Discrete Structures: CMPSC 102

Oliver BONHAM-CARTER

Fall 2019
Week 9

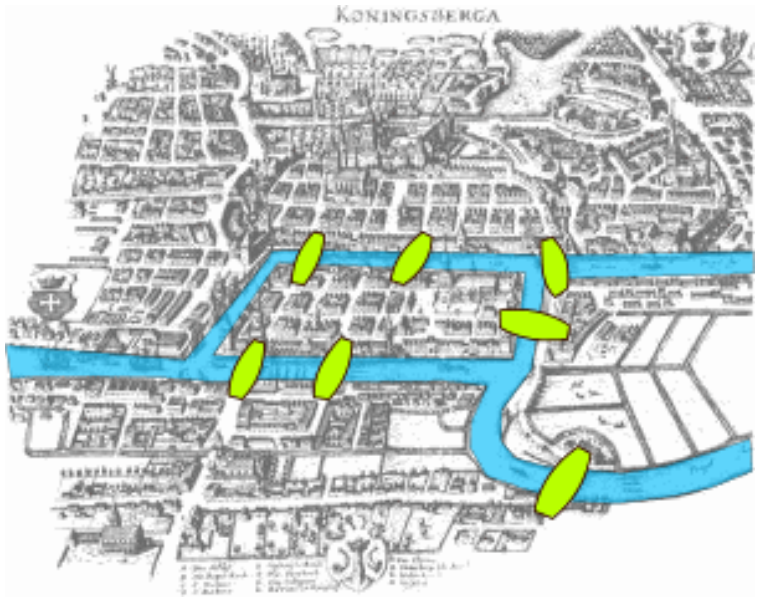
Leonhard Euler

Creator of Graph theory

- Swiss mathematician, physicist, astronomer, logician and engineer:
- 5 April 1707 - 18 September 1783
- Seven Bridges of Königsberg: the first model in graph theory

The Problem to Solve

Königsberg in Prussia (now Kaliningrad, Russia)

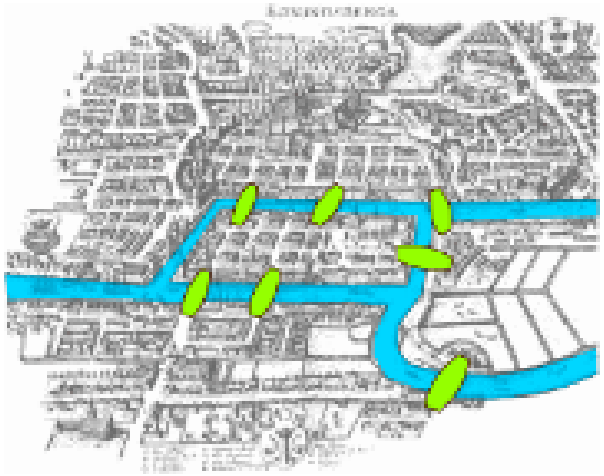


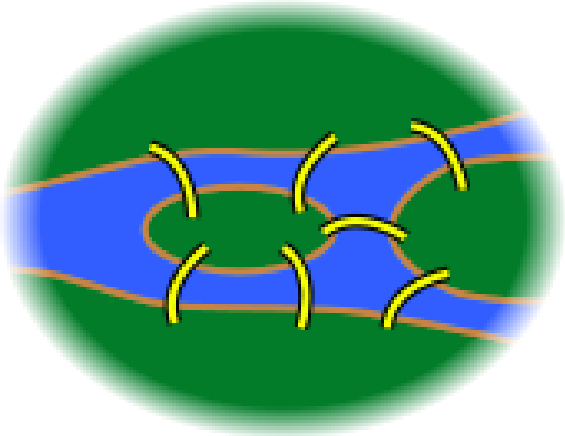
- Seven bridges connecting two mainland portions and an island
- The problem: Is there way to devise a walk through the city that would cross each of those bridges **once and only once**?

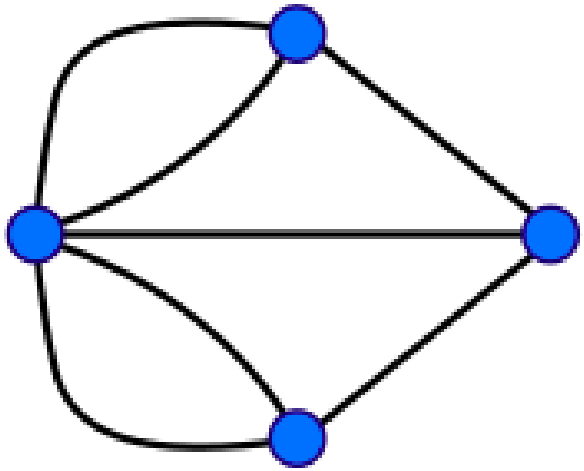
- Unacceptable solutions involve:
  - Reaching an island or mainland bank without using one of the bridges
  - Accessing any bridge without crossing to its other end

Model the Problem Using Graph Theory

Königsberg in Prussia (now Kaliningrad, Russia)



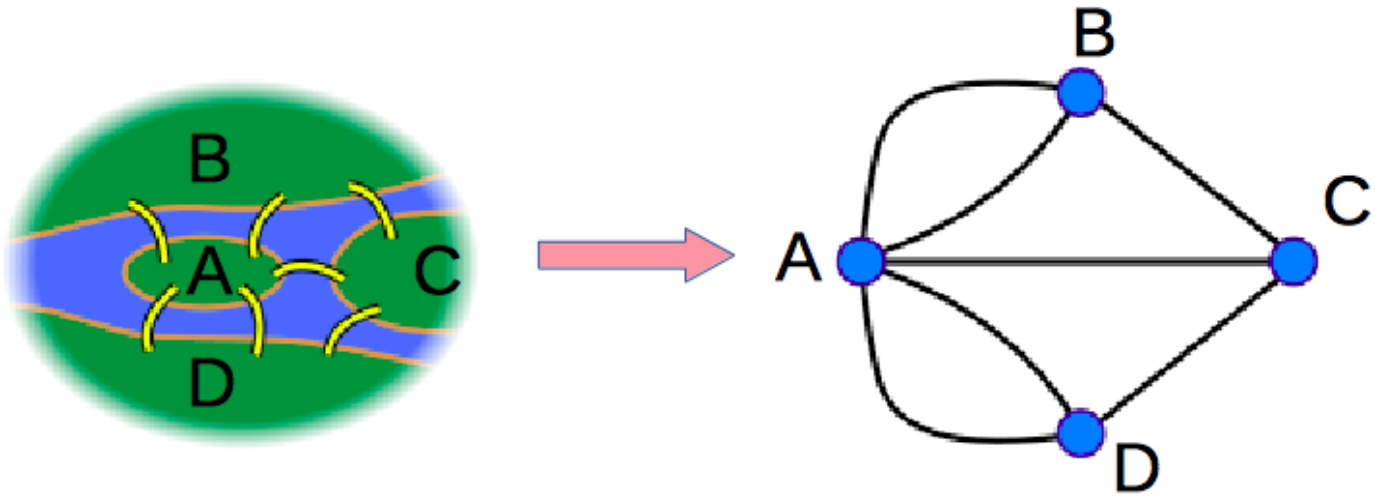




- The problem is converted into a simple graph to study

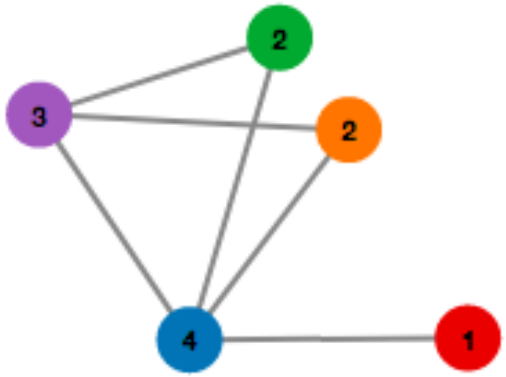
Model the Problem Using Graph Theory

Create Vertices



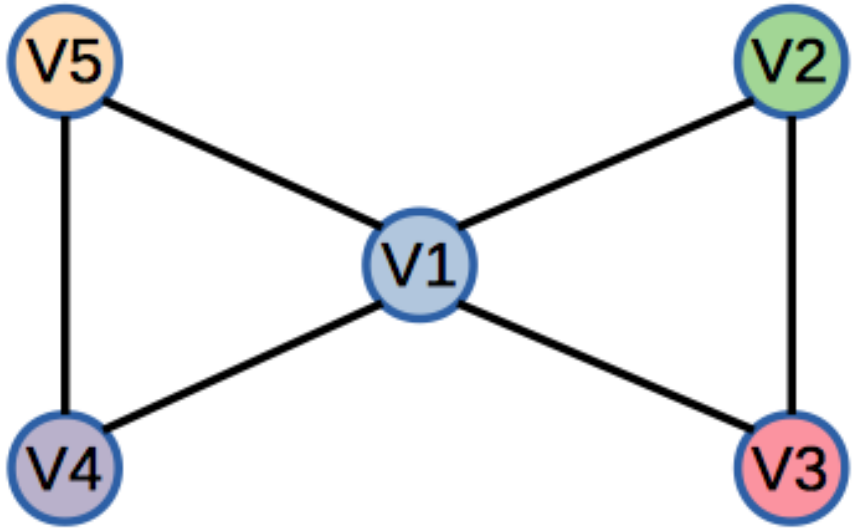
- Create the Vertices and Edges of the Problem

What is Graph Theory?



- **Graph Theory** is the mathematical study of structures which are used to study types of interactions, relationships by pair-wise modeling between objects.
- Graphs are made up of two main elements:
  - *Vertices*: The nodes or vertices
  - *Edges*: The connections between the vertices

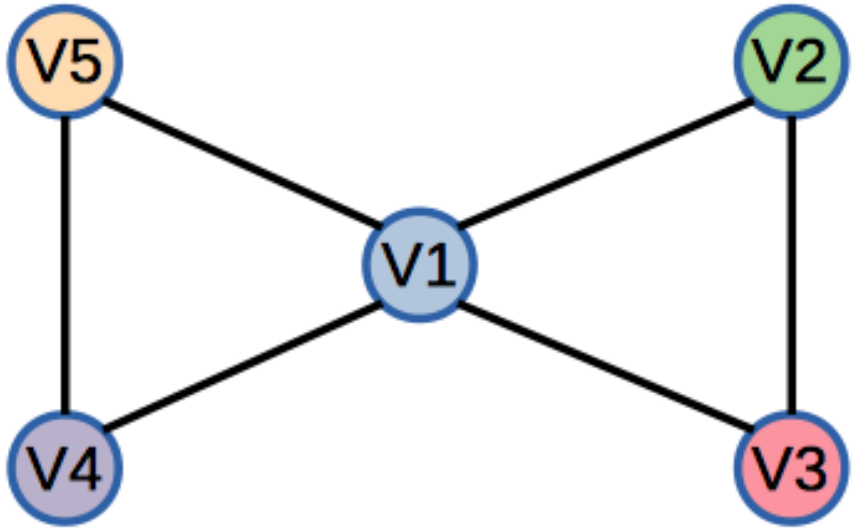
Define a Graph



A Bowtie Graph

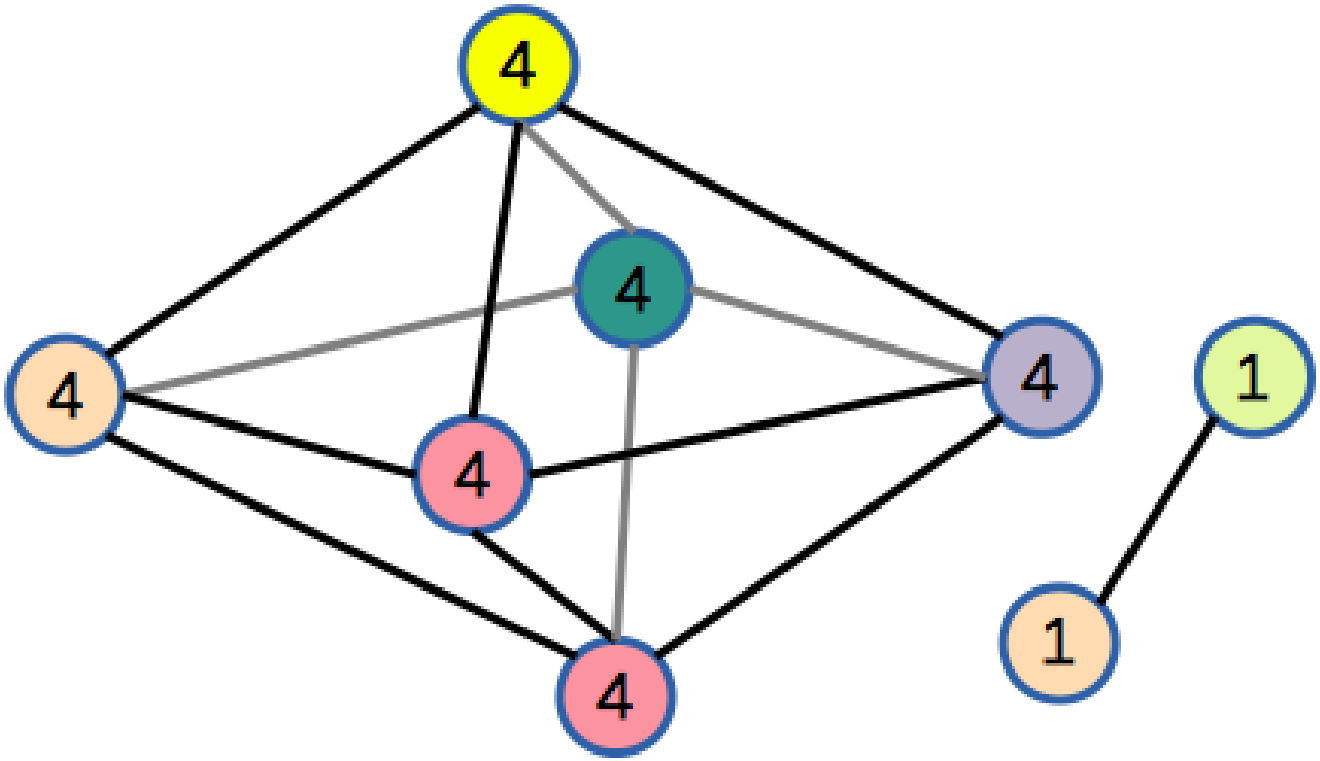
- We define a graph by its vertices and edges: G = (V, E)
  - Vertices: V(G) = $\{V_1, V_2, V_3, V_4, V_5\}$
  - Edges: E(G) = $\{V_1V_2,\ V_2V_3,\ V_3V_1,\ V_4V_1,\ V_5V_1,\ V_4V_5\}$

Degree and Adjacent Vertices



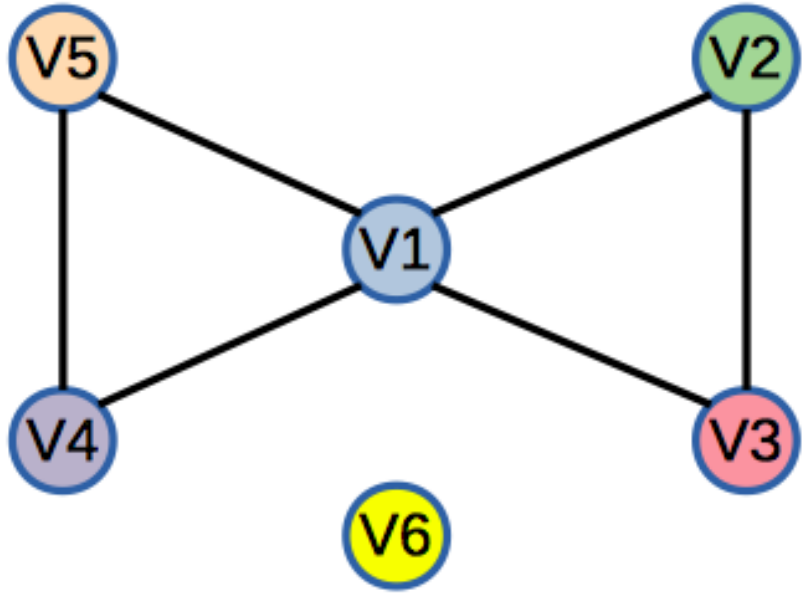
- Adjacency: vertices separated by an edge
- Degree of vertex is the number of its edges to *adjacent vertices*
  - $\text{Deg}(V_1) = 4$
  - $\text{Deg}(V_2) = \text{Deg}(V_3) = \text{Deg}(V_4) = \text{Deg}(V_5) = 2$

Degree Sequences
Disconnected graph



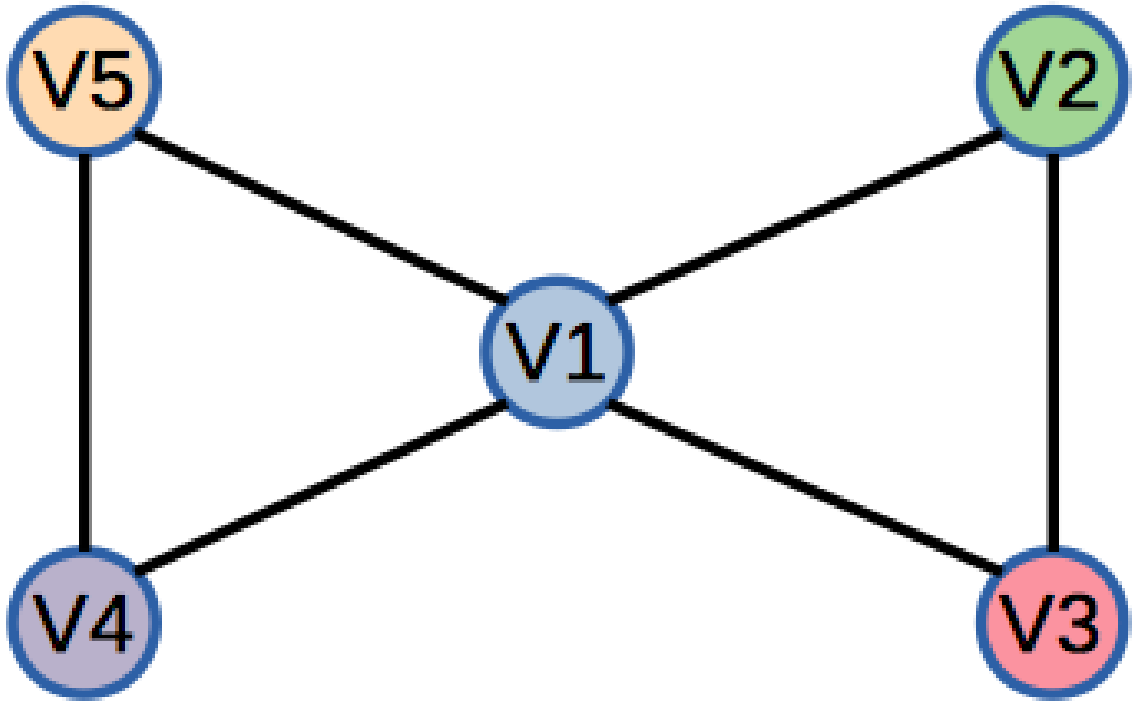
- A sequence of the vertex degrees of G.
- Degree Sequence: (4,4,4,4,4,4,1,1)

Max and Min



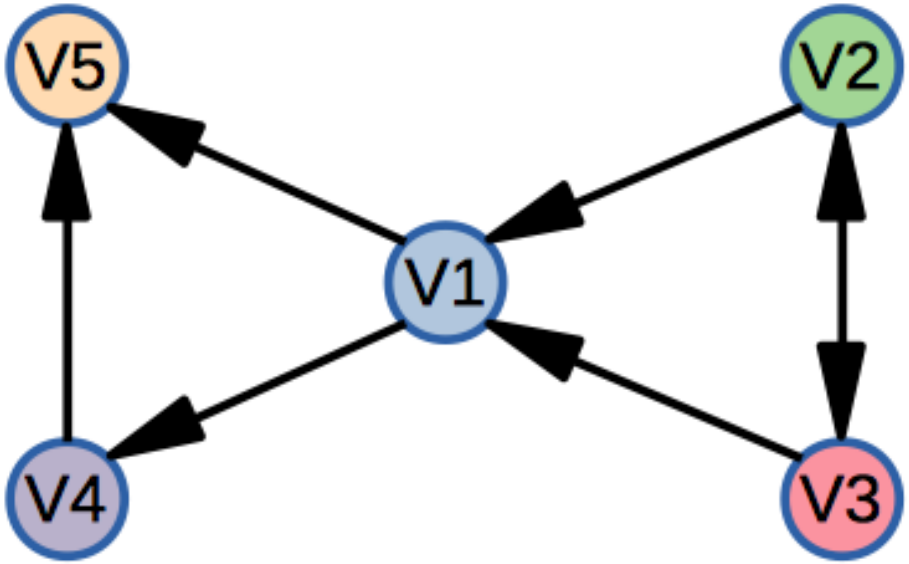
- The vertices of zero degree are called *isolated* vertices (V6) since they do not have any other vertex connected to them.
- Minimum degree (little delta) in a graph: $\delta(G) = 0$
- Maximum degree (big delta) in a graph: $\Delta(G) = 4$
- δ and Δ are properties of a graph, whereas the degree is property of a vertex

Size and Order



- Order: Number of number of vertices in the graph, $O(G) = 5$
- Size: Number of edges: $E(G) = 6$

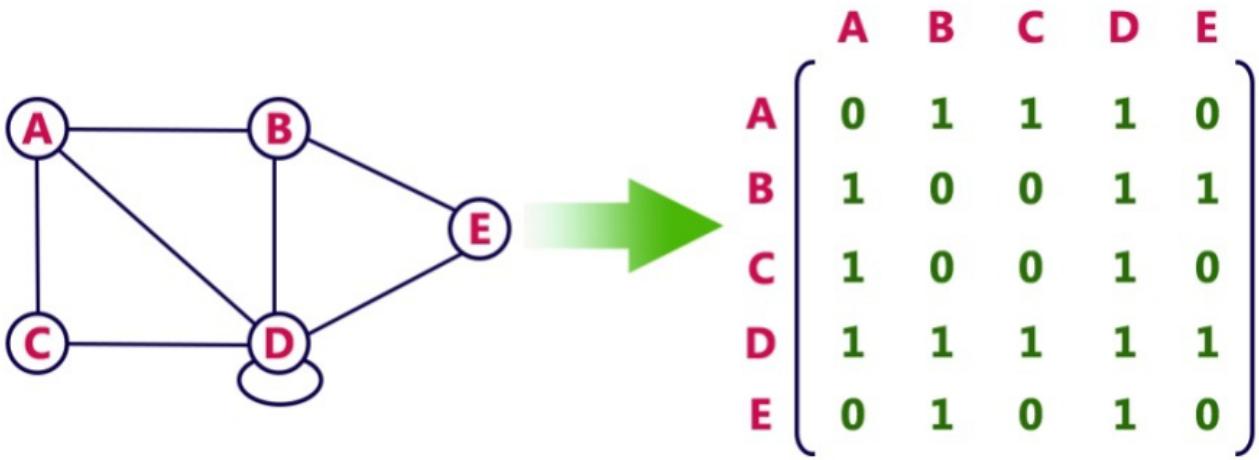
Directed Graph



A Directed Bowtie Graph

- Each vertex is connect by a directional edge.
- Start anywhere and end at the *sink*
- How do you find a sink?

Adjacency Matrices

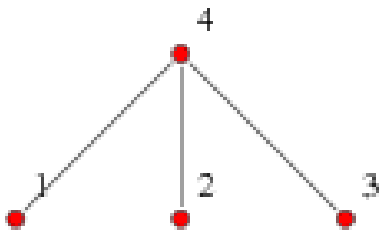


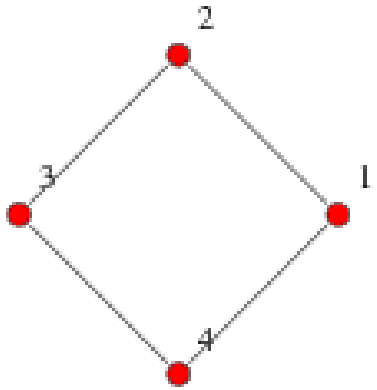
A matrix is used describe adjacent vertices

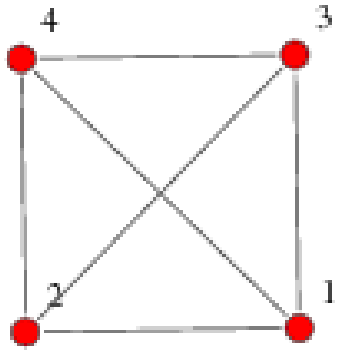
- A matrix contains rows and columns
- Vertices are labelled with a 1 or 0 in position (v_i, v_j) according to whether v_i and v_j are adjacent vertices

Adjacency Matrices

More examples







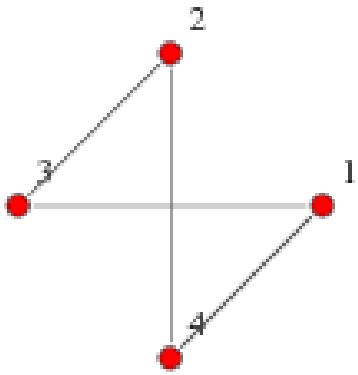
$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

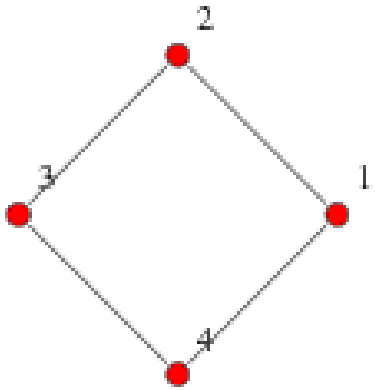
$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

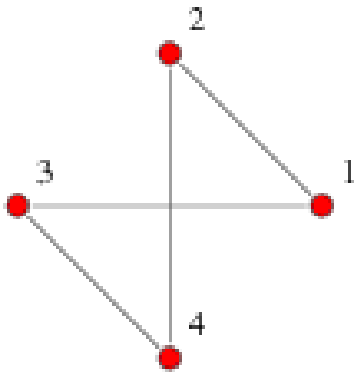
$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Adjacency Matrices

Yet, more examples







$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

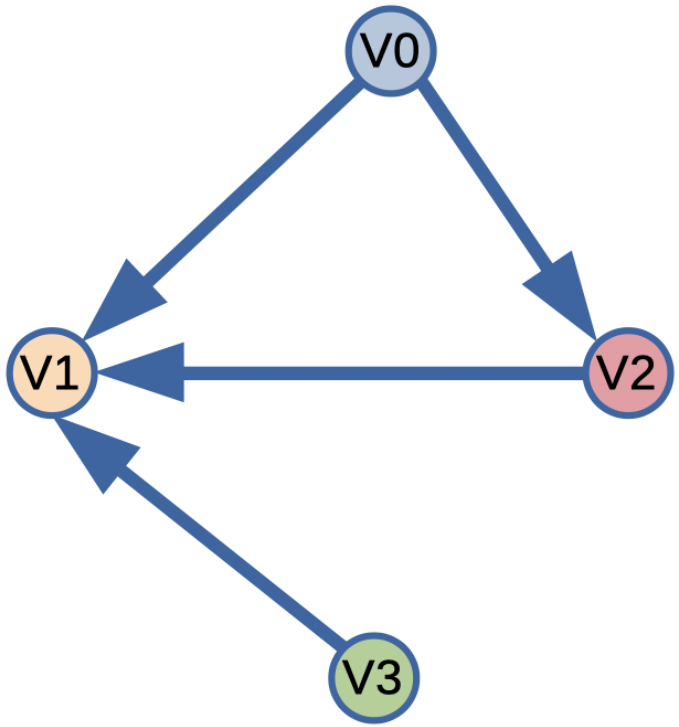
$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

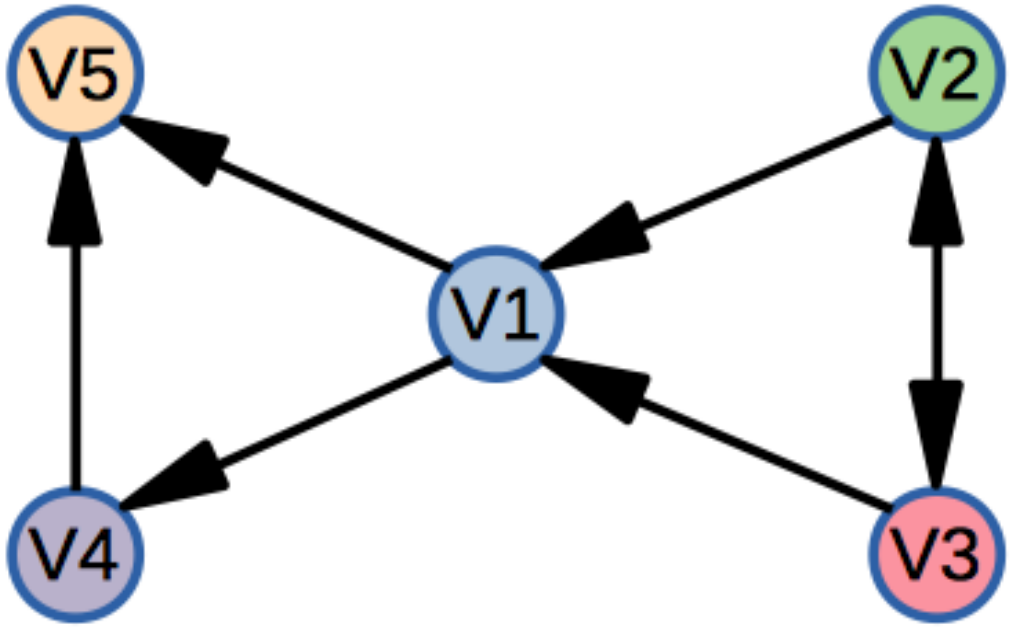
Directional Adjacency Matrices

Read matrix from left to right for edge directions

$\rightarrow$	V0	V1	V2	V3
V0	0	1	1	0
V1	0	0	0	0
V2	0	1	0	0
V3	0	1	0	0



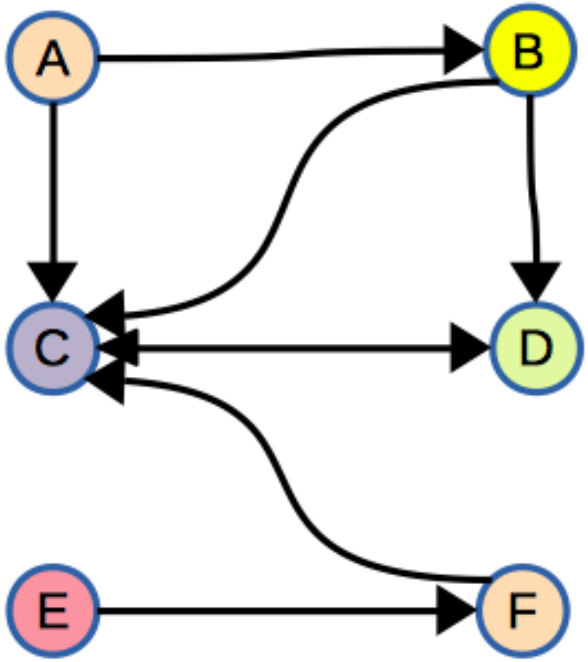
Paths



Find a *Path* through the graph

- Start, End at Vertex V_2, V_5, resp.
- Start, End at Vertex V_3, V_5, resp.
- Possible paths to get there?

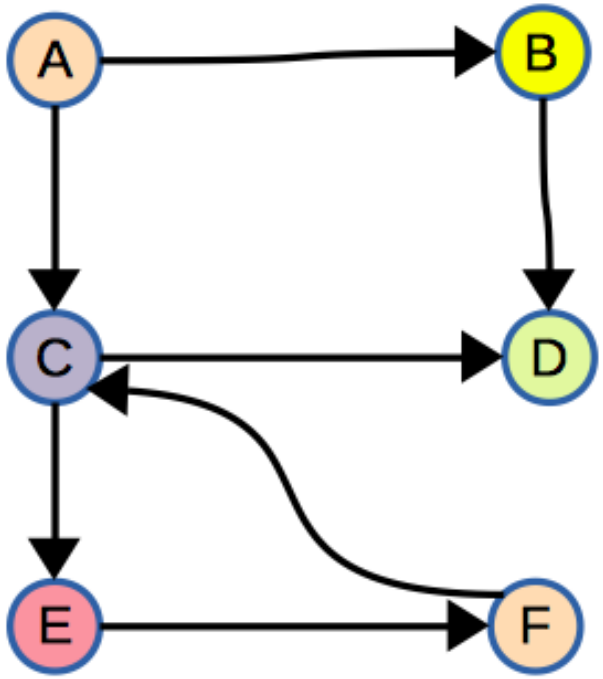
Graph 00: Find the following



Find a *Path* through the graph

- Start, End at Vertex A, D, resp.
- Start, End at Vertex D, F, resp.
- Possible paths to get there?

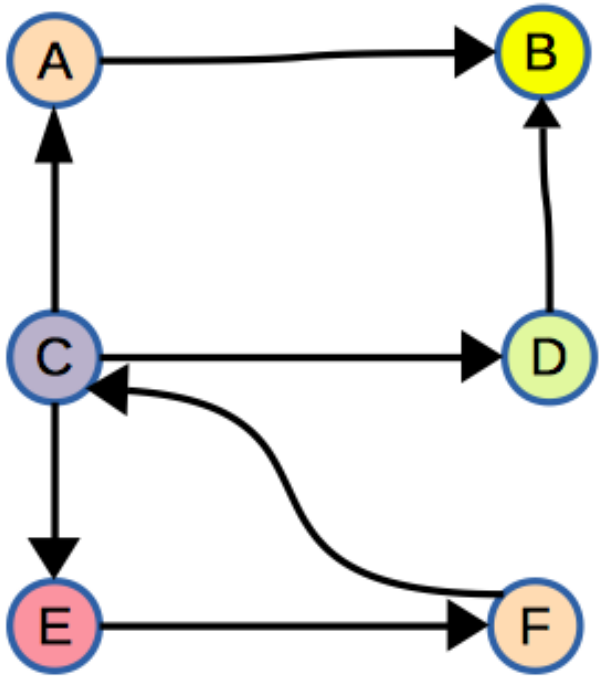
Graph 01: Find the following



Find a *Path* through the graph

- Start, End at Vertex A, C, resp.
- Start, End at Vertex B, E, resp.
- Possible paths to get there?

Graph 02: Find the following

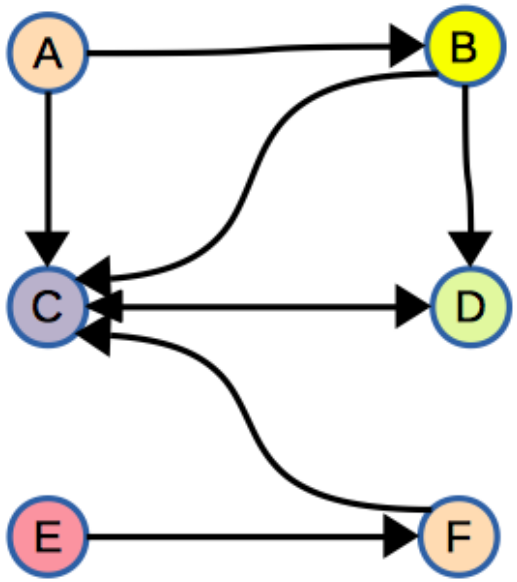


Find a *Path* through the graph

- Start, End at Vertex A, D, resp.
- Start, End at Vertex F, E, resp.
- Possible paths to get there?

Automating the Search in Graph 00

Edit: pathFinder_i.py

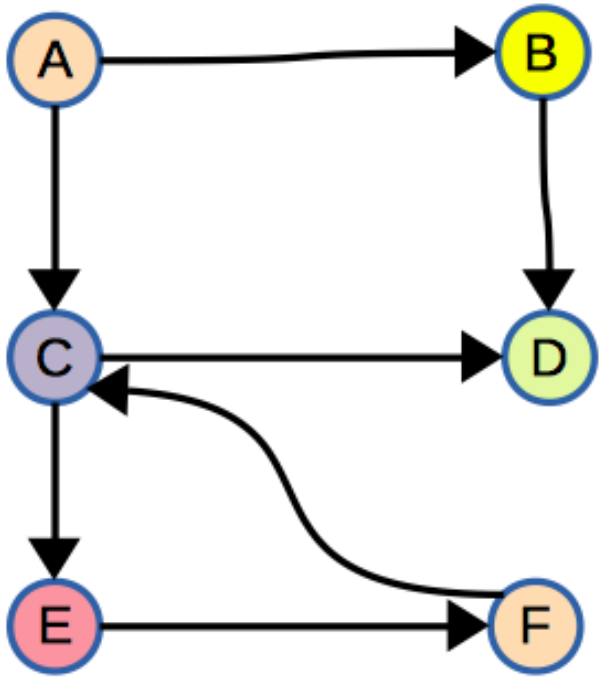


{ node character connects to list of characters}

```
graph = {'A': ['B', 'C'],
         'B': ['C', 'D'],
         'C': ['D'],
         'D': ['C'],
         'E': ['F'],
         'F': ['C']}
```

Automating the Search in Graph 01

Edit: pathFinder_i.py

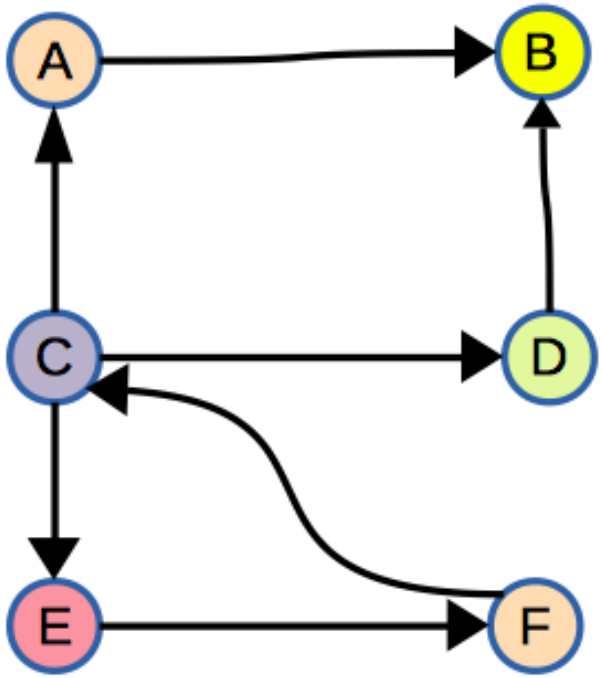


Build the dictionary to contain the graph.

```
graph = {...}??
```

Automating the Search in Graph 02

Edit: pathFinder_i.py



Build the dictionary to contain the graph.

```
graph = {...}??
```