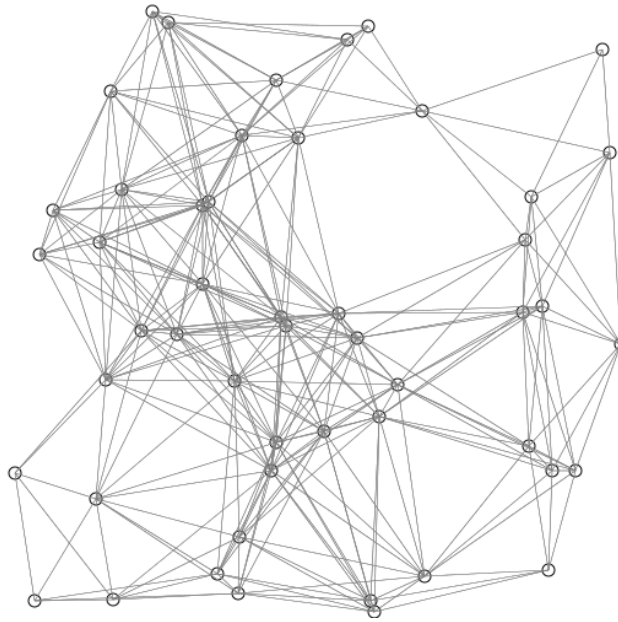


Matrix representations of graphs

Why?

Why would we want to replace an attractive vertex and edge diagram with a tabular matrix? Purely for practical reasons. A small graph is easily grasped and processed in its visual representation, but as graphs grow larger and their interconnections more dense it becomes more difficult to interpret the connections in them, and impossible to process them.



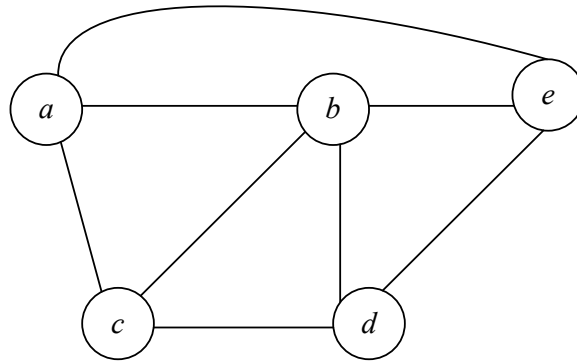
The matrix representation may become large, but it at least remains legible. And of course, a matrix representation can serve as the starting point for automated processing of a graph by a computer program.

How?

There is a straightforward way to represent a graph using a matrix: Have a row and a column for each vertex, and enter the edge information in the cells within the matrix.

Undirected Graph

For example given the graph:



We build a matrix with a row and a column for each vertex:

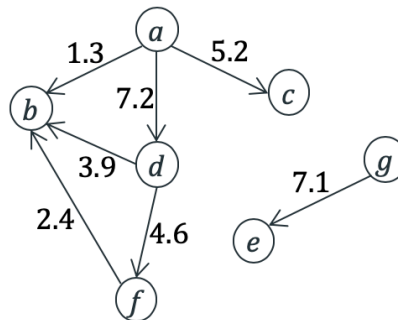
$$\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{array}{ccccc} a & b & c & d & e \\ \left[\begin{array}{ccccc} & & & & \end{array} \right] \end{array}$$

Then we enter the edge information in the matrix cells. That would typically be the edge weights, but since this graph is unweighted we enter 1s when there is an edge connecting two vertices, and a 0 when there isn't:

$$\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{array}{ccccc} a & b & c & d & e \\ \left[\begin{array}{ccccc} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{array} \right] \end{array}$$

Directed Graph

We follow the same process for any graph, for example this disconnected, weighted, directed graph:



We build a matrix with a row and a column for each vertex:

$$\begin{array}{c}
 a \\
 b \\
 c \\
 d \\
 e \\
 f \\
 g
 \end{array}
 \begin{bmatrix}
 & a & b & c & d & e & f & g \\
 a & & 1.3? & & & & & \\
 b & 1.3? & & & & & & \\
 c & & & & & & & \\
 d & & & & & & & \\
 e & & & & & & & \\
 f & & & & & & & \\
 g & & & & & & &
 \end{bmatrix}$$

The next step is to enter the edge weights in to the matrix, but that raises a question. Since the edges are directed should the weight of 1.3 on the edge from *a* to *b* go in the cell at row *a* and column *b*, or the cell at column *a* and row *b*? There is no right answer, but there is a convention, and that is to treat rows as From or origins, and columns as To or destinations. So the weight of 1.3 on the edge **from** *a* **to** *b* will go in the cell at **row** *a* and **column** *b*. That leads to this matrix representation of the graph:

$$\begin{array}{c}
 \text{From} \\
 a \\
 b \\
 c \\
 d \\
 e \\
 f \\
 g
 \end{array}
 \begin{array}{c}
 \text{To} \\
 a \quad b \quad c \quad d \quad e \quad f \quad g
 \end{array}
 \begin{bmatrix}
 - & 1.3 & 5.2 & 7.2 & - & - & - \\
 - & - & - & - & - & - & - \\
 - & - & - & - & - & - & - \\
 - & 3.9 & - & - & - & 4.6 & - \\
 - & - & - & - & - & - & - \\
 - & 2.4 & - & - & - & - & - \\
 - & - & - & - & 7.1 & - & -
 \end{bmatrix}$$

Notice that there are relatively few entries in the matrix; we call matrices like this *sparse*.

Notice too the rows that contain no numbers only dashes. These are vertices you cannot leave, and are called *sinks*.