matrix0.xls
File Edit View Insert Format Tools Data Window Help
Arial
E14

	A	B	C	D	E	F
1	0	1	1	0	0	
2	0	0	1	0	1	
3	0	0	0	1	0	
4	1	0	0	0	1	
5	0	0	0	1	0	
6						
7						
8						
9						
10						
11						

Matrix to be Squared

Function MMULT


Insert Function

Search for a function:

matrix

Or select a category: Recommended

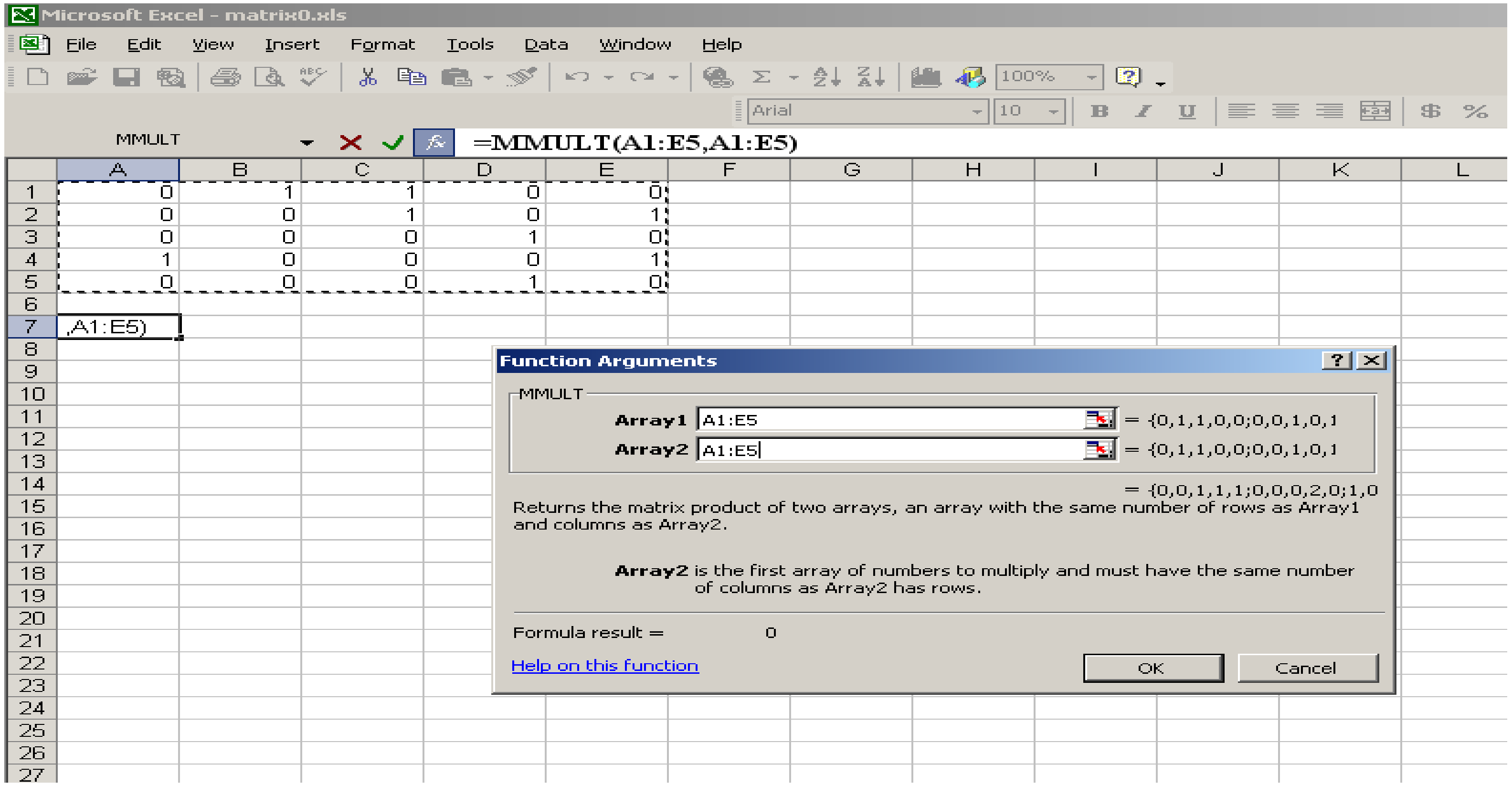
Select a function:

MINVERSE
MMULT
MDETERM

MMULT(array1,array2)

Returns the matrix product of two arrays, an array with the same number of rows as Array1 and columns as Array2.

Excel Matrix Multiplication


Microsoft Excel - matrix0.xls

=MMULT(A1:E5,A1:E5)

	A	B	C	D	E
1	0	1	1	0	0
2	0	0	1	0	1
3	0	0	0	1	0
4	1	0	0	0	1
5	0	0	0	1	0
6					
7	,A1:E5)				

Function Arguments

MMULT

Array1 A1:E5 = {0,1,1,0,0;0,0,1,0,1

Array2 A1:E5 = {0,1,1,0,0;0,0,1,0,1

= {0,0,1,1,1;0,0,0,2,0;1,0

Returns the matrix product of two arrays, an array with the same number of rows as Array1 and columns as Array2.

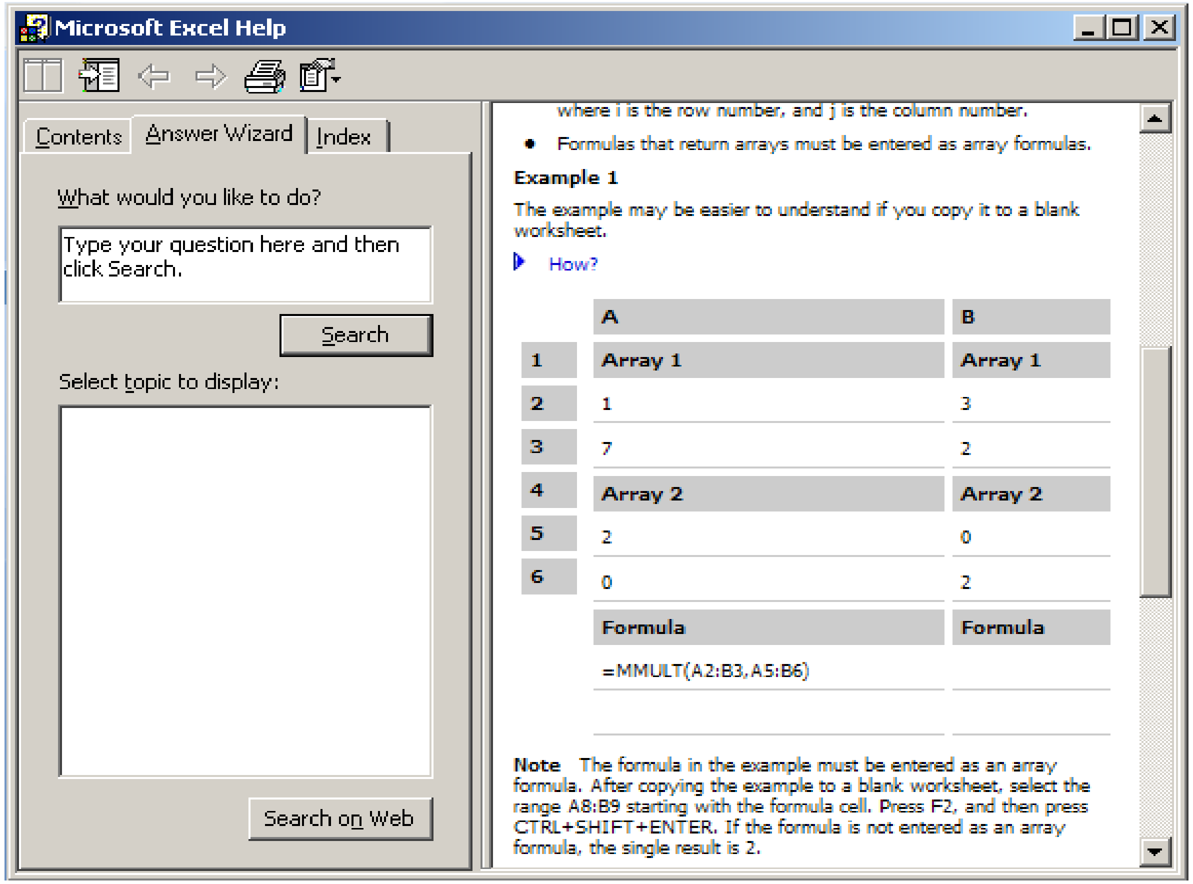
Array2 is the first array of numbers to multiply and must have the same number of columns as Array2 has rows.

Formula result = 0

Help on this function

Same range for Array1 and Array2 for square operation.

NOTE: After highlighting all the resulting matrix cells, press F2, and then press CTRL+SHIFT+ENTER. If the formula is not entered as an array formula, only a single result cell will have a value.



where i is the row number, and j is the column number.

- Formulas that return arrays must be entered as array formulas.

Example 1

The example may be easier to understand if you copy it to a blank worksheet.

	A	B
1	**Array 1**	**Array 1**
2	1	3
3	7	2
4	**Array 2**	**Array 2**
5	2	0
6	0	2
	Formula	**Formula**
	=MMULT(A2:B3,A5:B6)	

Note The formula in the example must be entered as an array formula. After copying the example to a blank worksheet, select the range A8:B9 starting with the formula cell. Press F2, and then press CTRL+SHIFT+ENTER. If the formula is not entered as an array formula, the single result is 2.

Excel Matrix Multiplication (con't)

- Click in the cell where you want the resulting matrix to appear (upper left corner cell)
- Hit the Fx (insert function button)
- Select MMULT, hit OK
- Highlight the range for Array1, hightlight the range for Array 2 (same for matrix square operation)
- Hit OK
- To expand the single value in the resulting cell: select the range for the resulting matrix (rows and columns), hit F2 (Ctrl-U on Mac), hit Ctrl+Shift+Enter

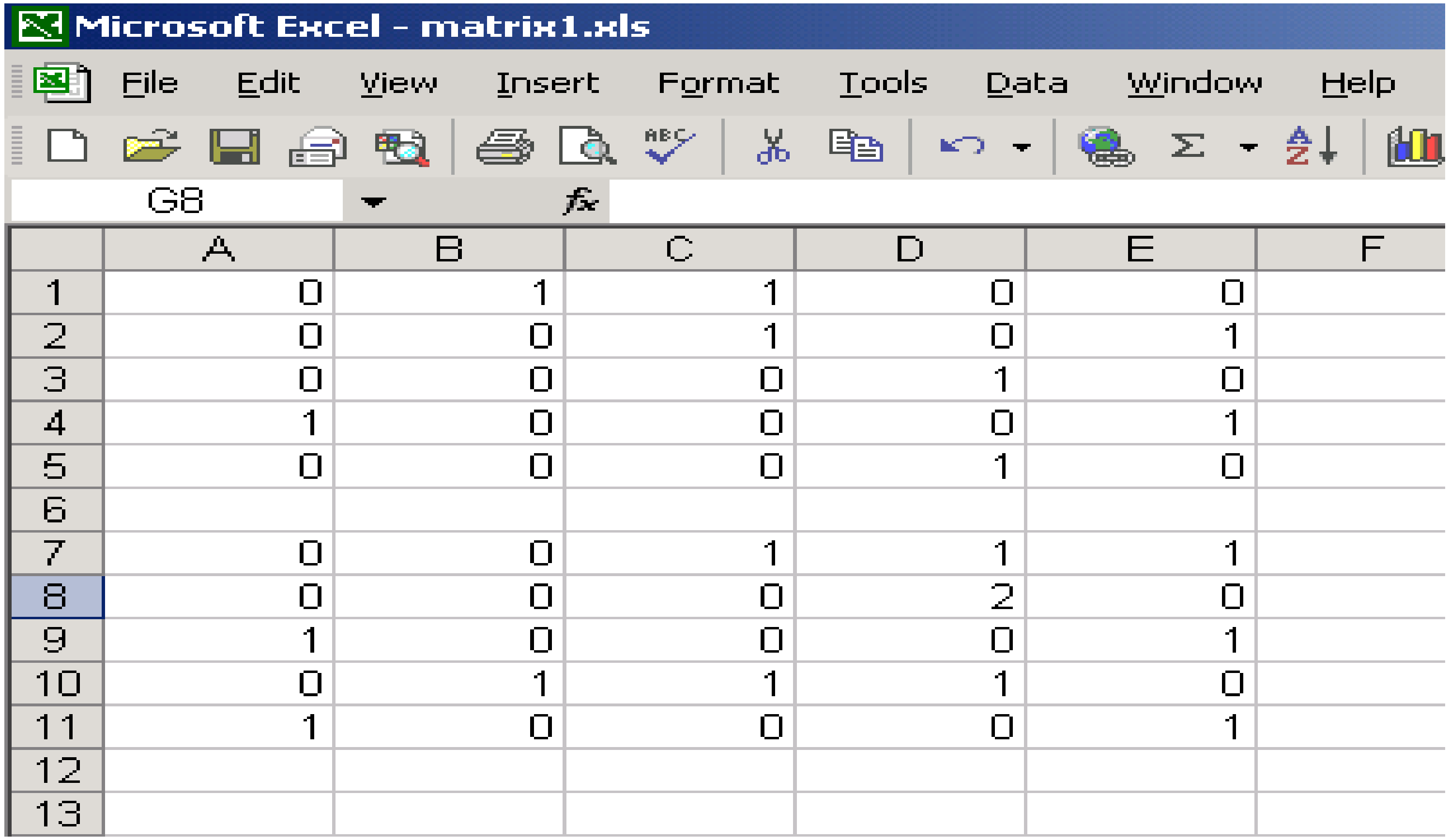
Calculate the Square of this Matrix via Excel !

	1	1		
		1		1
			1	
1				1
			1	

Click in the cell where you want the resulting matrix to appear (upper left corner cell)
Hit the Fx (insert function button)
Select MMULT, hit OK
Highlight the range for Array1, hightlight the range for Array 2 (same for square operation)
Hit OK
To expand the single value in the resulting cell: select the range for the resulting matrix (rows and columns), hit F2, hit Ctrl+Shift+Enter

- Do not look ahead !

Excel Results


Microsoft Excel - matrix1.xls
File Edit View Insert Format Tools Data Window Help
G8
fx

	A	B	C	D	E	F
1	0	1	1	0	0	
2	0	0	1	0	1	
3	0	0	0	1	0	
4	1	0	0	0	1	
5	0	0	0	1	0	
6						
7	0	0	1	1	1	
8	0	0	0	2	0	
9	1	0	0	0	1	
10	0	1	1	1	0	
11	1	0	0	0	1	
12						
13						

Communication Matrix

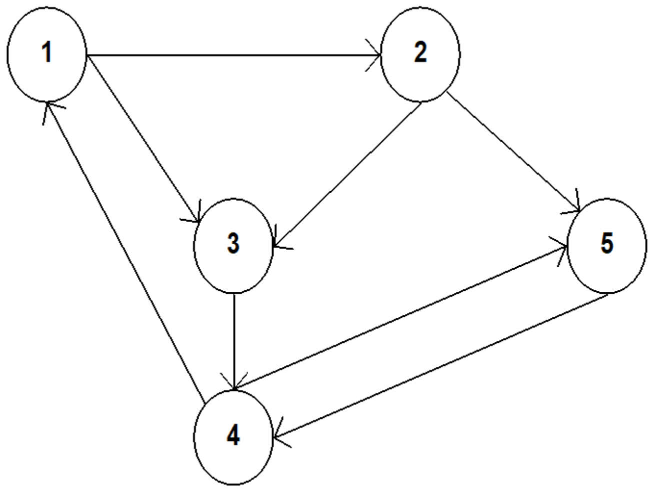
- The N’th power of the network matrix (A) will show which nodes can talk to which other nodes in N steps
- Thus if we keep taking higher powers of the network matrix (up to the size of the matrix minus 1) we can see how long it will take for any two nodes to communicate (some pairs may never communicate)
- Then we can build a “communication matrix” [C] which shows for C[i, j] how many steps it takes to go from node i to node j

Communication Matrix for Previous Example
[each cell is the least steps path from i to j]

	1	1	2	2
3		1	2	1
2	3		1	2
1	2	2		1
2	3	3	1	

Takes 3 steps to go from 2 to 1…

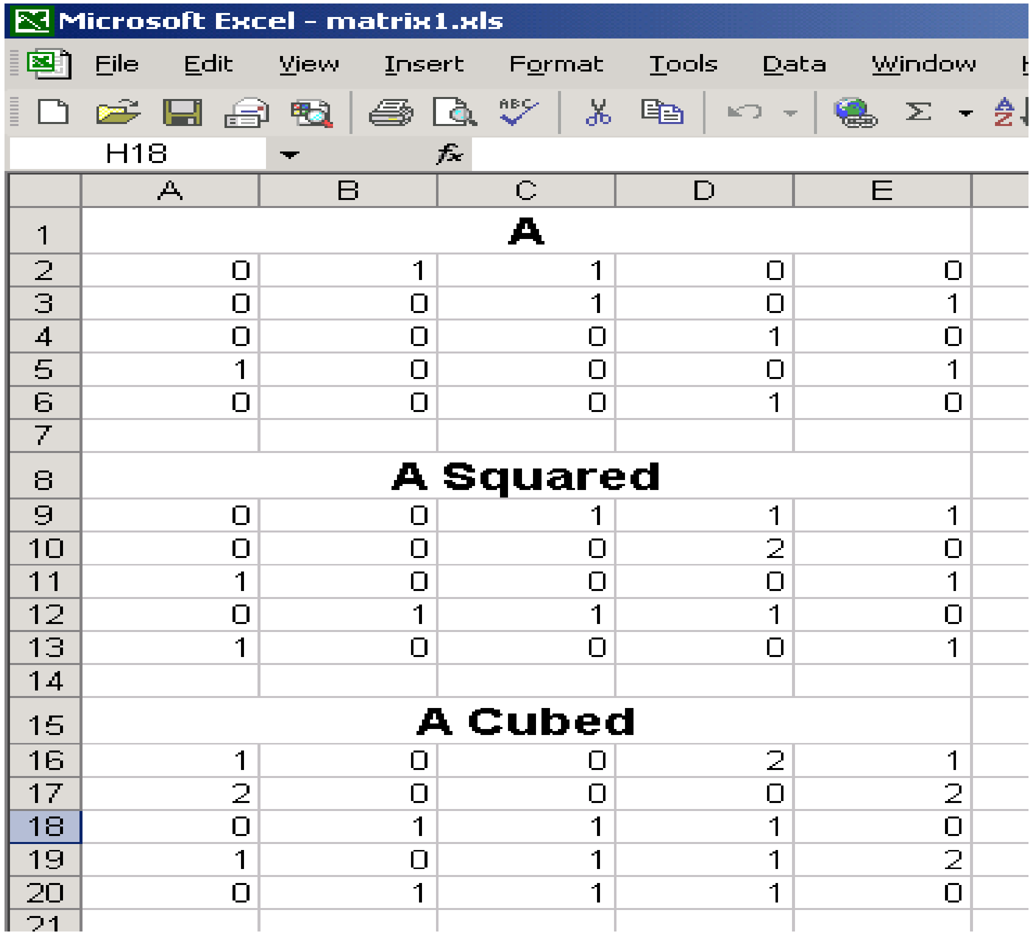
	1	1	2	2
3		1	2	1
2	3		1	2
1	2	2		1
2	3	3	1	



Building the Communications Matrix

- To get the Communication matrix you have to take at most A^N-1 powers (where N is number of nodes)
- All nodes might be able to reach each other in less steps, as was the case with the example here
- The Comm matrix starts out as the A matrix
  - then you take A^2, and any non-zeros in A^2 that are not already in the Comm matrix go in as 2's
  - then you take A^3, and any non-zeros in A^3 that are not already in the Comm matrix go in as 3's
  - and so on...

Calculation of Communication Matrix via Excel



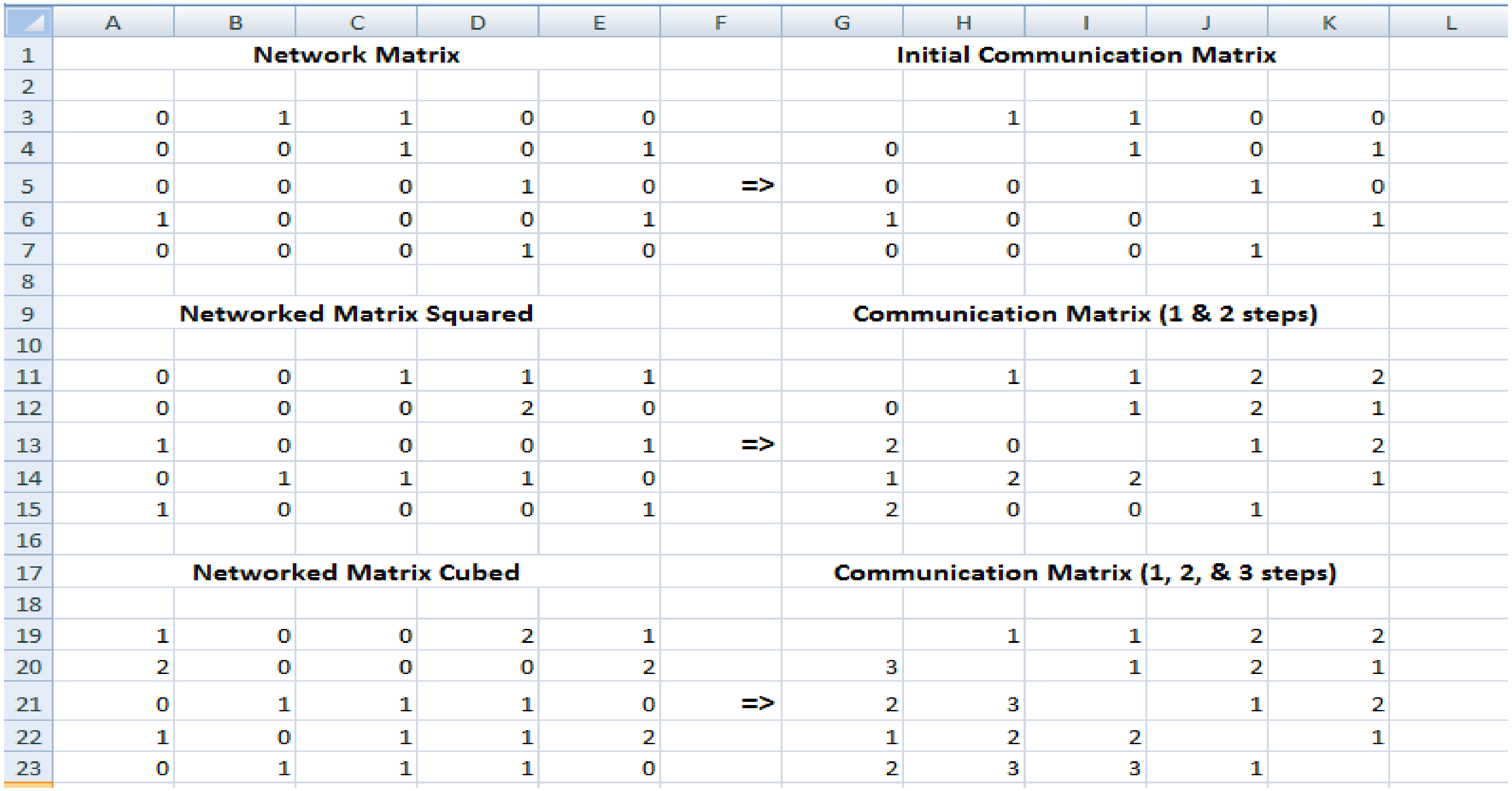
Microsoft Excel - matrix1.xls
File Edit View Insert Format Tools Data Window
H18

	A	B	C	D	E
1			A		
2	0	1	1	0	0
3	0	0	1	0	1
4	0	0	0	1	0
5	1	0	0	0	1
6	0	0	0	1	0
7					
8			A Squared		
9	0	0	1	1	1
10	0	0	0	2	0
11	1	0	0	0	1
12	0	1	1	1	0
13	1	0	0	0	1
14					
15			A Cubed		
16	1	0	0	2	1
17	2	0	0	0	2
18	0	1	1	1	0
19	1	0	1	1	2
20	0	1	1	1	0
21					

non-zeros in A^2 that are not already in the Comm matrix go in as 2's

non-zeros in A^3 that are not already in the Comm matrix go in as 3's

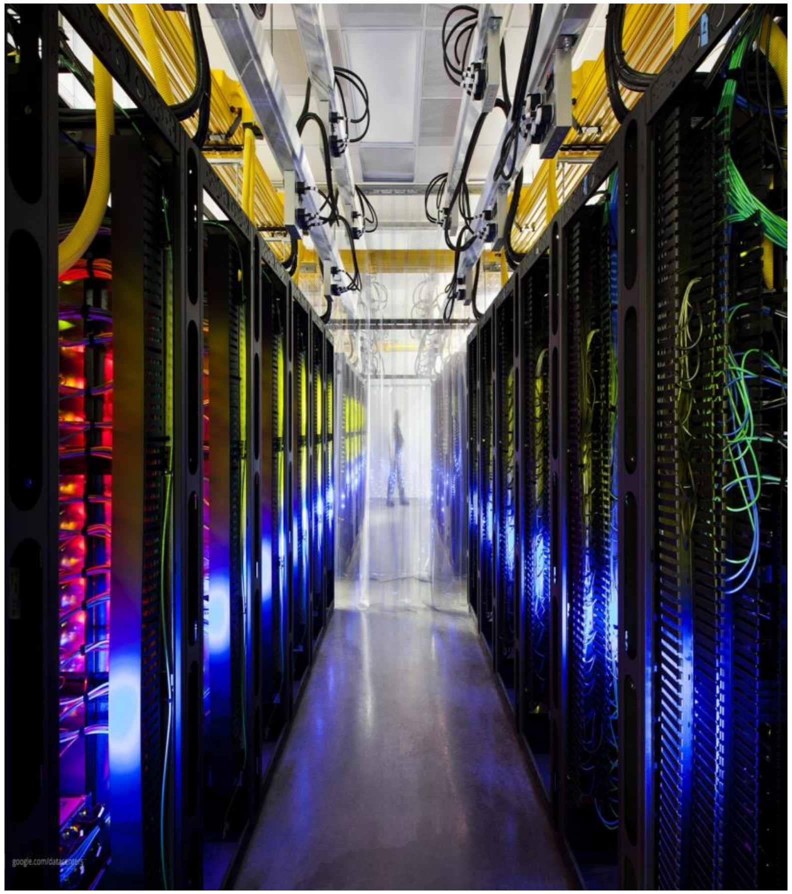
Calculation of Communication Matrix via Excel (con't)



	A	B	C	D	E	F	G	H	I	J	K	L
1	Network Matrix						Initial Communication Matrix					
2												
3	0	1	1	0	0			1	1	0	0	
4	0	0	1	0	1		0		1	0	1	
5	0	0	0	1	0	=>	0	0		1	0	
6	1	0	0	0	1		1	0	0		1	
7	0	0	0	1	0		0	0	0	1		
8												
9	Networked Matrix Squared						Communication Matrix (1 & 2 steps)					
10												
11	0	0	1	1	1			1	1	2	2	
12	0	0	0	2	0		0		1	2	1	
13	1	0	0	0	1	=>	2	0		1	2	
14	0	1	1	1	0		1	2	2		1	
15	1	0	0	0	1		2	0	0	1		
16												
17	Networked Matrix Cubed						Communication Matrix (1, 2, & 3 steps)					
18												
19	1	0	0	2	1			1	1	2	2	
20	2	0	0	0	2		3		1	2	1	
21	0	1	1	1	0	=>	2	3		1	2	
22	1	0	1	1	2		1	2	2		1	
23	0	1	1	1	0		2	3	3	1		

Communication Metadata

- Network analysis to examine telephone, text message, and computer communication metadata
- Used by Homeland Security, NSA, law enforcement, etc.
- Courts have upheld that metadata is public (no subpoena necessary)



Shortest/cheapest Route Technique

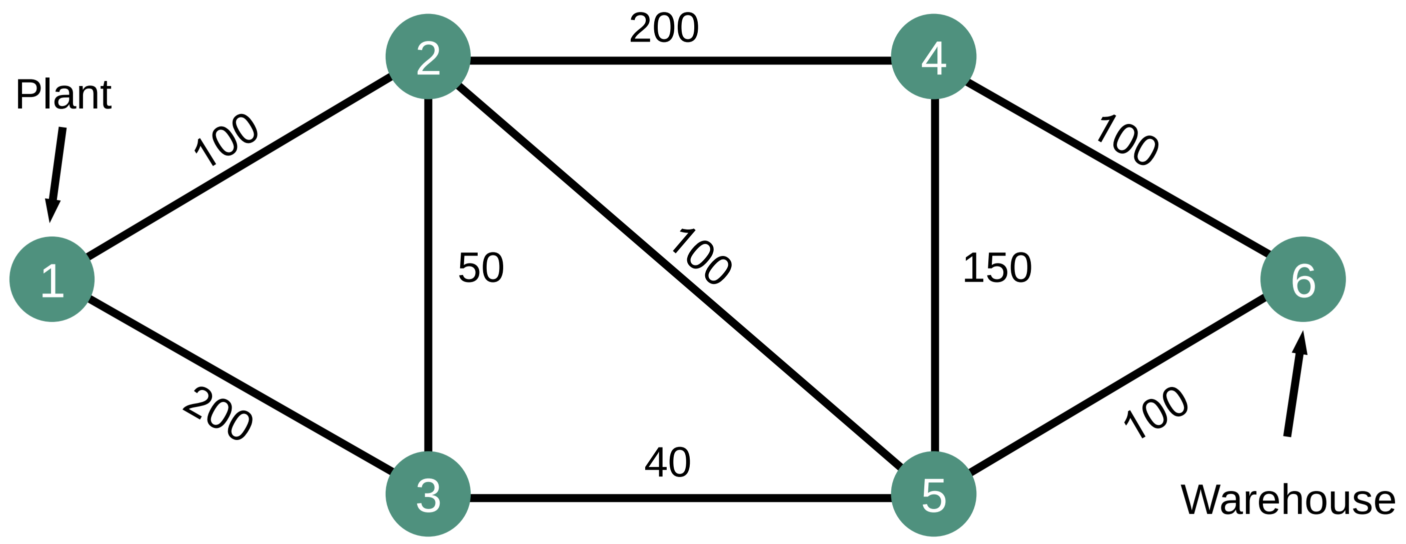
- The *shortest-route technique* finds how a person or vehicle can travel from one location to another while minimizing the total distance or time traveled
- This is different from the TSP in that there is a fixed network, one cannot travel from any point to any other point
- It finds the shortest route to a series of destinations

Shortest/cheapest Route Technique (con't)

- A company transports items from the factory to the warehouse
- They would like to find the route with the shortest distance
- The road network is shown in the following figure

Shortest-Route Technique (con't)

- Roads (with mileage) from plant to warehouse

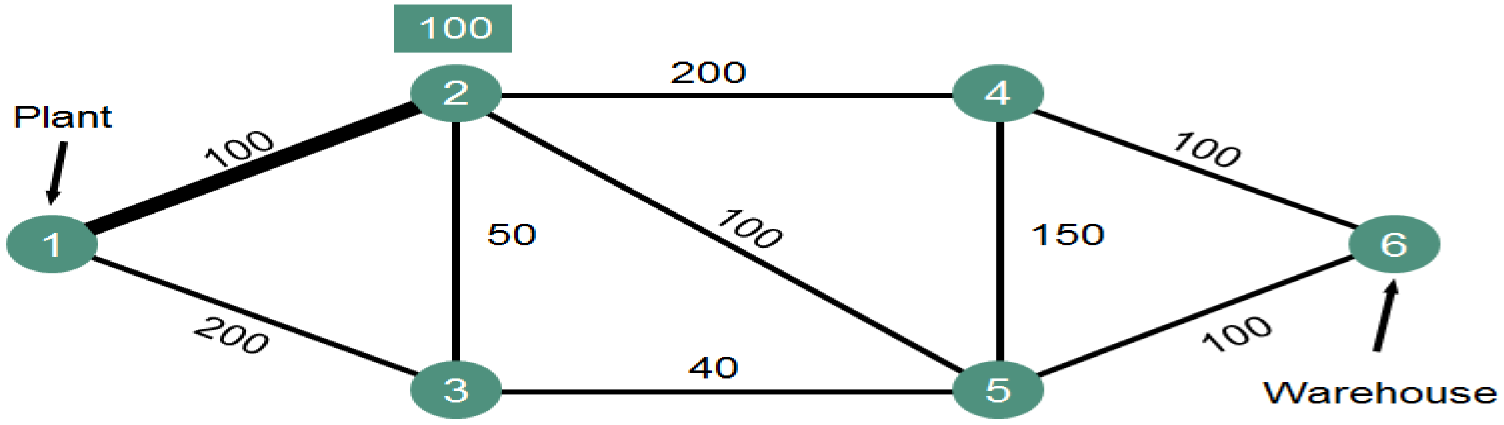


Shortest-Route Technique (con't)

- Steps of the shortest-route heuristic technique:
  1. Find the nearest node to the origin (plant). Put the distance in a box by the node.
  2. Find the next-nearest node to the origin and put the distance in a box by the node. Several paths may have to be checked to find the nearest node.
  3. Repeat this process until you have gone through the entire network. The last distance at the ending node will be the distance of the shortest route.

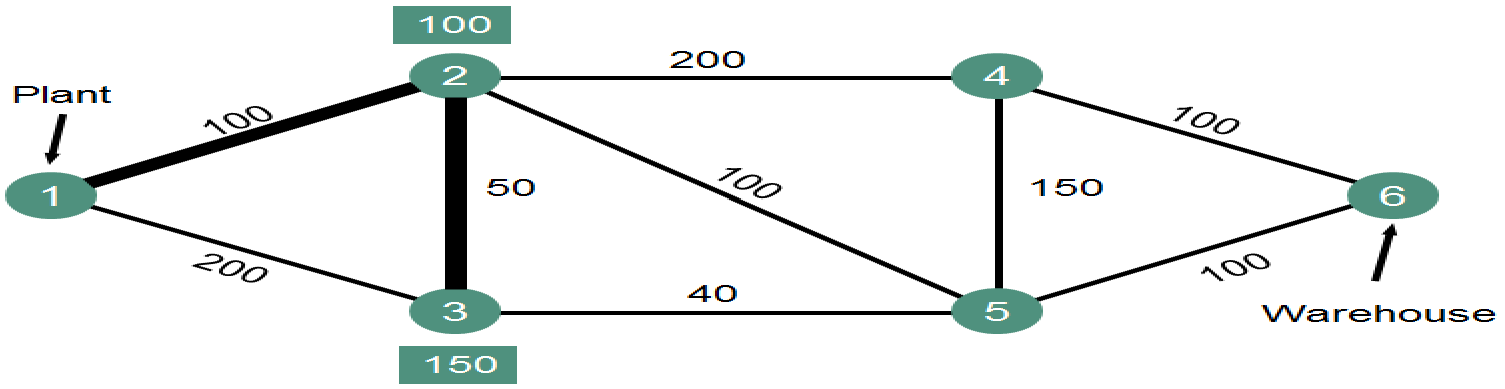
Shortest-Route Technique (con't)

- We can see that the nearest node to the plant is node 2
- We connect these two nodes
- After investigation, we find node 3 is the next nearest node but there are two possible paths
- The shortest path is 1–2–3 with a distance of 150



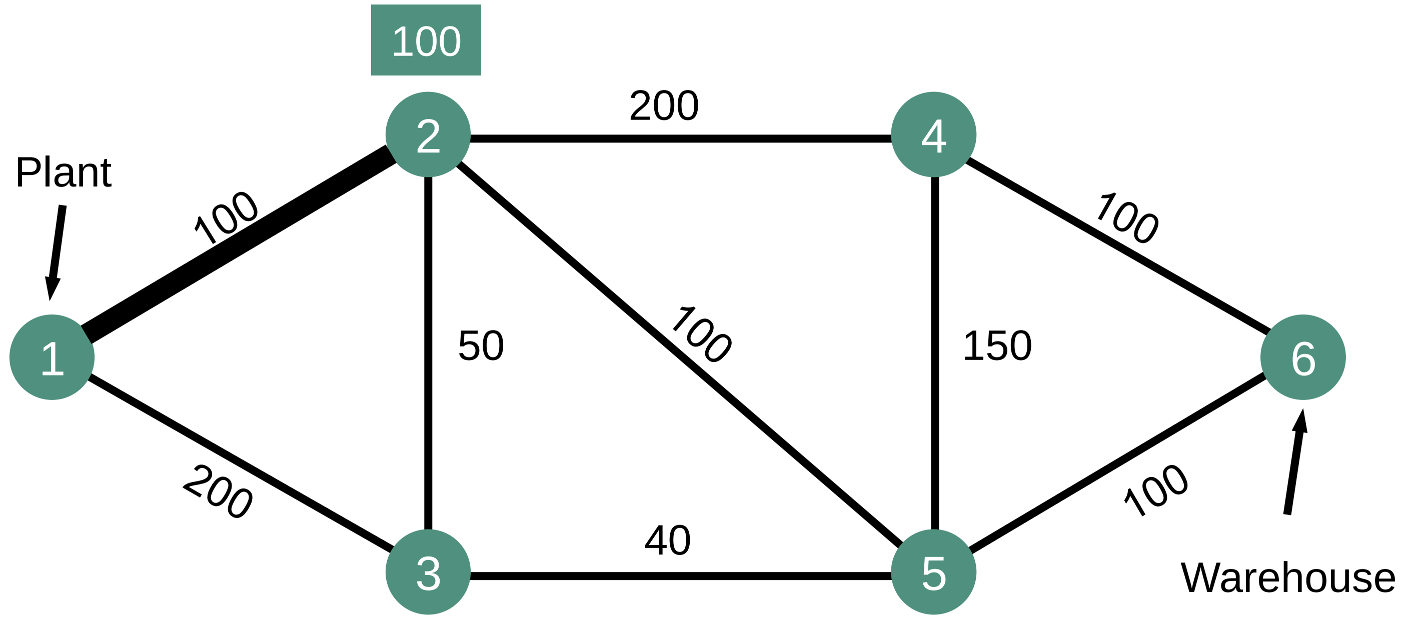
Shortest-Route Technique (con't)

- We repeat the process and find the next node is node 5 by going through node 3
- The next nearest node is either 4 or 6, and 6 turns out to be closer
- The shortest path is 1–2–3–5–6 with a distance of 290 miles



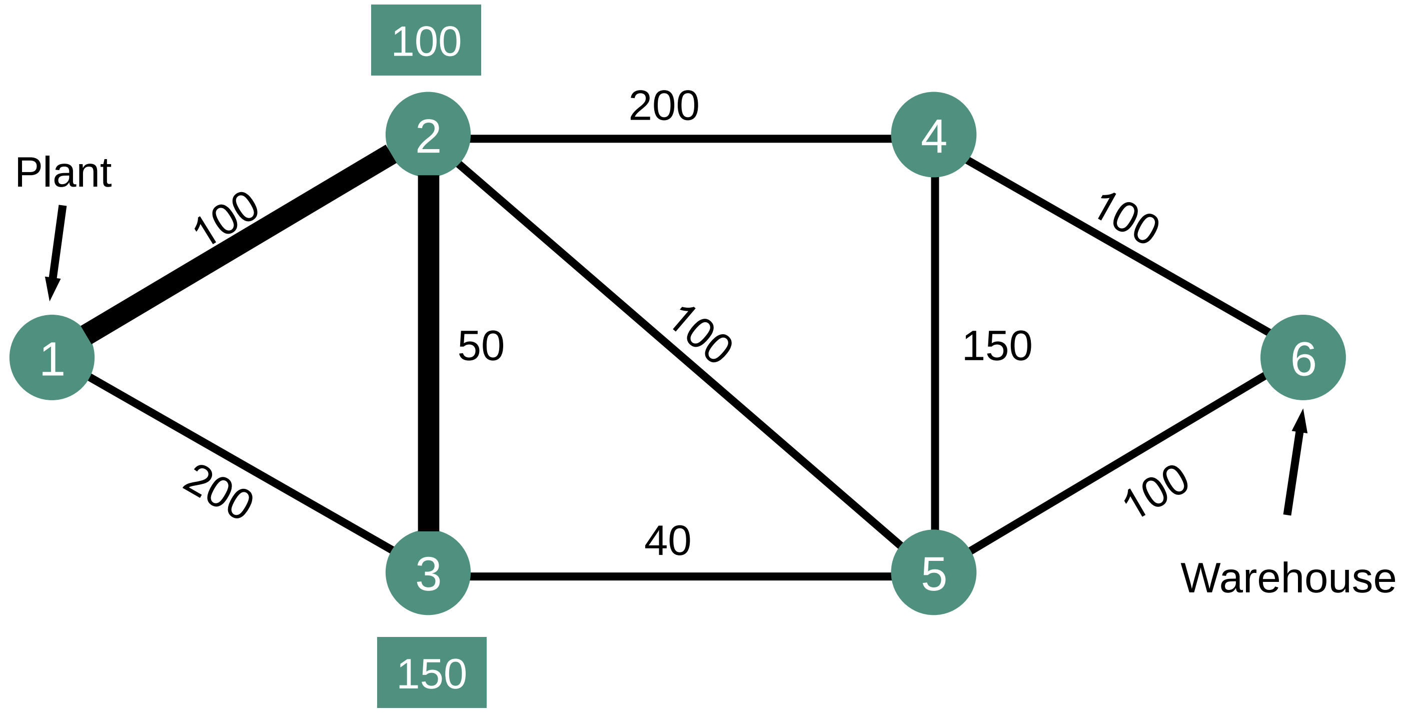
Shortest-Route Technique (con't)

- First iteration



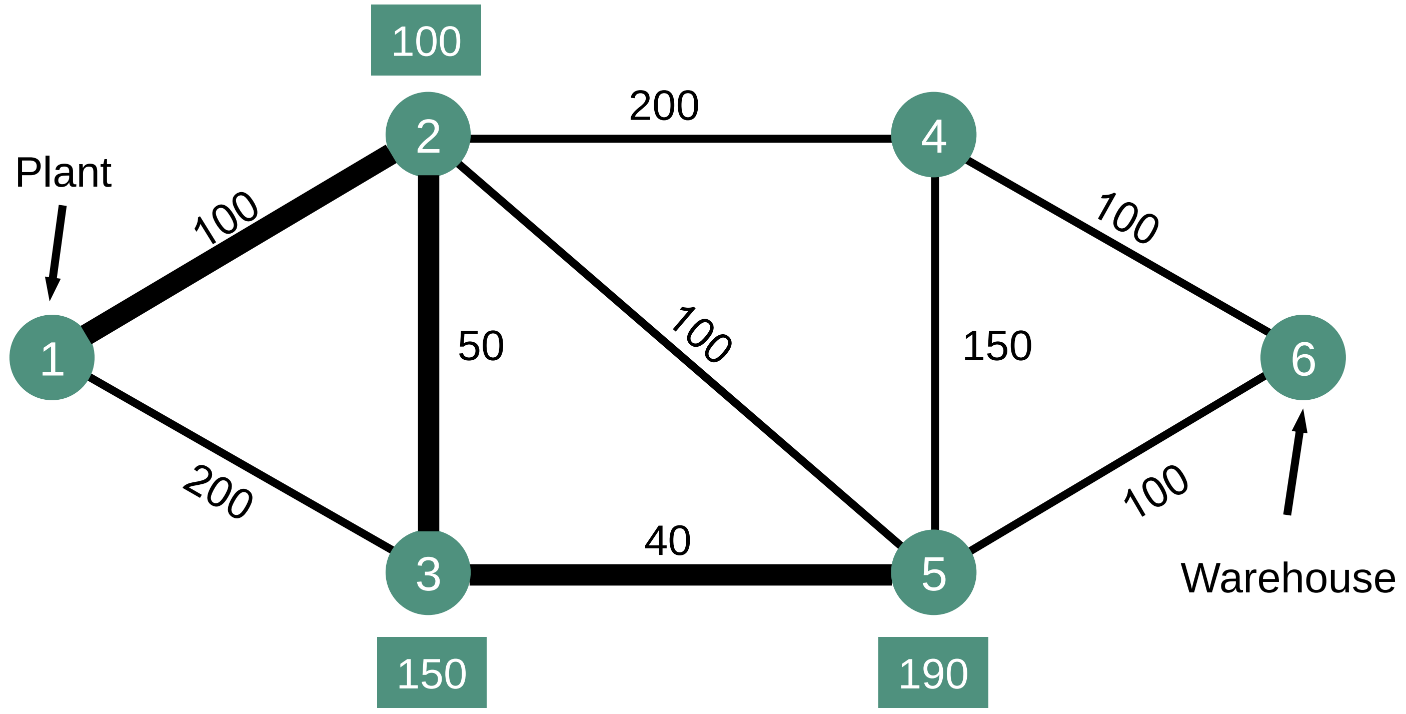
Shortest-Route Technique (con't)

- Second iteration



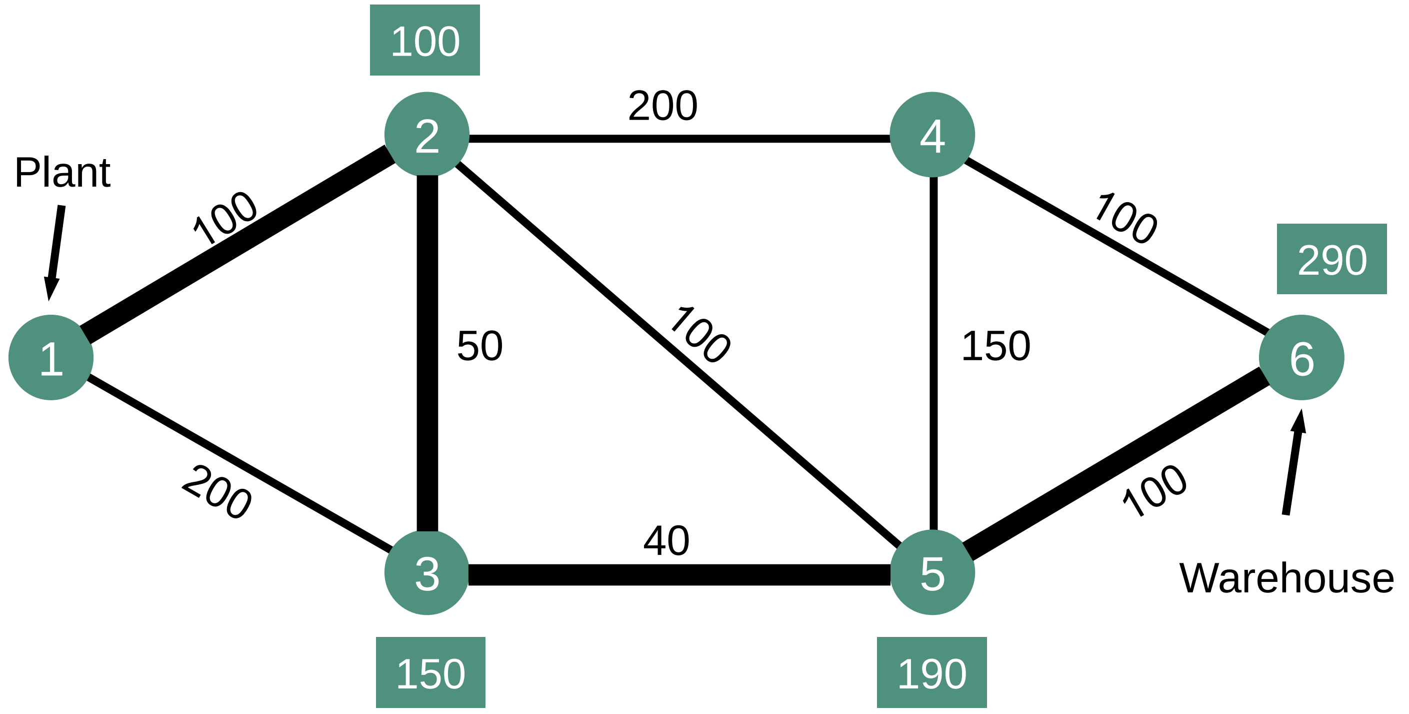
Shortest-Route Technique (con't)

- Third iteration

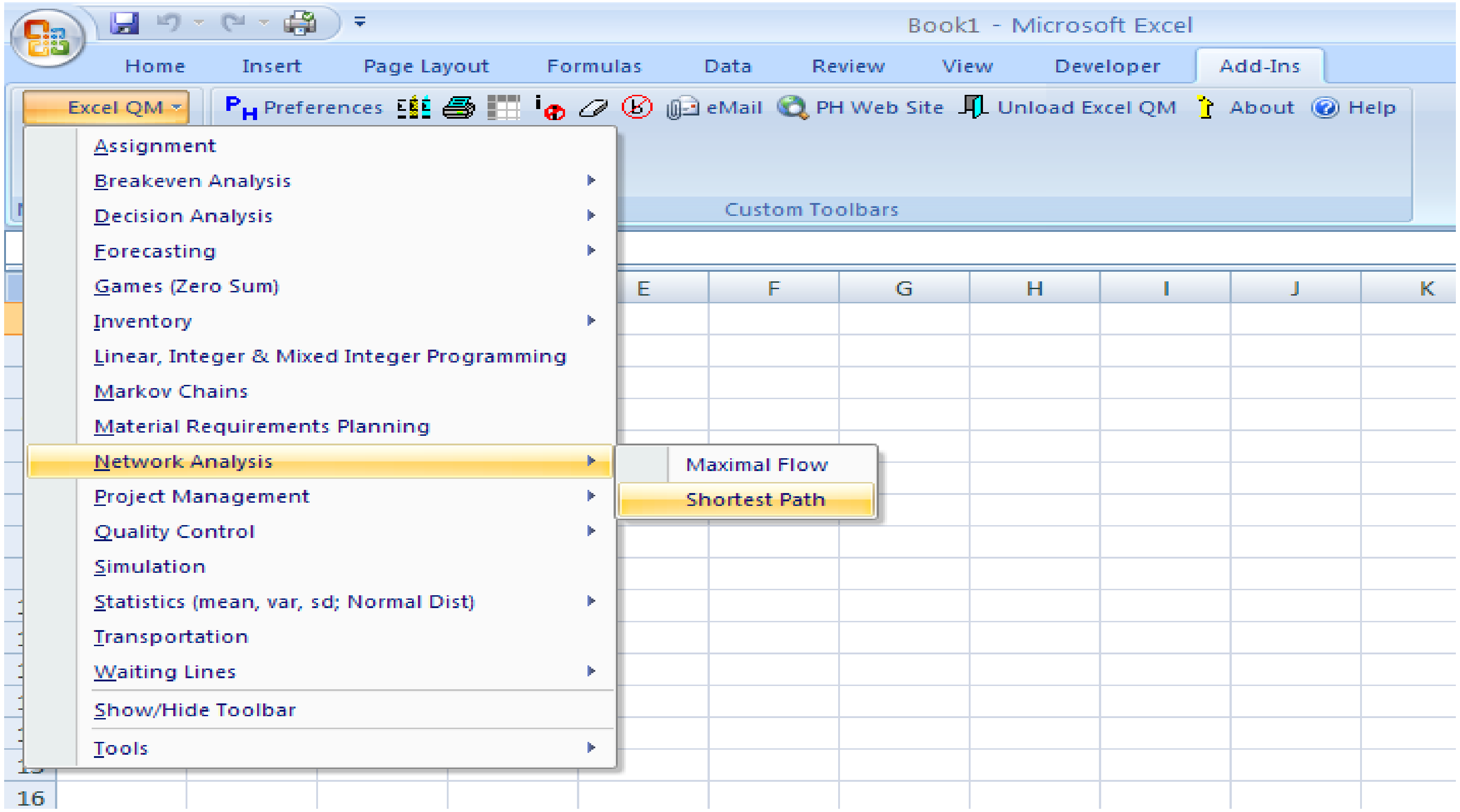


Shortest-Route Technique (con't)

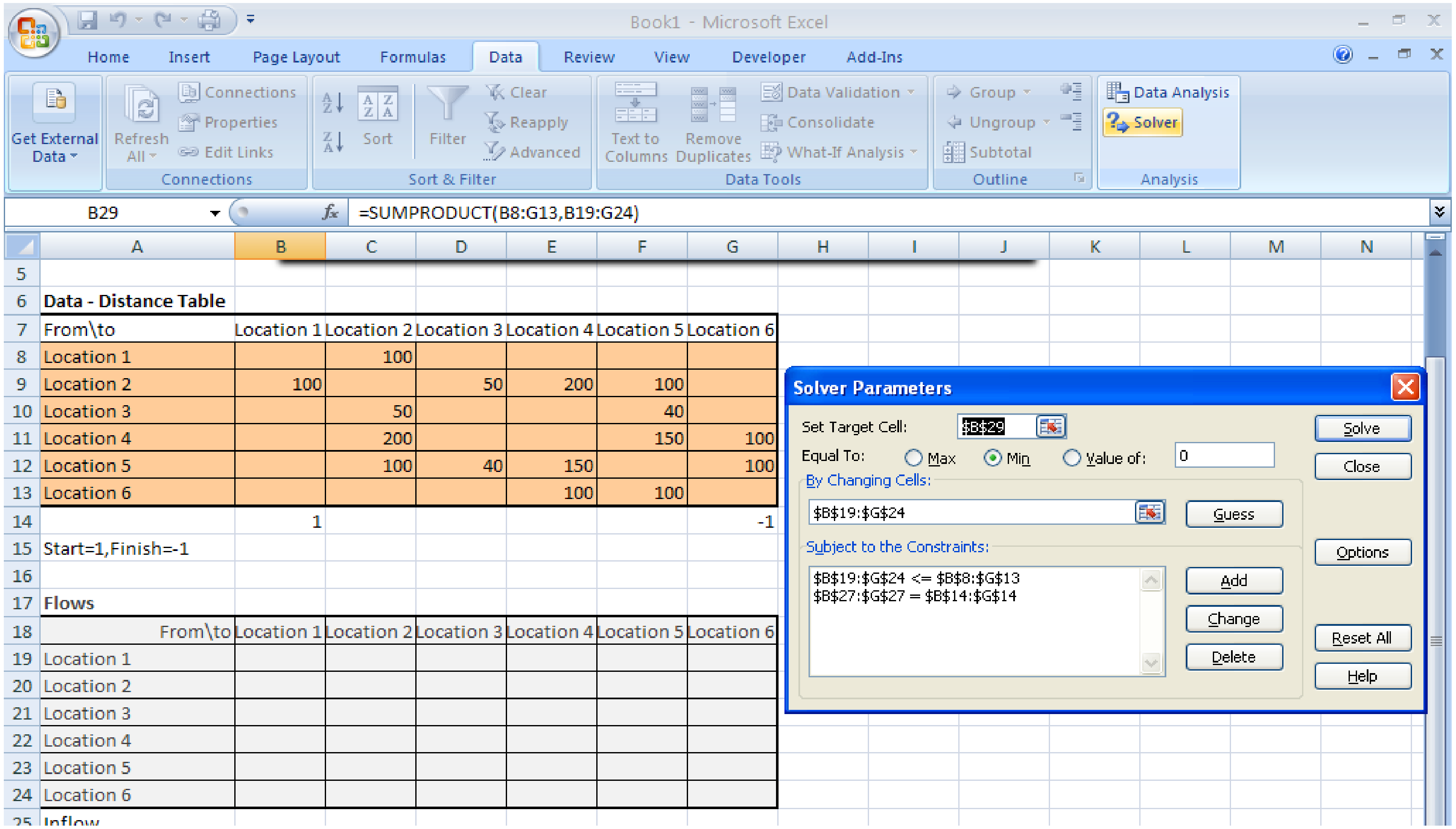
- Fourth and final iteration



Shortest-Route Technique via LP (solver) in Excel QM



Shortest-Route Technique (con't)



B29 =SUMPRODUCT(B8:G13,B19:G24)

	A	B	C	D	E	F	G
5							
6	**Data - Distance Table**						
7	From\to	Location 1	Location 2	Location 3	Location 4	Location 5	Location 6
8	Location 1		100				
9	Location 2	100		50	200	100	
10	Location 3		50			40	
11	Location 4		200			150	100
12	Location 5		100	40	150		100
13	Location 6				100	100	
14		1					-1
15	Start=1,Finish=-1						
16							
17	**Flows**						
18	From\to	Location 1	Location 2	Location 3	Location 4	Location 5	Location 6
19	Location 1						
20	Location 2						
21	Location 3						
22	Location 4						
23	Location 5						
24	Location 6						
25	Inflow						

Solver Parameters

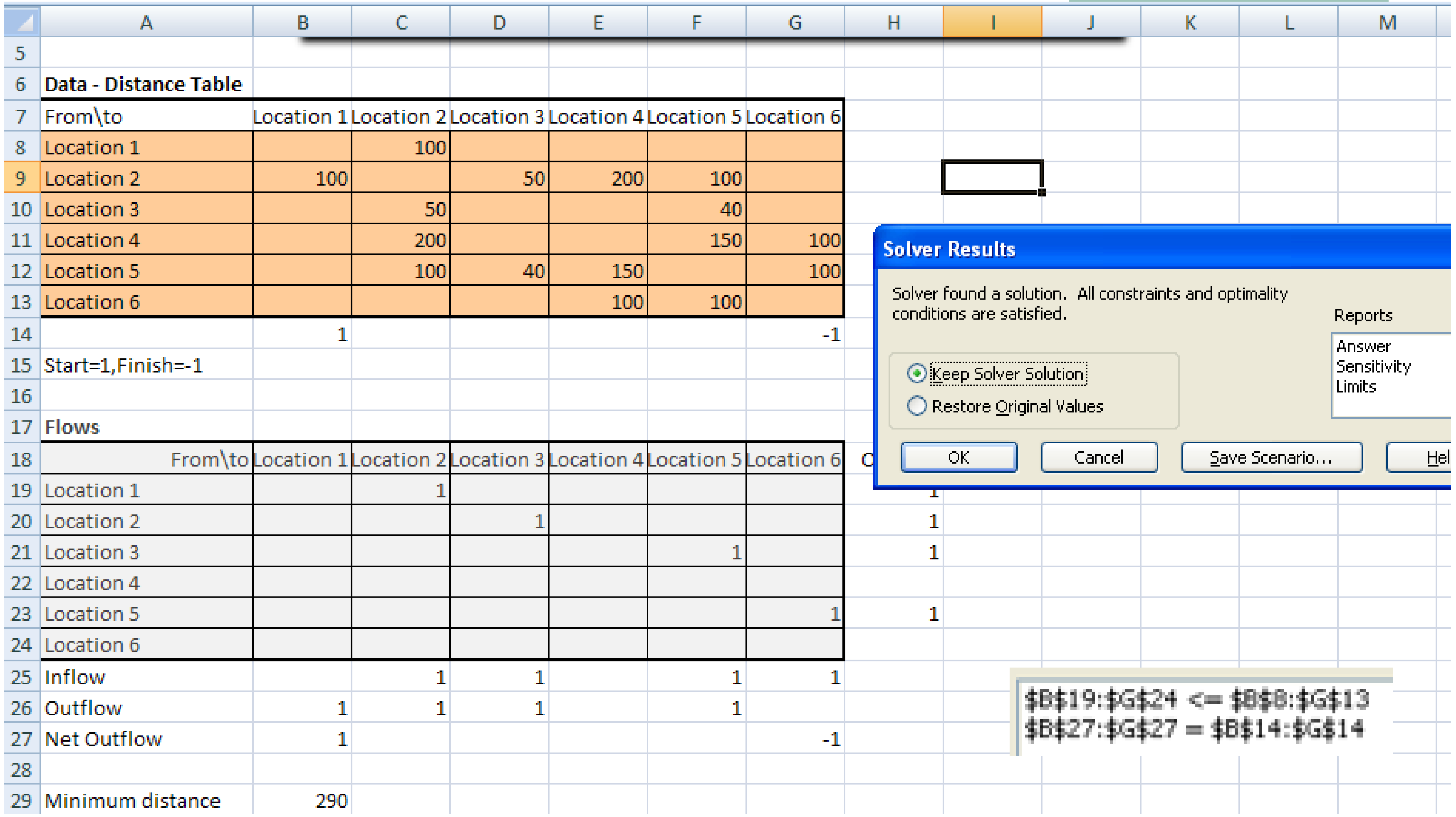
Set Target Cell: $B$29

Equal To: Max [✓ Min] Value of: 0

By Changing Cells: $B$19:$G$24

Subject to the Constraints:
$B$19:$G$24 <= $B$8:$G$13
$B$27:$G$27 = $B$14:$G$14

Shortest-Route Technique (con't)



Data - Distance Table

From\to	Location 1	Location 2	Location 3	Location 4	Location 5	Location 6
Location 1		100				
Location 2	100		50	200	100	
Location 3		50			40	
Location 4		200			150	100
Location 5		100	40	150		100
Location 6				100	100	
	1					-1
Start=1,Finish=-1						

Flows

From\to	Location 1	Location 2	Location 3	Location 4	Location 5	Location 6	C
Location 1		1					1
Location 2			1				1
Location 3					1		1
Location 4							
Location 5						1	1
Location 6							
Inflow		1	1		1	1	
Outflow	1	1	1		1		
Net Outflow	1					-1	
Minimum distance	290						

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

[● Keep Solver Solution] ○ Restore Original Values

Reports: Answer, Sensitivity, Limits

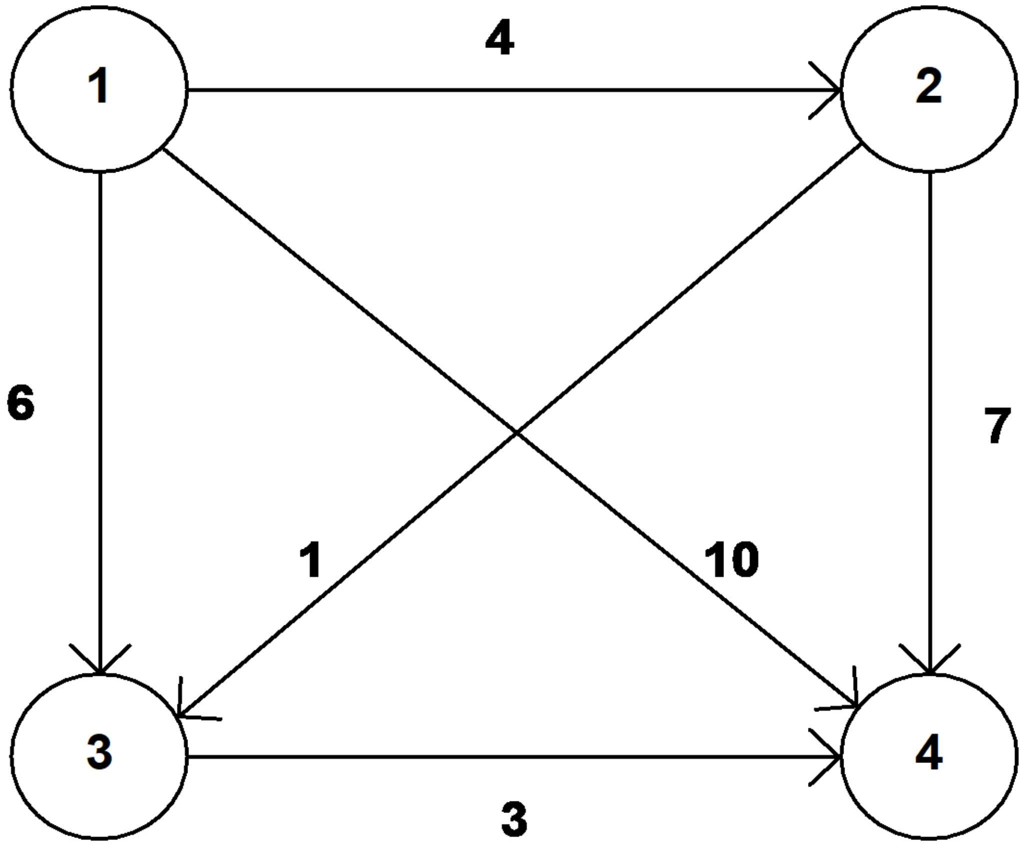
OK | Cancel | Save Scenario... | Hel

$B$19:$G$24 <= $B$8:$G$13
$B$27:$G$27 = $B$14:$G$14

Matrix Solution to Cheapest Routes

- Shortest route problems can also be found by matrix techniques as well as via the shortest route heuristic and via linear programming

Example Network with Costs (or distance) [4 nodes and 6 edges]



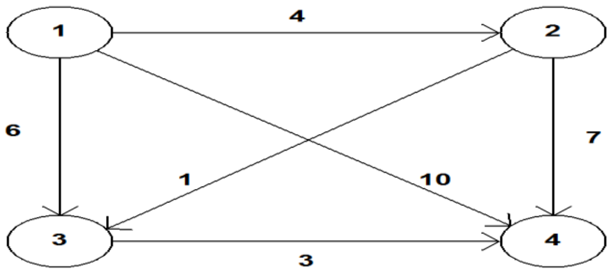
We need to find the least cost route from node 1 to node 4.

What is that route ?

Matrix Representation of Cost Network

[One Step Costs] {i.e. 1 to 4 directly cost 10}

	4	6	10
		1	7
			3



Least Cost thru a Network

- If we “square@” the matrix A, we get another square matrix (B) which shows the cost to move from node to node in two steps (by going thru another [1] intermediate node)
- The square@ operation (“minimum of squares”):
  - B[i,j] = MIN (A[i,k]+A[k,j])
    - (where the min index is k)

Matrix Minimum of Squares Operation

- The "minimum of squares" operation is just like matrix multiplication except we use the minimum operator instead of the sum operator (and add instead of multiply the edges, where both terms are non-zero)
- The number in each cell in the B matrix represents the least cost of any two step links between those two nodes

Exercise: Calculate A @A

	4	6	10
		1	7
			3

@

	4	6	10
		1	7
			3

B[i,j] = MIN (A[i,k]+A[k,j])

- Do not look ahead !

B[i,j] = MIN (A[i,k]+A[k,j])

	4	6	10
		1	7
			3

@

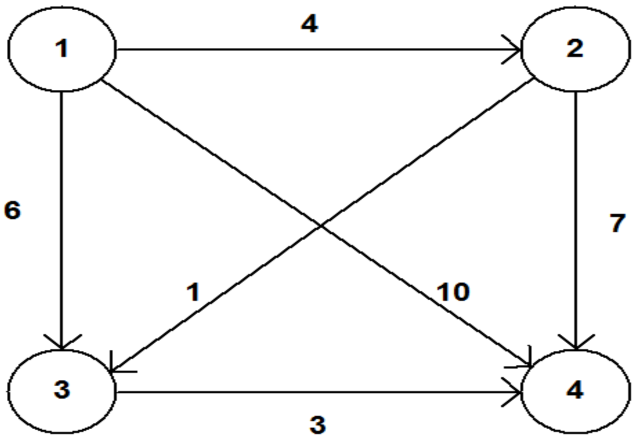
	4	6	10
		1	7
			3

$$B_{1,3} = \min(A_{1,1} + A_{1,3};\ A_{1,2} + A_{2,3};\ A_{1,3} + A_{3,3};\ A_{1,4} + A_{4,3})$$
$$= \min(NA ;\ 5 ;\ NA ;\ NA)$$
$$= 5$$

Min Two Step Costs (A @ A = B)

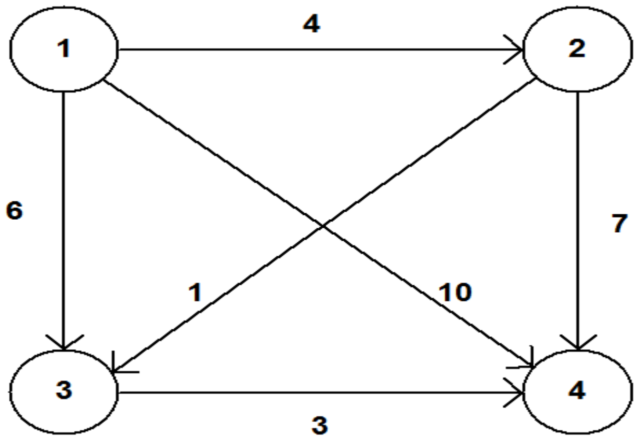
[the min is 9 from 1 to 4]

		5	9
			4



Min Three Step Costs (A @ B = C)

			8



(i.e. the least cost 3 step move from 1 to 4 is at a cost of 8)

Least Cost thru a Network (con't)

- After we do N-1 @ operations (for N nodes), then we can look thru each of the resulting N-1 matrices to find the least cost move between any two points, since the least cost move might be a 1 step move, 2 step move, …, or a N-1 step move
- If we save the "detail" of the @ function in separate matrices for each steps, we can save the actual routes that give us those minimum costs
- The order of computation here is N to the 4'th power, since it takes N cubed operations to do the square@ operation between 2 matrices

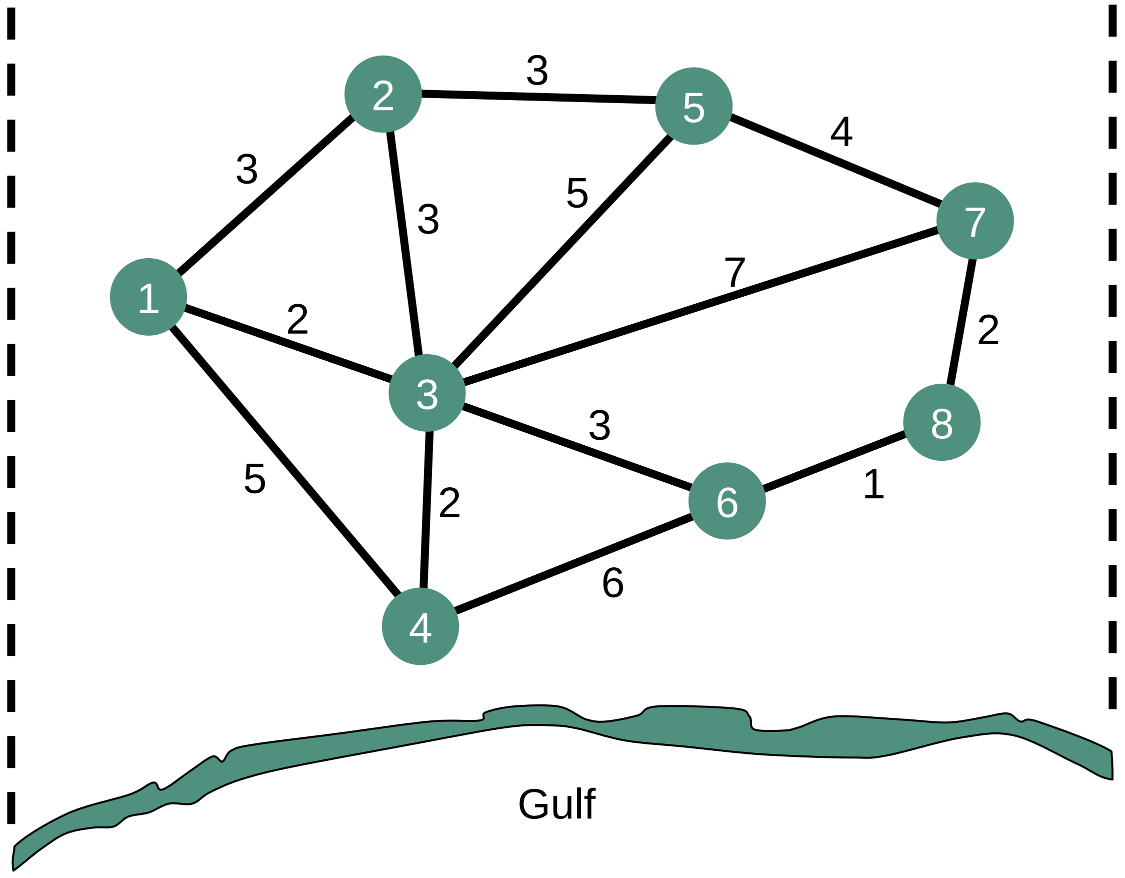
Minimal-Spanning Tree Technique

- The minimal-spanning tree technique involves connecting all the points of a network together minimizing the distance of the connecting lines
- Example:
  - A construction company is developing a housing project
  - They want to determine the least expensive way to provide utilities each house
  - There are 8 houses in the project and the distance between them is shown in the figure

DON'T WORRY, IT'S GONNA BE COOL. HOMER'S ON THE JOB, MAN!
THIS SITE UNDER CONSTRUCTION

Minimal-Spanning Tree Technique (con’t)

- Network for construction company



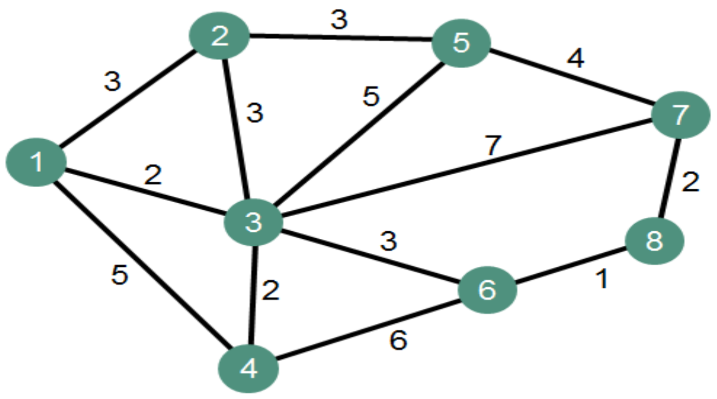
Minimal-Spanning Tree Technique (con’t)

- Steps for the minimal-spanning tree heuristic:
  1. Select any node in the network
  2. Connect this node to the nearest node that minimizes the total distance
  3. Considering all the nodes that are now connected, find and connect the nearest node that is not connected. If there is a tie, select one arbitrarily. A tie suggests there may be more than one optimal solution.
  4. Repeat the third step until all nodes are connected

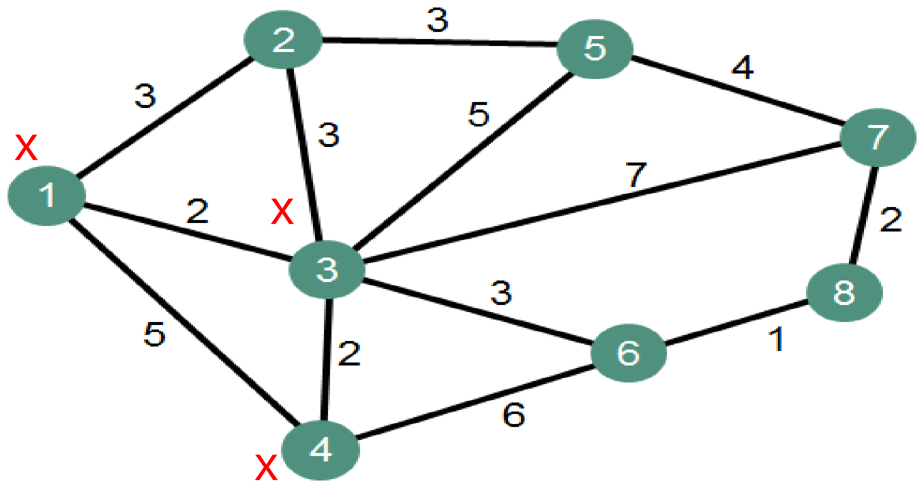
DON'T WORRY, IT'S GONNA BE COOL... HOMER'S ON THE JOB, MAN!
THIS SITE UNDER CONSTRUCTION

Minimal-Spanning Tree Technique (con't)

- Start by arbitrarily selecting node 1
- The nearest node is node 3 at a distance of 2 (200 feet) and we connect those nodes
- Considering nodes 1 and 3, we look for the next nearest node
- This is node 4, the closest to node 3
- We connect those nodes

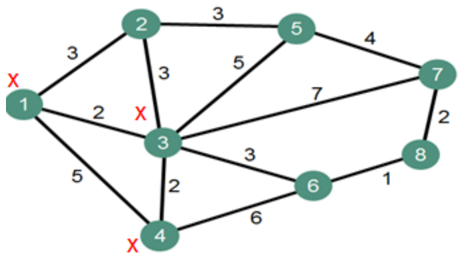


- We now look for the nearest unconnected node to nodes 1, 3, and 4
- This is either node 2 or node 6
- We pick node 2 and connect it to node 3



Minimal-Spanning Tree Technique (con't)

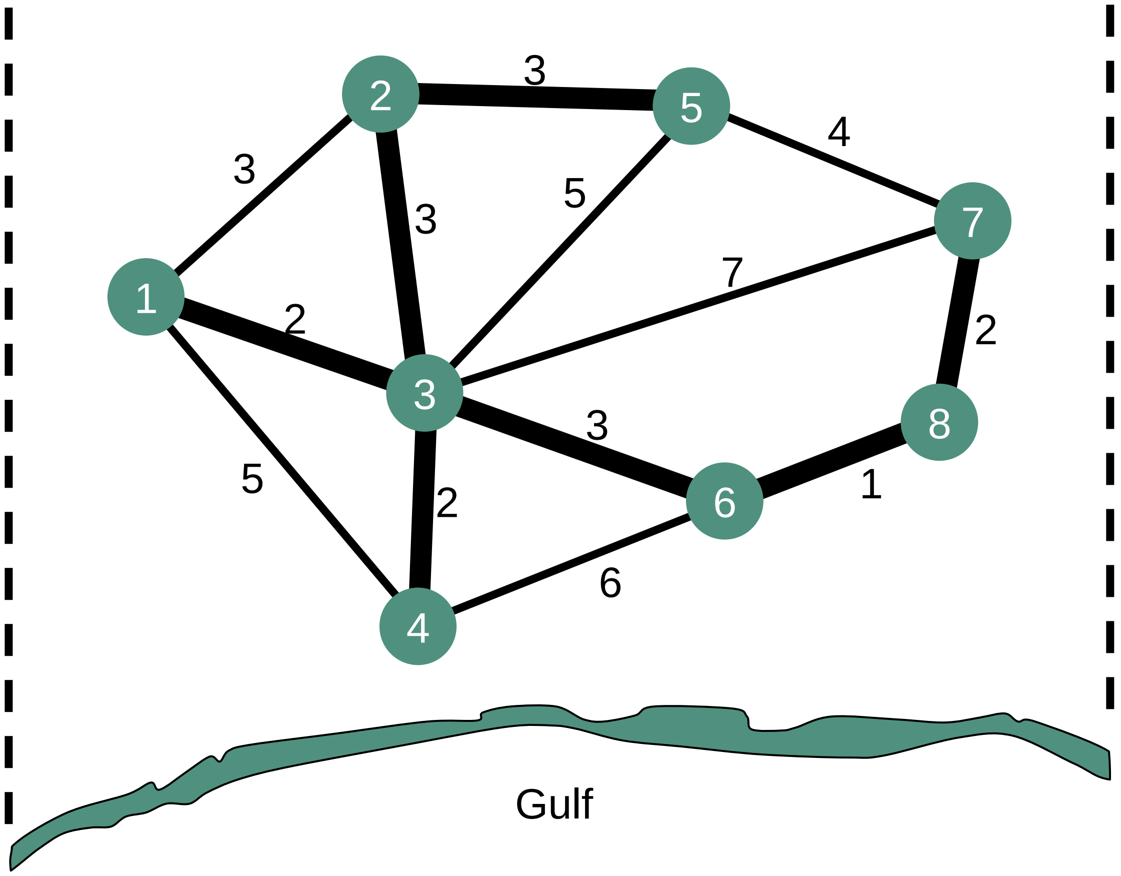
- Following this same process we connect from node 2 to node 5
- We then connect node 3 to node 6
- Node 6 will connect to node 8
- The last connection to be made is node 8 to node 7
- The total distance is found by adding up the distances in the arcs used in the spanning tree



2 + 2 + 3 + 3 + 3 + 1 + 2 = 16 (or 1,600 feet)

Minimal-Spanning Tree Technique (con't)

- After all iterations



Maximal-Flow Technique

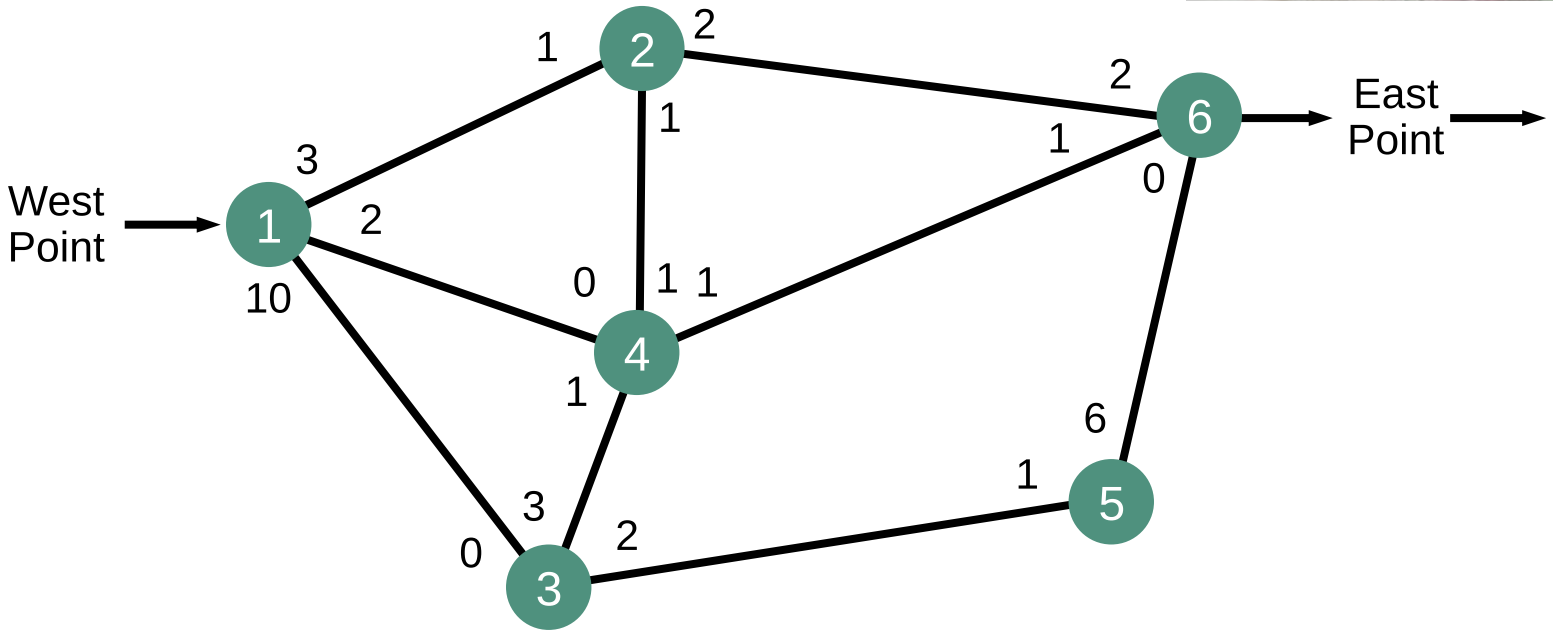
- The maximal-flow technique allows us to determine the maximum amount of a material that can flow through a network

- Example:
  - Waukesha Wisconsin is in the process of developing a road system for the downtown area
  - They want to determine the maximum number of cars that can flow through the town from west to east
  - The road network is shown in the following figure
  - The numbers by the nodes indicate the number of cars that can flow *from* the node

Maximal-Flow Technique (con't)

- Road network (not symmetric)

Capacity in Hundreds of Cars per Hour



Maximal-Flow Technique (con't)

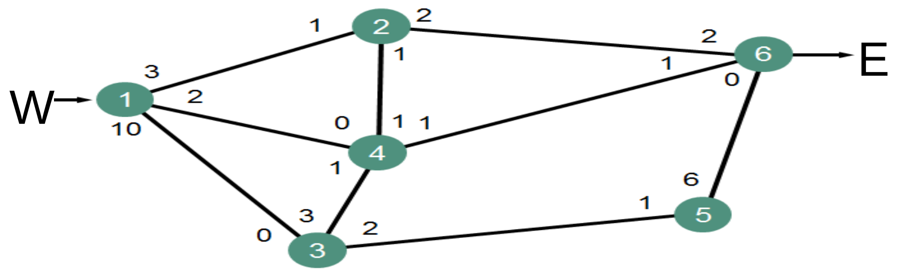
- Four steps of the Maximal-Flow heuristic:
  1. Pick any path from the start (*source*) to the finish (*sink*) with some flow. If no path with flow exists, then the optimal solution has been found.
  2. Find the arc on this path with the smallest flow capacity available. Call this capacity C. This represents the maximum additional capacity that can be allocated to this route.

Maximal-Flow Technique (con't)

- Four steps of the Maximal-Flow Technique
  3. For each node on this path, decrease the flow capacity in the direction of flow by the amount C. For each node on the path, increase the flow capacity in the reverse direction by the amount C.
  4. Repeat these steps until an increase in flow is no longer possible

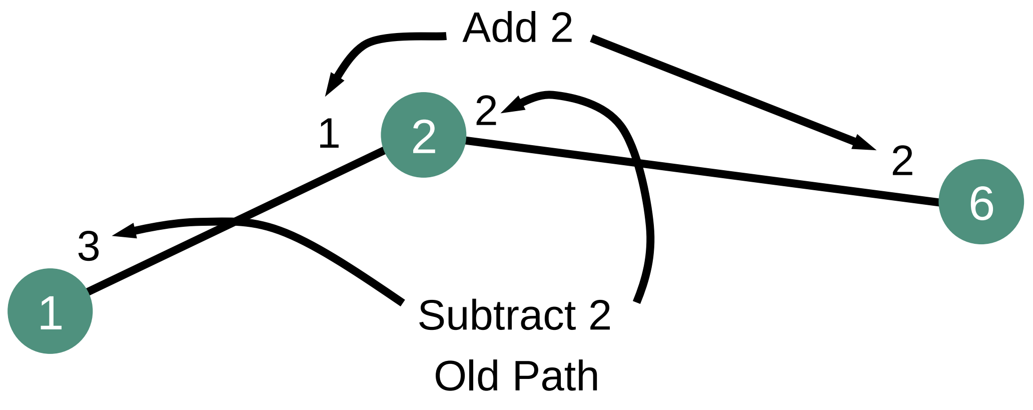
Maximal-Flow Technique (con’t)

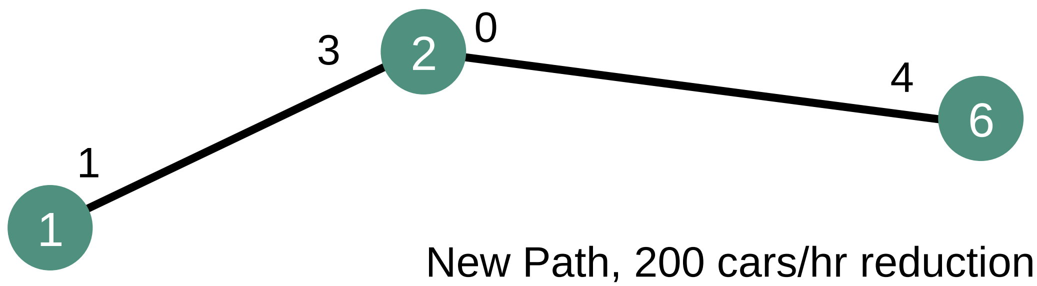
- We start by arbitrarily picking the path 1–2–6 which is at the top of the network
- The maximum path flow is 2 units from node 2 to node 6
- The path capacity is adjusted by adding 2 to the westbound flows and subtracting 2 from the eastbound flows (the units subtracted will accumulate to the maximum flow)
- The result is the new path in the following figure which shows the new relative capacity of the path at this stage



Maximal-Flow Technique (con't)

- Capacity adjustment for path 1–2–6 iteration 1

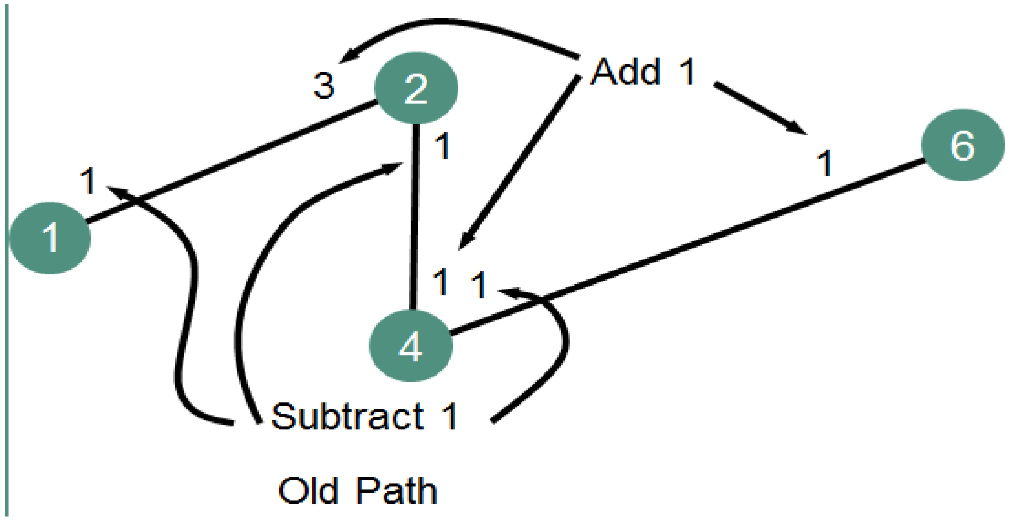




New Path, 200 cars/hr reduction West to East

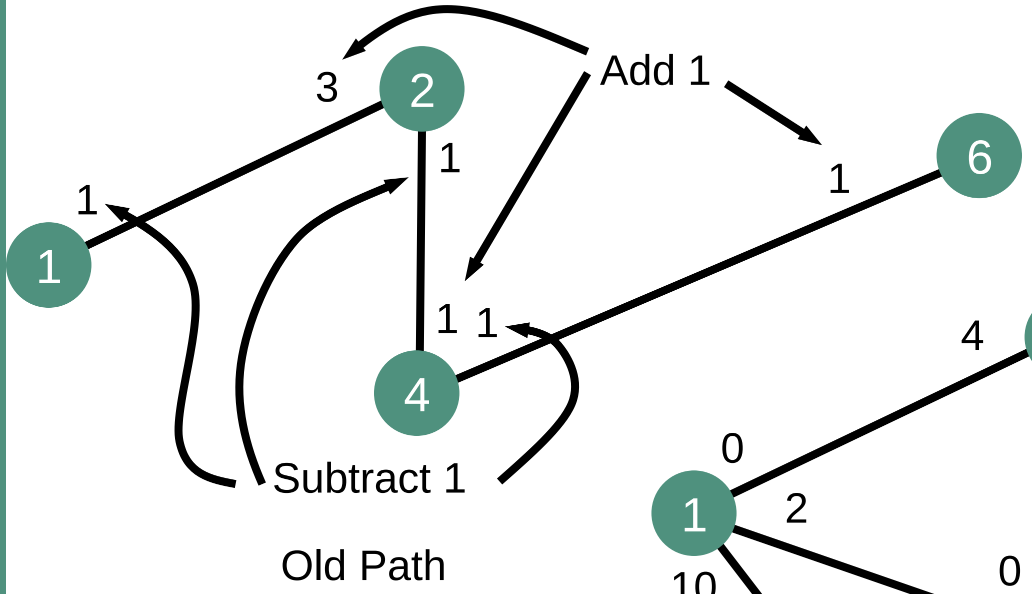
Maximal-Flow Technique (con't)

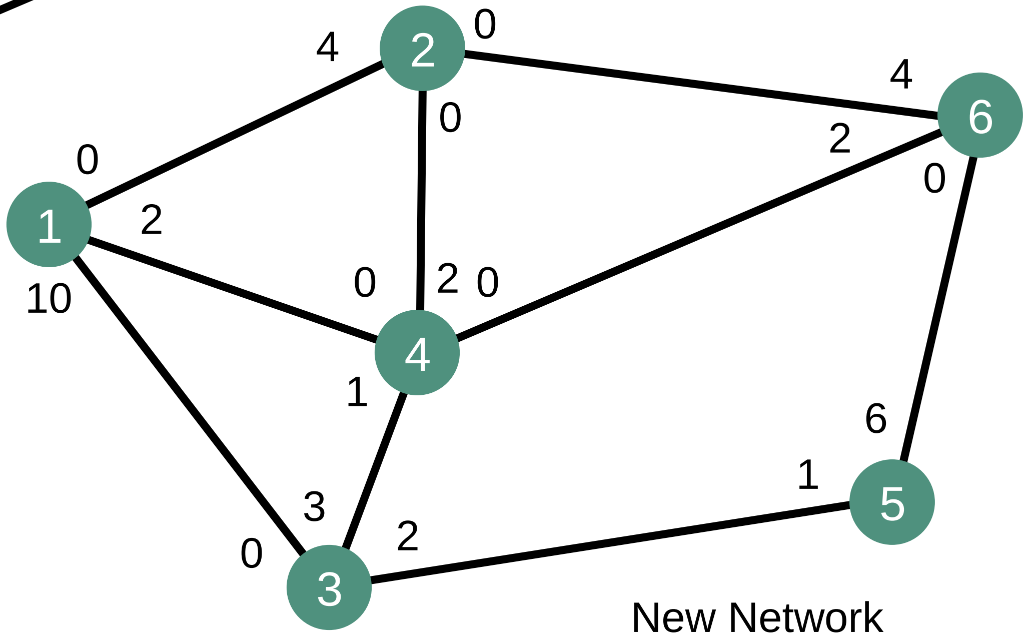
- We repeat this process by picking the path 1–2–4–6
- The maximum capacity along this path is 1
- The path capacity is adjusted by adding 1 to the westbound flows and subtracting 1 from the eastbound flows



Maximal-Flow Technique (con't)

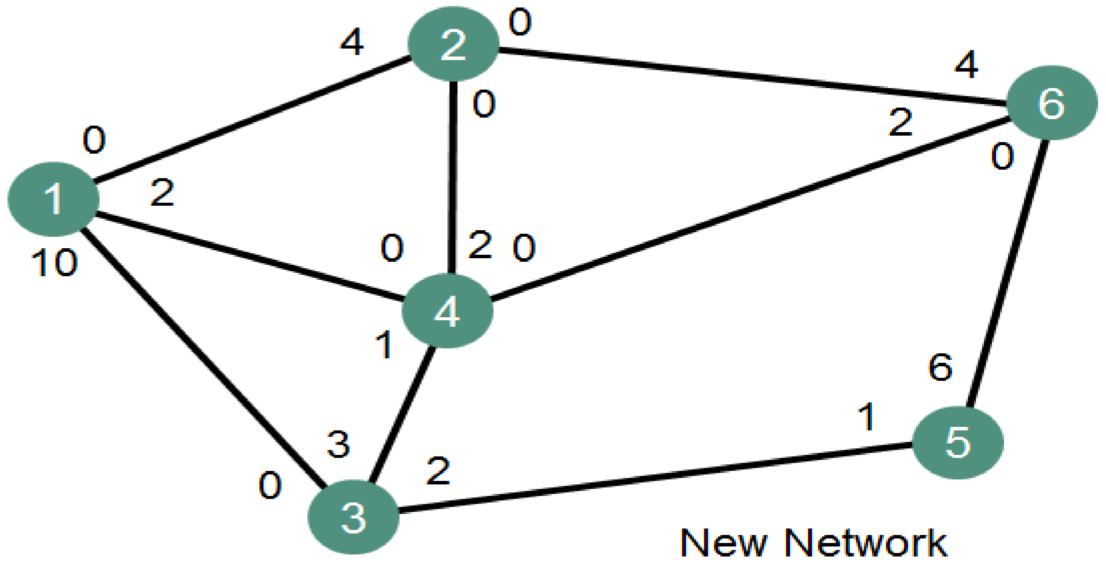
- Second iteration for Waukesha road system





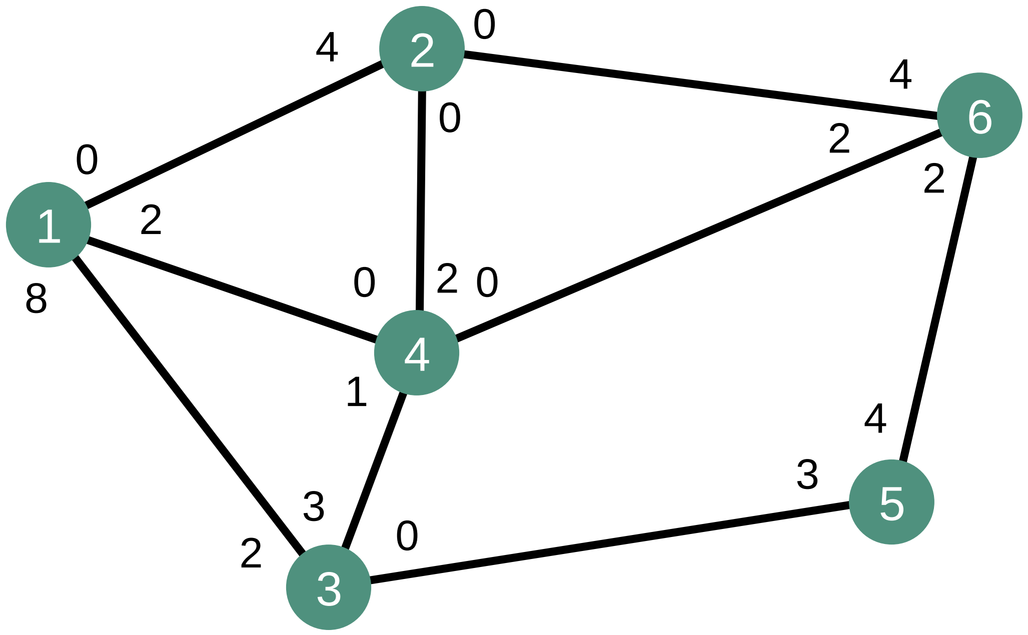
Maximal-Flow Technique (con't)

- We repeat this process by picking the path 1–3–5–6
- The maximum capacity along this path is 2



Maximal-Flow Technique (con't)

- Third and final iteration for Waukesha road system

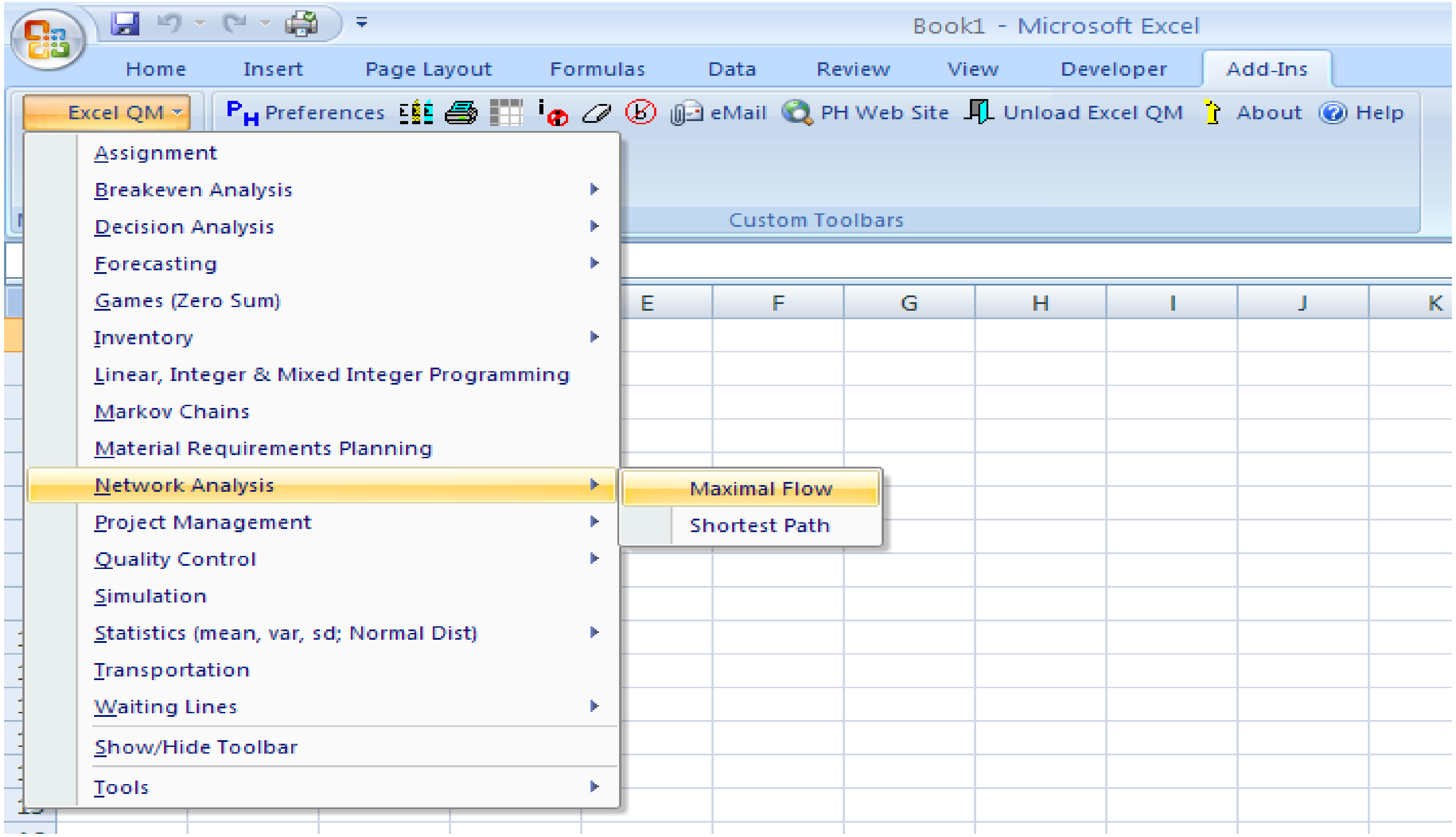


Maximal-Flow Technique (con’t)

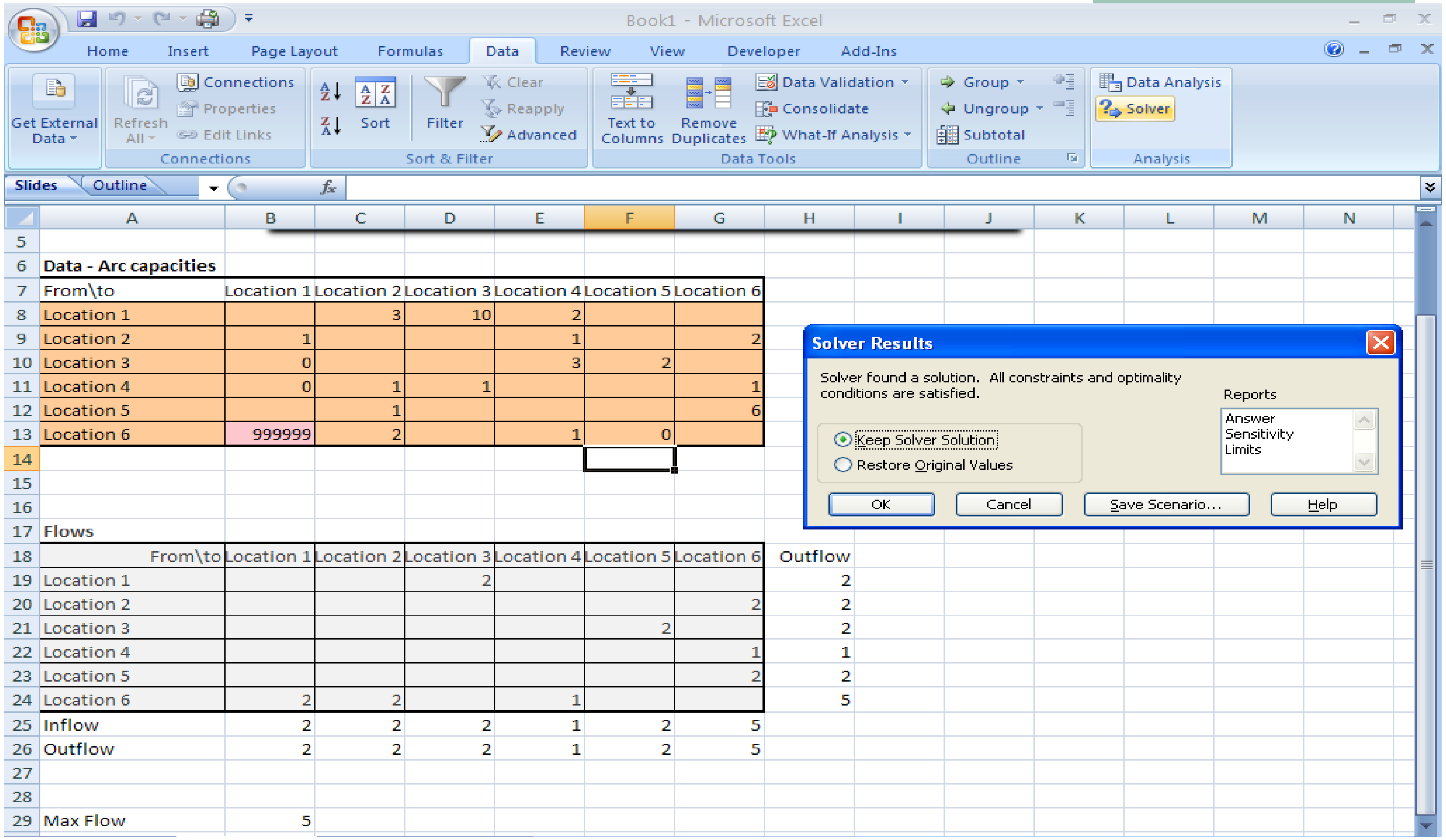
- There are no more paths from nodes 1 to 6 with unused capacity so this represents a final iteration
- The maximum flow through this network is 500 cars (the sum of the amounts subtracted out)

PATH		FLOW (CARS PER HOUR)
1–2–6		200
1–2–4–6		100
1–3–5–6		200
	Total	500

Maximal-Flow via LP & QM



Maximal-Flow Technique (con't)



Data - Arc capacities

From\to	Location 1	Location 2	Location 3	Location 4	Location 5	Location 6
Location 1		3	10	2		
Location 2	1			1		2
Location 3	0			3	2	
Location 4	0	1	1			1
Location 5		1				6
Location 6	999999	2		1	0	

Flows

From\to	Location 1	Location 2	Location 3	Location 4	Location 5	Location 6	Outflow
Location 1			2				2
Location 2						2	2
Location 3					2		2
Location 4						1	1
Location 5						2	2
Location 6	2	2		1			5
Inflow	2	2	2	1	2	5	
Outflow	2	2	2	1	2	5	

Max Flow 5

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

[✓ Keep Solver Solution] Restore Original Values

Reports: Answer, Sensitivity, Limits

OK | Cancel | Save Scenario... | Help

Sparse Matrices

- Most Matrices developed in “real world” algorithms and applications are characterized by the fact that most of the cells are empty or have zero values
- As the problem size increases, the cost of using traditional matrix algebra becomes very expensive
- For example, matrix multiplication operations are of order N cubed (3 nested loops)
- The field of “sparse matrix algebra” is concerned with the development of efficient data structure algorithms for matrix operations

Sparse Matrices (con't)

- In sparse matrix algebra, <u>only non-zero matrix elements are stored and operated upon</u>
- For example, matrix multiplication can be done in an order of N*NZ (instead of N*N*N) where NZ is the number of non-zero elements in the matrix
- The most efficient Quant systems use this type of matrix algebra ! (Microsoft Excel is not in this category)

Full Matrix Storage for Previous Example Graph

0	1	1	0	0
0	0	1	0	1
0	0	0	1	0
1	0	0	0	1
0	0	0	1	0

Sparse Representation of Data

[in cases where the costs are 1, A need not be stored at all !]

A	JA	CUMA
1	2	2
1	3	4
1	3	5
1	5	7
1	4	8
1	1	
1	5	
1	4	

JA and CUMA are integer vectors, and A is a real vector

Parallel and Vector Computers

- Some specialized scientific computers utilize either vector and/or parallel computation to minimize the time to do matrix type calculations
- Vector processors allow one computer hardware instruction to do vector/matrix computations by “massive pipelining”
- Parallel computation uses a large number of processors to do a single matrix operation
- Both of these special types of computers require special software algorithms designed to operate on full and/or sparse matrices

Cray Supercomputers
[http://www.cray.com/]

Cray Cluster Connect (C3)

Complete Lustre Solution for x86 Linux

Cray's trusted storage solutions, now available for commercial HPC clusters of all types

News

09.10.13 Cray Supercomputers Now Available With the New Intel (R) Xeon(R) Processor E5-2600 V2 Product Family **read more >>**

09.05.13 Japan's Railway Technical Research Institute Puts Multiple Cray Supercomputers Into Production **read more >>**

Events

09.09.13-09.11.13 HPC User Forum **read more >>**

09.17.13-09.19.13 Metacomp Technologies Symposium 2013 **read more >>**

CHAPEL

Blog

Chapel Parallel Iterators: Giving Programmers Productivity with Control

© 2013 Cray Inc. | Terms of Use | Trademarks | Privacy | Site Map

901 Fifth Avenue, Suite 1000, Seattle, WA 98164 | 206-701-2000 tel | 206-701-2500 fax

SHARE

HPE Cray EX “Frontier”

- The US is on top of the supercomputing world in the Top500 ranking of the most powerful systems (May 2022)
- The Frontier system from Oak Ridge National Laboratory (ORNL) running on AMD EPYC CPUs took first place from last year's champ, Japan's ARM A64X Fugak
- Frontier, powered by Hewlett Packard Enterprise's (HPE) Cray EX platform, was the top machine by a wide margin
- It's the first (known) true exascale system, hitting a peak 1.1 exaflops on the Linmark benchmark (1 quintillion calculations per second)
- Fugaku, meanwhile, managed less than half that at 442 petaflops, which was still enough to keep it in first place for the previous two years

HPE Cray EX “Frontier” (con’t)

CRAY
OAK RIDGE National Laboratory
U.S. DEPARTMENT OF ENERGY
Hewlett Packard Enterprise
AMD
FRONTIER

Dynamic Programming

- The problem of finding the least costly or shortest path can be extended where the path may be one in time and/or space (an N dimensional graph)
- Dynamic programming is a method for finding the best sequence of “activities” or “events”
- A common problem is finding the most EOQ (economical order quantity) when the future demand is known but is not constant (such as demands with trends and seasonal variations)

Dynamic Programming (con't)

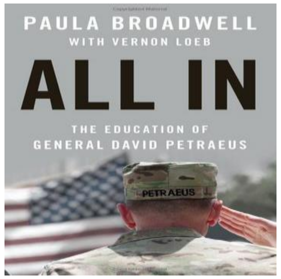
- Dynamic programming algorithms use similar matrix manipulations and are based upon:
  - Embedding – we will find the functions of objective points (i.e. EOQ's over time) that provides the lowest overall cost by solving the general problem for all EOQ's at all the points in time
  - For example, in the shortest-route problems, we found the value of the best route thru every node in the network as a means for finding the best route from node 1 to another specific node
- Textbook online module M2

References

- Statistical Analysis of Network Data: Methods and Models (Springer Series in Statistics) by Eric D. Kolaczyk
- Dynamic Programming (Princeton Landmarks in Mathematics and Physics) - (July 21, 2010) by Richard Bellman and Stuart Dreyfus
- A Management Guide to Pert/Cpm: With Gert/Pdm/Dcpm and Other Networks by Jerome D. Weist
- Network Optimization: Continuous and Discrete Models (Optimization, Computation, and Control) by Dimitri P. Bertsekas

Homework

- Textbook Sections 9.5 – 9.7, online module 8
- Quiz on network models
- Project Nine →



Project 9

- Paula Broadmind maintains a clandestine connection of contacts, some of whom can communicate directly with each other and some of whom she does not trust to directly communicate with each other due to likely wiretaps
- Her network matrix is shown on the following slide
- Build the communication matrix showing the <u>minimum number of steps</u> it takes each node to communicate with another

		1	1		1				
1									
			1				1	1	
							1		
	1	1	1				1	1	
		1					1	1	
		1	1		1			1	
				1				1	
							1		1
		1				1			