Seven Bridges of Königsberg

Graph Theory

Discrete Structures: CMPSC 102

Oliver BONHAM-CARTER

Fall 2018
Week 9

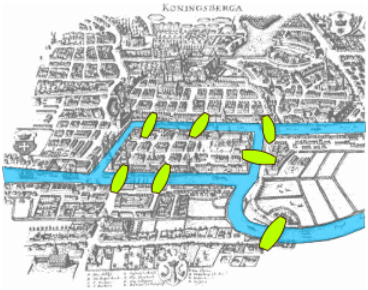
Leonhard Euler

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)

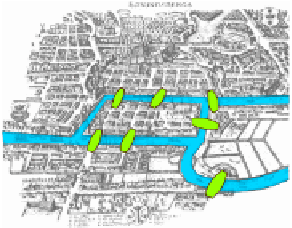


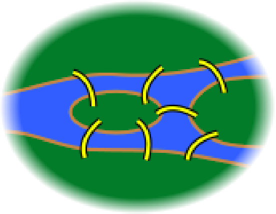
- Seven bridges connecting two mainland portions and an island
- The problem: Is there way to devise a walk through the city that would cross each of those bridges **once and only once**?

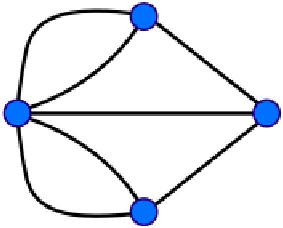
- Unacceptable solutions involve:
  - Reaching an island or mainland bank without using one of the bridges
  - Accessing any bridge without crossing to its other end

Model the Problem Using Graph Theory

Königsberg in Prussia (now Kaliningrad, Russia)



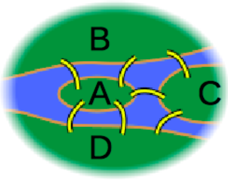


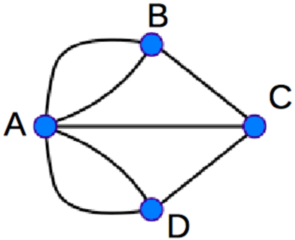


- The problem is converted into a simple graph to study

Model the Problem Using Graph Theory

Create Vertices

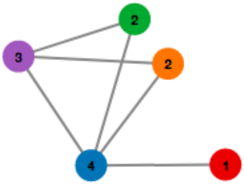




- Create the Vertices and Edges of the Problem

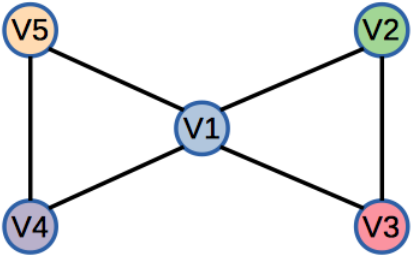
What is Graph Theory?

Seven Bridges of Königsberg
Graph Theory
Degree and Adjacent Vertices
Degree and Adjacent Vertices
Max and Min
Size and Order
Directed
Adjacency Matrices
Path
Consider This
Python Work



- **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

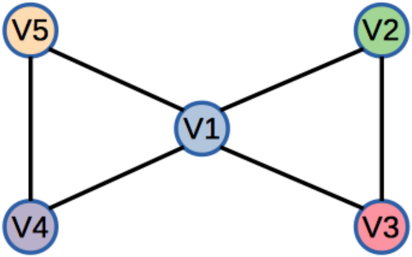
Define a Graph



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) = $\{V_1V_2,\ V_2V_3,\ V_3V_1,\ V_4V_1,\ V_5V_1,\ V_4V_5\}$

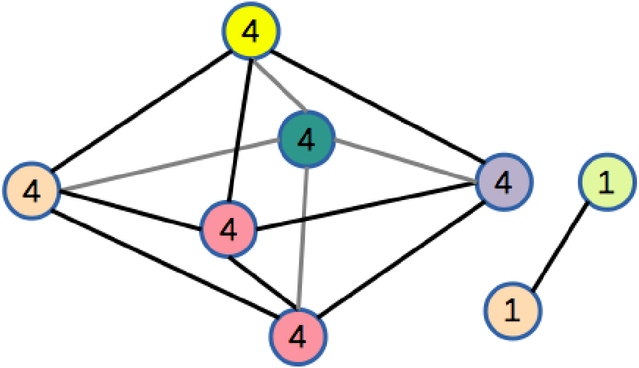
Degree and Adjacent Vertices



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

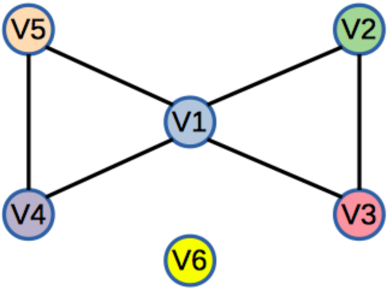
Degree Sequences
Disconnected graph

Seven Bridges of Königsberg
Graph Theory
Degree and Adjacent Vertices
Degree and Adjacent Vertices
Max and Min
Size and Order
Directed
Adjacency Matrices
Path
Consider This
Python Work



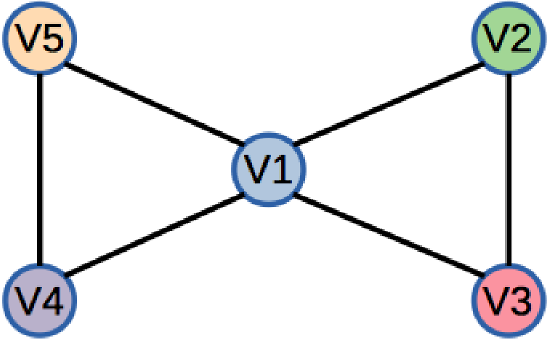
- A sequence of the vertex degrees of G.
- Degree Sequence: (4,4,4,4,4,4,1,1)

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

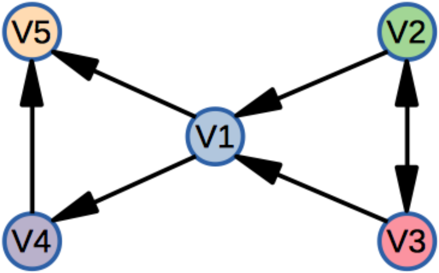
Size and Order



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

Directed Graph

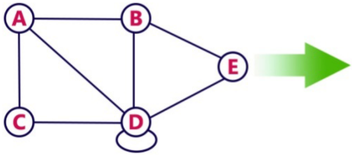
Seven Bridges of Königsberg
Graph Theory
Degree and Adjacent Vertices
Degree and Adjacent Vertices
Max and Min
Size and Order
Directed
Adjacency Matrices
Path
Consider This
Python Work



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



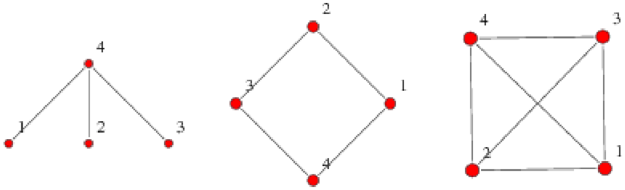
$$
\begin{array}{c c}
 & \begin{array}{ccccc} A & B & C & D & E \end{array} \\
\begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} &
\begin{bmatrix}
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

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

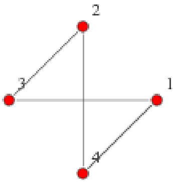
More examples

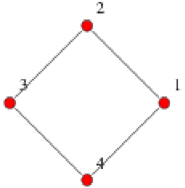


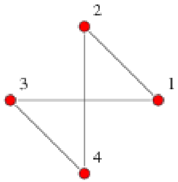
$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Adjacency Matrices

Yet, more examples

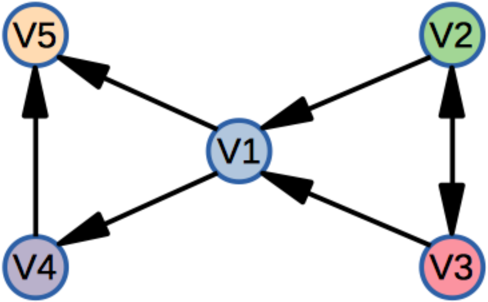






$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

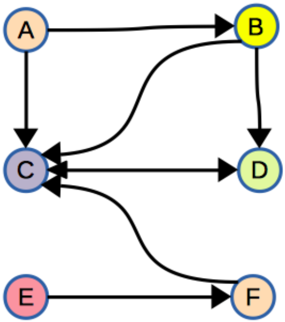
Paths



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?

Act 00: Find the following

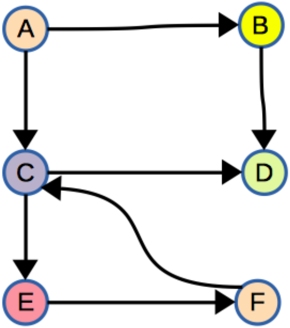


Find a *Path* through the graph

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

Seven Bridges of Königsberg
Graph Theory
Degree and Adjacent Vertices
Degree and Adjacent Vertices
Max and Min
Size and Order
Directed
Adjacency Matrices
Path
Consider This
Python Work

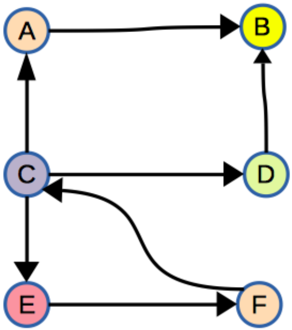
Act 01: Find the following



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?

Act02: Find the following



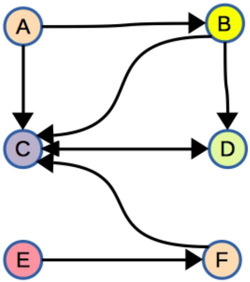
Find a *Path* through the graph

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

Finding Paths in A Graph 00

pathFinder_part02.py

Seven Bridges of Königsberg
Graph Theory
Degree and Adjacent Vertices
Degree and Adjacent Vertices
Max and Min
Size and Order
Directed
Adjacency Matrices
Path
Consider This
Python Work



{ node character connects to list of characters}

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

Participation and GitHub

Completed code due by the end of class today

Seven Bridges of Königsberg
Graph Theory
Degree and Adjacent Vertices
Degree and Adjacent Vertices
Max and Min
Size and Order
Directed
Adjacency Matrices
Path
Consider This
Python Work

Instructions

- Run the GitHub commands in your participation repository
- Note, the repository address was: https://classroom.github.com/a/X9XPMFnb
- Find the file: src/pathFinder_part02.py in the participation repository
- Add the dictionaries (see next slides) for the other graphs into the code and run it to see how Python is able to a path between vertices and all paths.
- Be sure to check the `TODO` tags to see where your dictionaries go in the code

Two Git commands: The first is used only once.

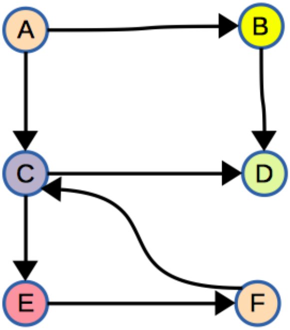
```
git remote add download
  git@github.com:Allegheny-Computer-Science-102-F2018/cs102_participation_starters

git pull download master
```

Finding Paths in A Graph 01

pathFinder_part02.py

Seven Bridges of Königsberg
Graph Theory
Degree and Adjacent Vertices
Degree and Adjacent Vertices
Max and Min
Size and Order
Directed
Adjacency Matrices
Path
Consider This
Python Work



Build the dictionary to contain the graph.

```
graph = ??
```

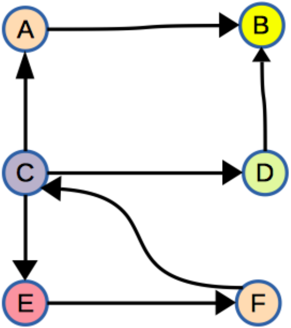
Finding Paths in A Graph 02

pathFinder_part02.py

Seven Bridges of Königsberg

Graph Theory

- Degree and Adjacent Vertices
- Degree and Adjacent Vertices
- Max and Min
- Size and Order
- Directed
- Adjacency Matrices
- Path
- Consider This
- Python Work



Build the dictionary to contain the graph.

```
graph = ??
```