# Graphs and network matrices

Olli-Pekka Hämäläinen

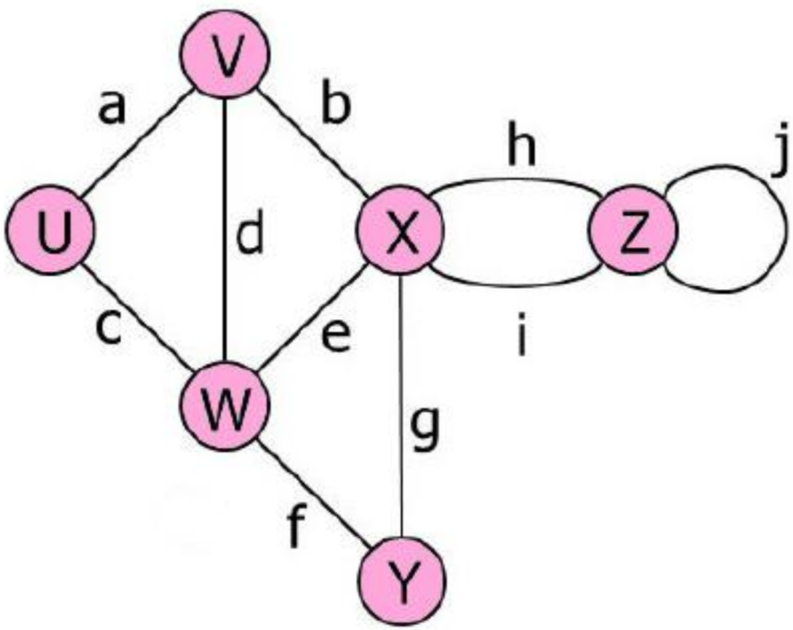
# Graphs

- We already used graphs in order to illustrate relations:
  - Domain-range-graph
  - Digraph (directed graph)
- Let's now get ourselves acquainted with graphs on general level in order to familiarize ourselves with terminology and gain more analysis tools
- With relations, the nodes of the graph were domain/range elements
  - Generally, in graphs these nodes are called *vertices**
- Arrows represented the connections between domain/range elements
  - Generally, these are called *edges*
  - Can be either directed (as in arrows) or undirected
- This branch of mathematics that focuses on analysis of graphs is graph theory or network theory

*Personally I prefer the term "node" and will use that in these lectures.

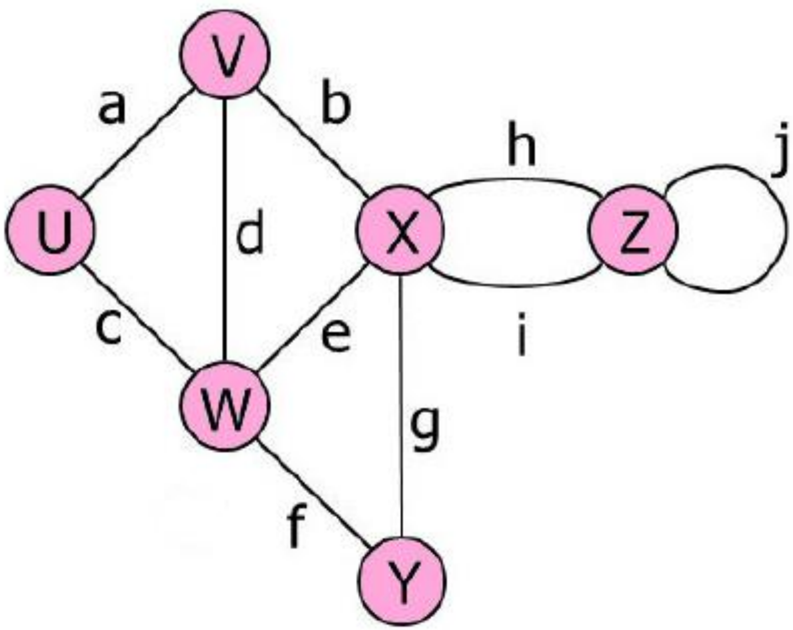
# Graph terminology

- *Endpoints* of an edge = nodes where the edge starts and ends (logically); for example, edge h has endpoints X and Z
- Edges a, b and d are *connected* to node V
- The *degree* of a node is the number of edges that are connected to it (example: degree of V = 3 & degree of W = 4



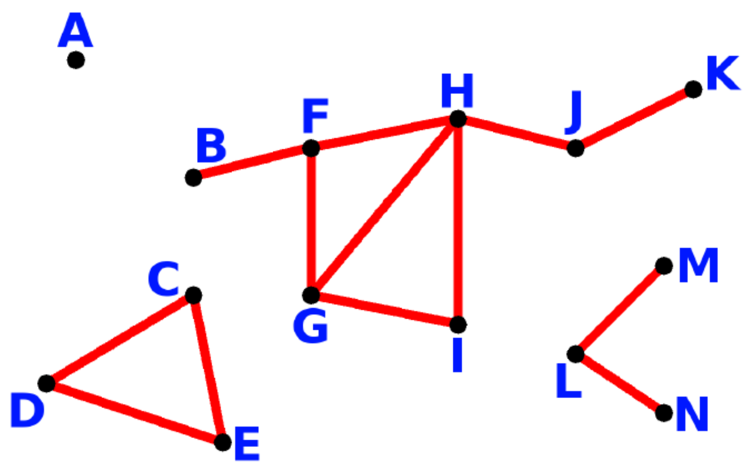
# Graph terminology

- Edges which have same endpoints are called *parallel* (for example, h and i here)
- If there is an edge between two nodes, these nodes are said to be *adjacent* (example: W and Y adjacent, Y and V are not)
- If the edge has only one endpoint, it's called a *loop* (example: j)



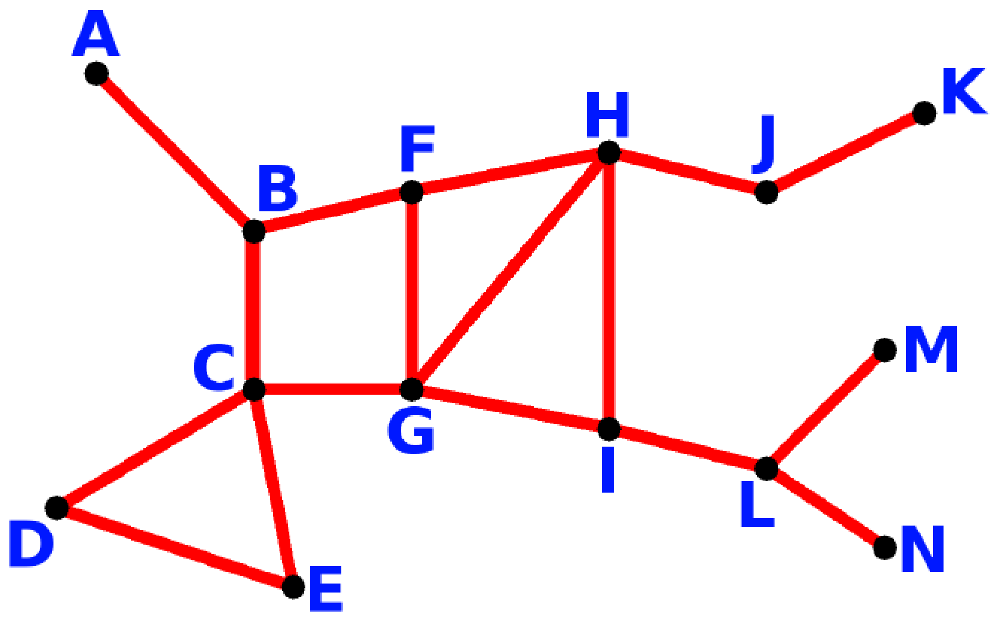
# Graph terminology

- If we can get from node X to node Y via one or several edges, there is a connection between X and Y – a *path*
  - There can be several possible paths - for example, here from B to I there are four: BFGI, BFHI, BFHGI, BFGHI
- If there exists a path between any two nodes, the graph is said to be *connected*
  - The graph below is not
- Nodes of degree 0 are called *isolated nodes*
  - In the graph below, A is an isolated node



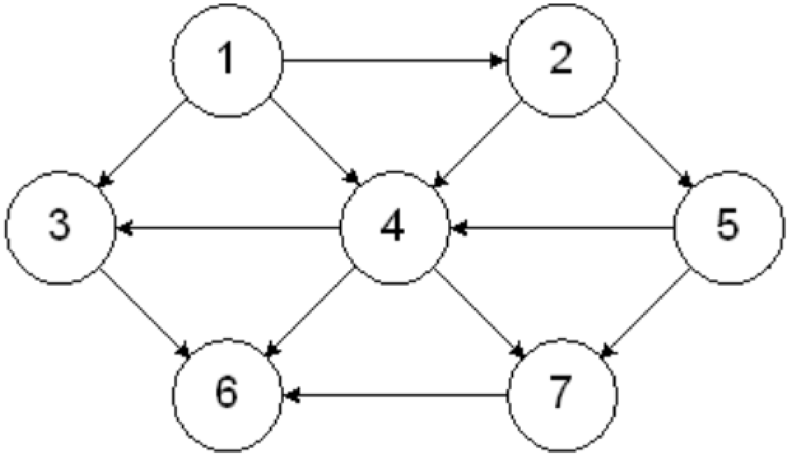
# Graph terminology

- If there is even one path that starts from arbitrary node X and comes back to it without travelling via the same edge twice, the graph is called *cyclic*
  - In the graph below, for example path BCGFB



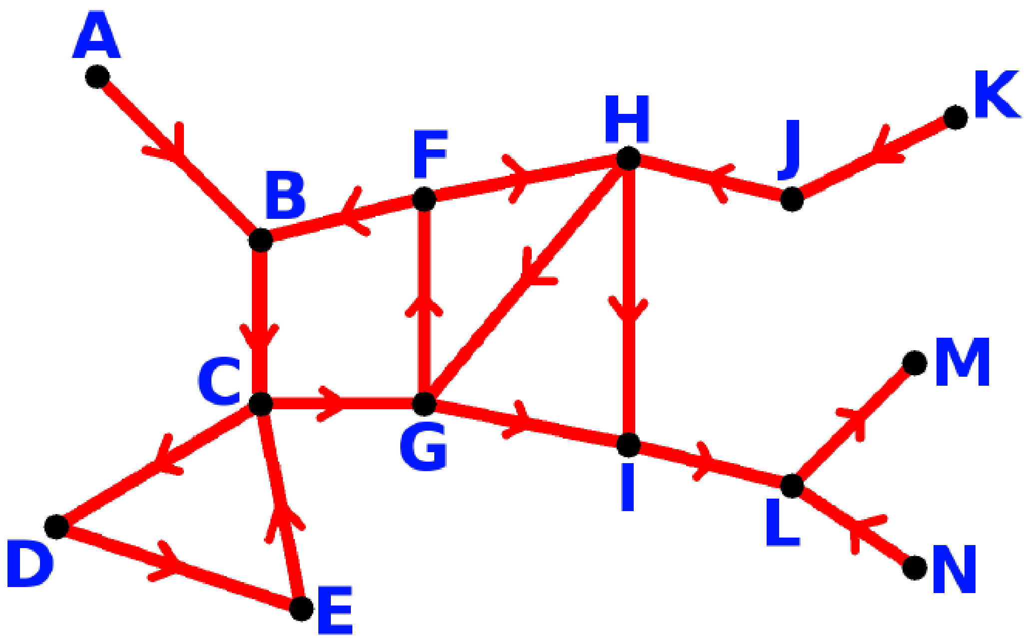
# Directed graph

- In previous examples, the graphs were undirected – so, the connections between nodes were automatically 2-way
- We can declare our edges to have a direction by marking the edge as an arrow
- This kind of graph is called a *directed graph* or *digraph*
  - We already used these to illustrate relations!



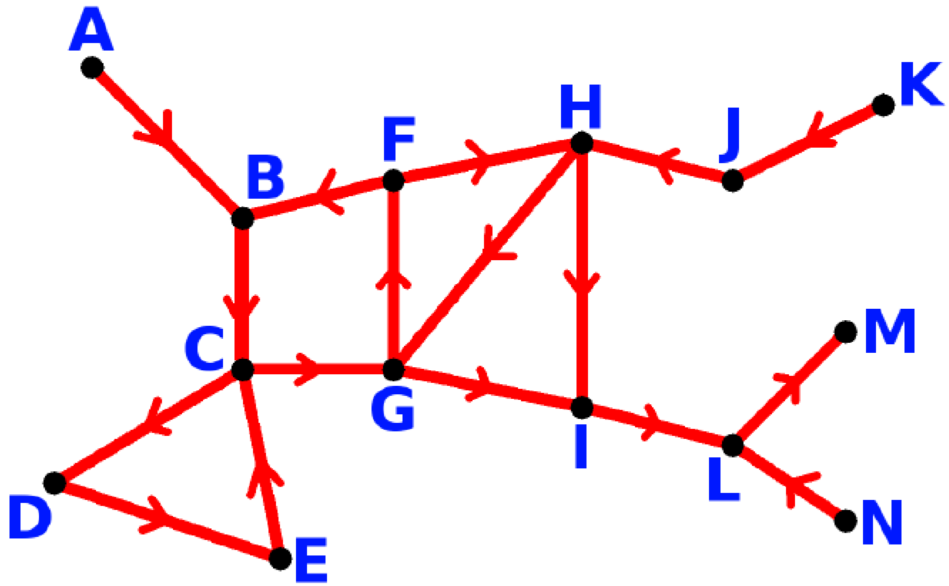
# Directed graph

- The definition of a path is not changed in the case of digraph, but the number of possible paths is decreased
  - Example: now we can only get from B to I via path BCGI or BCGFHI
  - Going from I to B is impossible



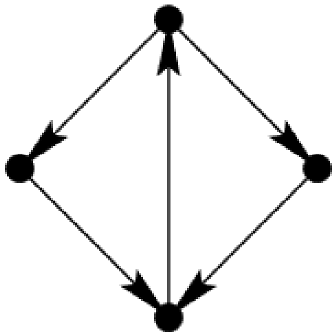
# Directed graph

- The concept of degree of a node is split twice:
  - *Indegree* of a node = edges that arrive the node
  - *Outdegree* of a node = edges that leave the node
- If the indegree is zero, but the outdegree is not, the node is called a *source* (example below: A, K, N)
- If the outdegree is zero, but the indegree is not, the node is called a *sink* (example below: M)

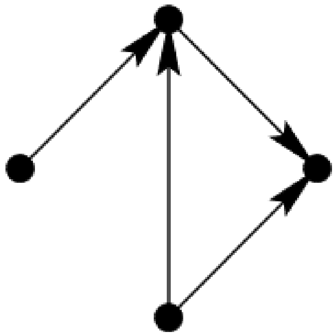


# Directed graph

- If the undirected graph that corresponds to the directed graph is connected, also the directed graph is connected
- If we can get from any node to any other node, the directed graph can be called *strongly connected* (graph $G_1$)
  - If the undirected graph is connected, but we can't get from any node to any other in the directed graph, the graph is called *weakly connected* (graph $G_2$)
- A strongly connected graph has no sources nor sinks



$G_1$



$G_2$

# Choice of graph type

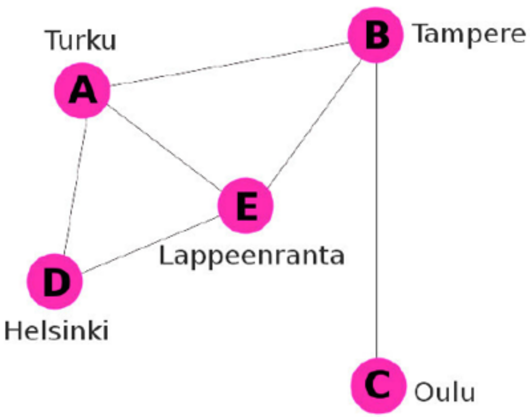
- Whether the graph should be directed or not, and what kind of structures the graph is allowed to have, depends naturally on the process or situation to be modeled
  - For example, road networks are usually undirected – unless there are one-way streets in the network
- Undirected graphs are often “denied” the option to possess parallel edges or loops
  - If such structures exist, the graph is actually a *multigraph*
  - Most source books use the word “graph” quite vaguely as a general term – even though a graph is a special case of a multigraph
  - In order to make distinctions, graphs that have no parallel edges or loops, are often referred to as *simple graphs*

# Network matrices

- Small graphs are easy to analyze graphically, but as the size of the graph increases, analysis gets tougher
  - Best option is to take advantage of computers
- The easiest way to define a graph for a computer is to describe it via *network matrices*
  - Notions "graph" and "network" have no fundamental difference
  - Some sources do link the notion "network" only to directed graphs, though
- Let's familiarize ourselves with some network matrices
  - Applications are mostly left for follow-up courses

# Adjacency matrix A

- *Adjacency matrix* describes which nodes are adjacent to each other (as the name suggests)
  - Nodes as rows and columns → always a square matrix
  - If there is an edge* from node i to node j, the value of element $a_{ij}$ of the adjacency matrix is 1 (otherwise 0)
  - Diagonal elements either 0 or 2 (if the node has a loop)
- The adjacency matrix of an undirected graph is always symmetric (example: train network between cities)



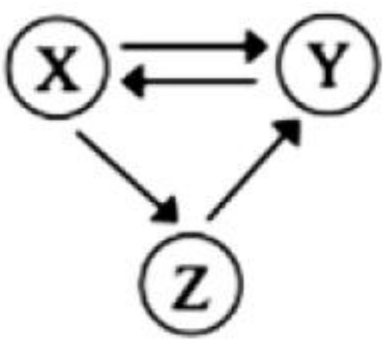

$A =$

| | A | B | C | D | E |
|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 1 | 1 |
| B | 1 | 0 | 1 | 0 | 1 |
| C | 0 | 1 | 0 | 0 | 0 |
| D | 1 | 0 | 0 | 0 | 1 |
| E | 1 | 1 | 0 | 1 | 0 |

*Or, in case of multiple parallel edges, $a_{ij}$ = number of edges

# Adjacency matrix A

- In a directed graph, directions are taken into account:
  - If there is an arrow from node i to node j, then $a_{ij} = 1$
  - This arrow doesn’t make element $a_{ji}$ to one
- Therefore, the adjacency matrix of a directed matrix is not (usually) symmetric
- The relation matrices we used during the previous week are actually adjacency matrices of relation digraphs!



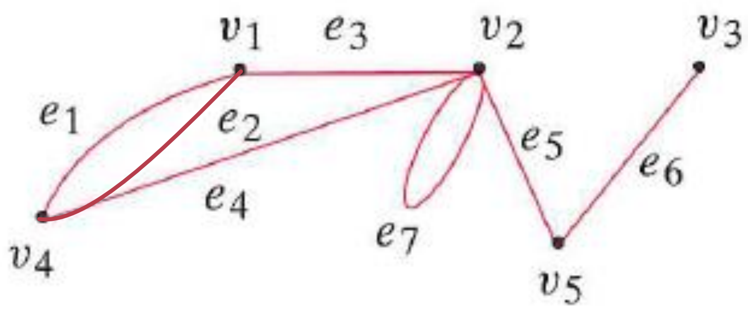

| | X | Y | Z |
|---|---|---|---|
| X | 0 | 1 | 1 |
| Y | 1 | 0 | 0 |
| Z | 0 | 1 | 0 |

# Incidence matrix B

- Another possible way to depict a graph in matrix form is to define its *incidence matrix*
  - Rows are nodes ($x_i$ or $v_i$), columns are edges ($e_j$)

$$b_{ij} = \begin{cases} 1, if\ e_j\ is\ connected\ to\ node\ x_i \\ 0, otherwise \end{cases}$$

  - NOTE! This is not a square matrix (unless in special cases)

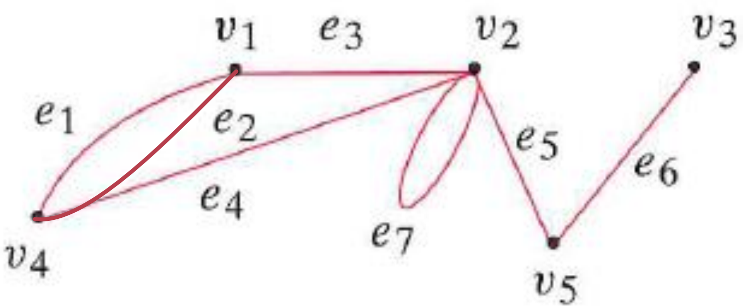


# Incidence matrix B

- Another possible way to depict a graph in matrix form is to define its incidence matrix
  - Rows are nodes ($x_i$ or $v_i$), columns are edges ($e_j$)

$$b_{ij} = \begin{cases} 1, if\ e_j\ is\ connected\ to\ node\ x_i \\ 0, otherwise \end{cases}$$

  - NOTE! This is not a square matrix (unless in special cases)



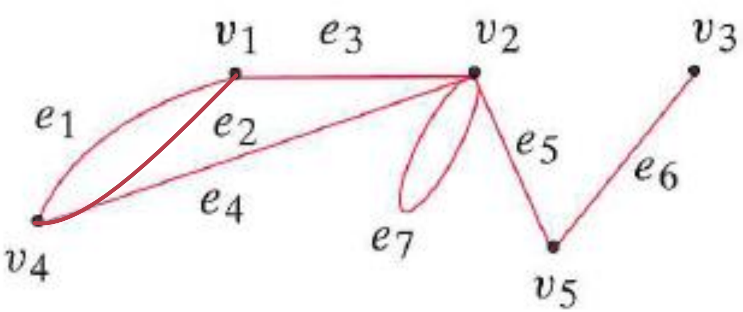

$$B = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

# Incidence matrix B

- Another possible way to depict a graph in matrix form is to define its incidence matrix
  - Rows are nodes ($x_i$ or $v_i$), columns are edges ($e_j$)

$$b_{ij} = \begin{cases} 1, if\ e_j\ is\ connected\ to\ node\ x_i \\ 0, otherwise \end{cases}$$

  - NOTE! This is not a square matrix (unless in special cases)




$B =$

| | $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ |
|---|---|---|---|---|---|---|---|
| $v_1$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $v_2$ | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| $v_3$ | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| $v_4$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| $v_5$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Identical columns = edges 1 and 2 are parallel!

Columns that have just one 1 = loops

# Incidence matrix B

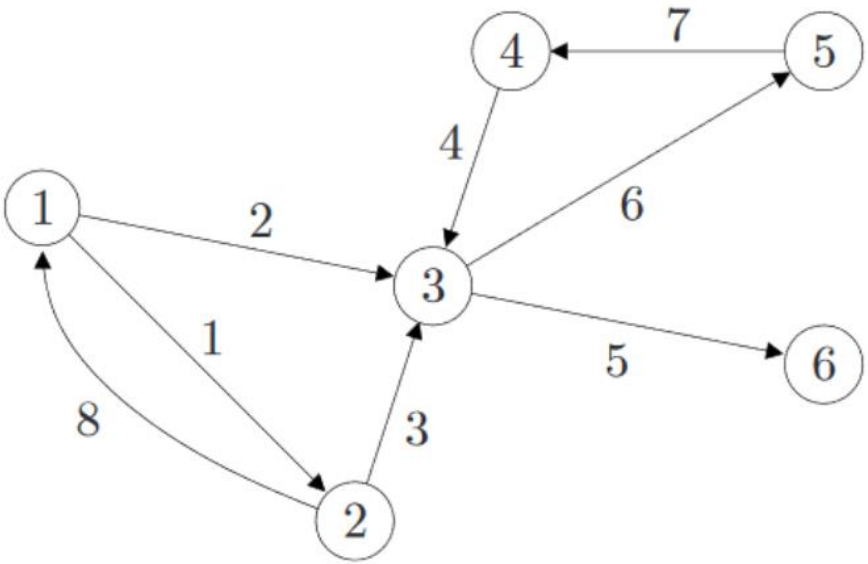
- In case of a directed graph, the edges (= arrows) can get either a positive or negative value – depending on the direction of the arrow:

$$b_{ij} = \begin{cases} 1, & if\ arrow\ j\ starts\ from\ node\ i \\ -1, & if\ arrow\ j\ ends\ in\ node\ i \\ 0, & otherwise \end{cases}$$

- Incidence matrix is usually greater in size than adjacency matrix (depends on the number of edges, though)
- Removal of careless errors: because each column corresponds to one edge, each column must have at most two 1s (undirected graph) or 1 and -1 (directed graph)

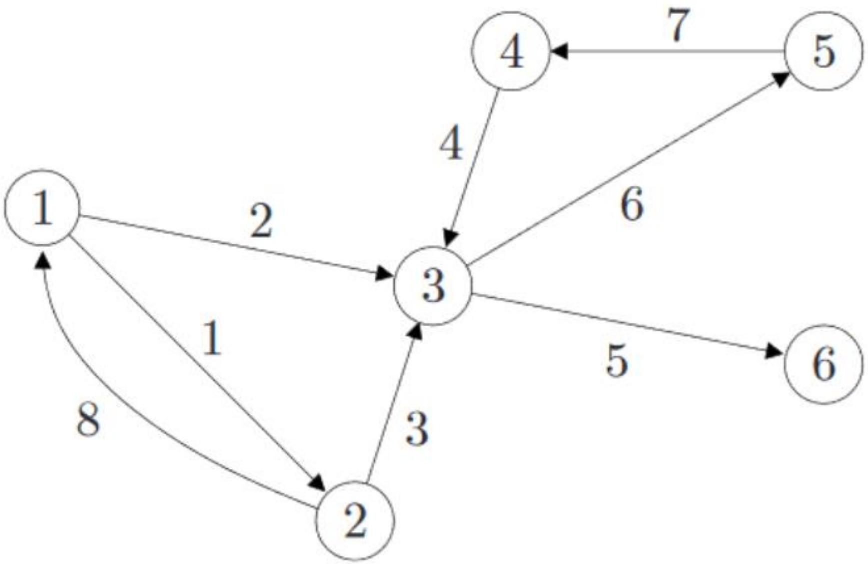
# Incidence matrix B

- Example:



# Incidence matrix B

- Example:



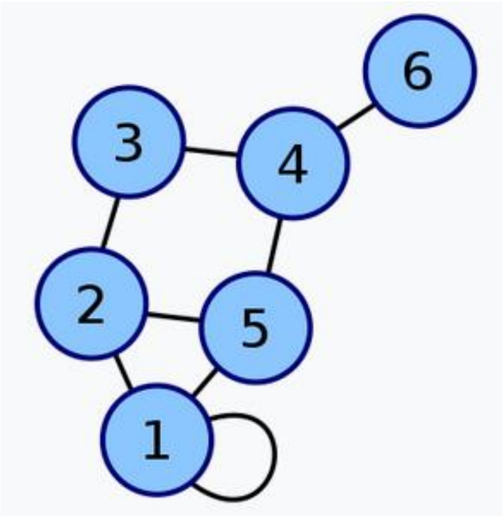

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

Nodes as rows (6 pcs)

Arrows as columns (8 pcs)

# Degree matrix D

- *Degree matrix* specifies, how many edges are connected to each node – so, degrees of nodes
  - Nodes as rows and columns → always a square matrix
  - Pure diagonal matrix
  - Loops are calculated twice (because a loop both starts and ends in same node); this is why here degree of node 1 = 4




$$D = \begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

# Degree matrix D

- In case of a directed graph, the degree matrix has three alternative interpretations:
  - Degree matrix of the corresponding undirected graph
  - Indegree matrix of the directed graph
  - Outdegree matrix of the directed graph

- Pay attention to definitions when reading sources!
- Usually, if text just says “degree matrix” with a directed graph, this means the first alternative
  - If the degree matrix is linked to indegrees or outdegrees, this is specifically mentioned

# Laplace matrix L

- In deeper graph analysis, one commonly needed tool is *Laplace matrix*
- This can be defined via previously introduced matrices
  - 2 possible methods
- Directed graph: product of incidence matrix and its transpose

$$L = BB^T$$

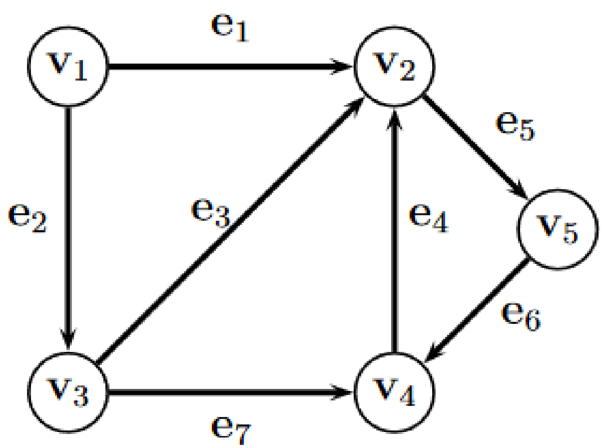
  - NOTE: for undirected graphs, $L = BB^T - 2A$
- Undirected graph: Difference of degree matrix and adjacency matrix

$$L = D - A$$

  - NOTE: This can be used for directed graphs, too – in this case, D and A must be the degree and adjacency matrices of the corresponding undirected graph

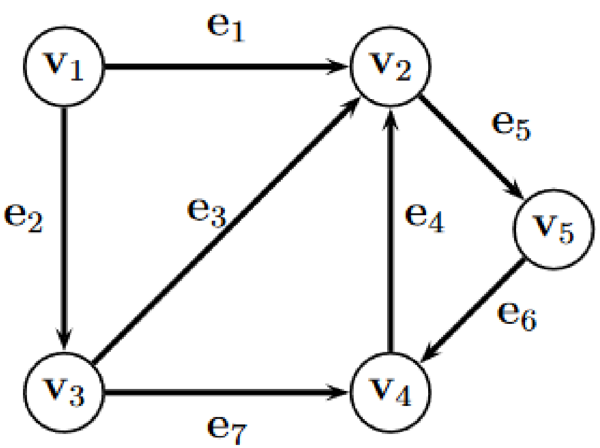
# Network matrices: example

- Directed graph $G_1$ is as follows:



# Network matrices: example

- Directed graph $G_1$ is as follows:

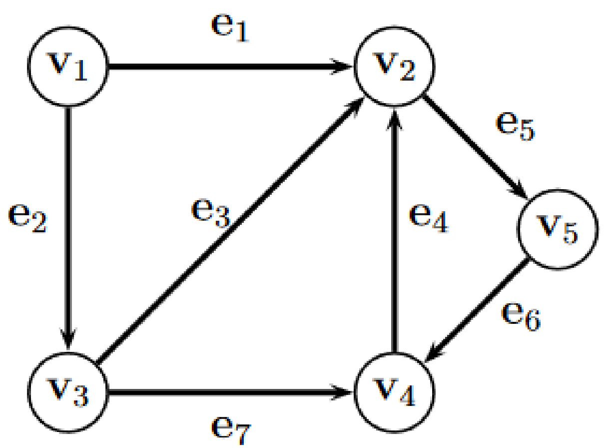



- Incidence matrix

$$B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{pmatrix}$$

# Network matrices: example

- Directed graph $G_1$ is as follows:



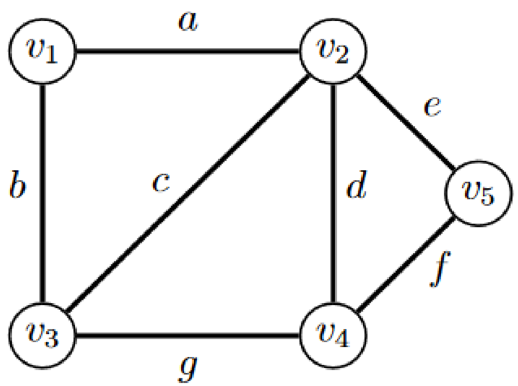

- Incidence matrix and adjacency matrix

$$B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{pmatrix}$$

$$A(G_1) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

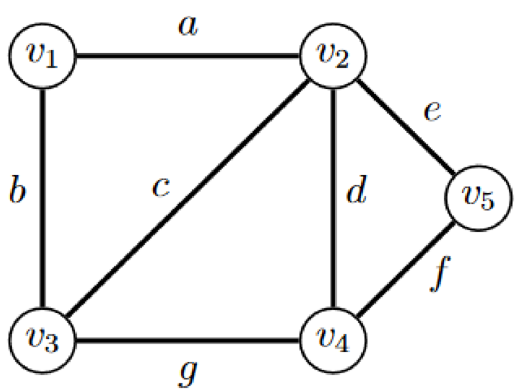
# Network matrices: example

- Convert the graph to an undirected graph $G_2$:



# Network matrices: example

- Convert the graph to an undirected graph $G_2$:

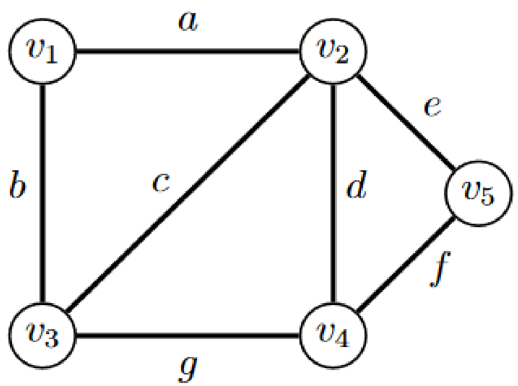



- Adjacency matrix of $G_2$

$$A(G_2) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

# Network matrices: example

- Convert the graph to an undirected graph $G_2$:




- Adjacency matrix of $G_2$ and its degree matrix

$$A(G_2) = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

# Network matrices: example

- Laplace matrix using incidence matrix: directed graph, so

$$L = BB^T = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

# Network matrices: example

- Laplace matrix using incidence matrix: directed graph, so

$$L = BB^T = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

- ...or, by using degree and adjacency matrices of $G_2$:

$$L = D - A(G_2) = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

# Network matrices: example

- Laplace matrix using incidence matrix: directed graph, so

$$L = BB^T = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

- ...or, by using degree and adjacency matrices of $G_2$:

$$L = D - A(G_2) = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

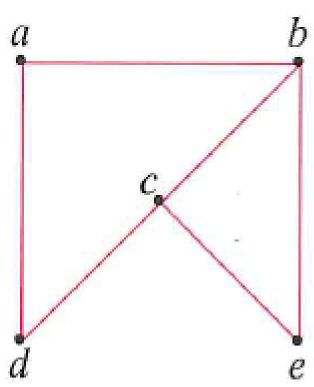
$$= \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

# Properties of matrices A and B

- Which matrix (A or B) should we use, then?
  - If there are more edges than nodes in the graph (most common situation), adjacency matrix is smaller and requires less space & calculation time; therefore, it is also more commonly used
  - If there are less edges than nodes, incidence matrix is smaller
  - One bonus of B: enables separation of parallel edges!
- Degree numbers of nodes can be seen directly from the adjacency matrix by calculating row and column sums
- In an incidence matrix, degrees of nodes can be calculated from row sums
  - Column sums are always either 0 (directed graph), 1 (loop) or 2 (undirected graph)

# Properties of matrices A and B

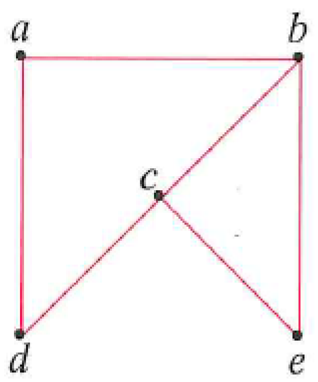
- For a simple undirected graph, the powers of adjacency matrix have a special meaning:
  - Element $a_{ij}$ of matrix $A^n$ tells the number of different possible n-step paths from node i to node j
  - In other words: in how many ways can we travel from node i to node j via n pcs of edges




$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

# Properties of matrices A and B

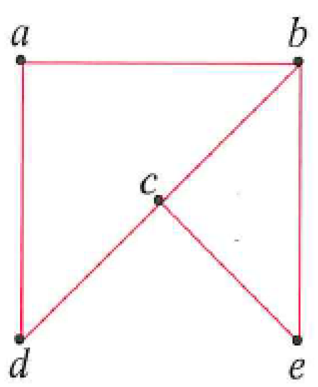
- For a simple undirected graph, the powers of adjacency matrix have a special meaning:
  - Element $a_{ij}$ of matrix $A^n$ tells the number of different possible n-step paths from node i to node j
  - In other words: in how many ways can we travel from node i to node j via n pcs of edges




$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array} \qquad A^2 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{pmatrix} 2 & 0 & 2 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 1 \\ 0 & 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{pmatrix} \end{array}$$

# Properties of matrices A and B

- For a simple undirected graph, the powers of adjacency matrix have a special meaning:
  - Element $a_{ij}$ of matrix $A^n$ tells the number of different possible n-step paths from node i to node j
  - In other words: in how many ways can we travel from node i to node j via n pcs of edges




$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array} \quad A^2 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 2 & 0 & 2 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 1 \\ 0 & 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{pmatrix} \end{array}$$

Example: from node a to node c using 2 steps, there are 2 path options (abc, adc). From node c to node c using 2 steps, there are 3 path options (cbc, cdc, cec).

# Properties of Laplace matrix

- Laplace matrix is always symmetric
- All row and column sums are zero
- All eigenvalues are positive
- Laplace matrix and its eigenvalues can be used for many purposes – for example:
  - Minimal spanning tree search
  - Other network optimization problems
  - Computer vision and machine learning solutions
- The actual applications are left for follow-up courses
  - Ask Jouni Sampo during matrix calculation course ☺

# Thank you!