

# Introduction to Graph Theory

Brian Eft

Tech Elevator Columbus

June 29, 2021

# Outline

## Introduction

Basic Definitions

Motivation

## Concepts

Euler Trails

Graph Planarity

Graph Algorithms

## Conclusion

Real-World Problems

## References

# Introduction

## Definition

**Graph Theory** is a field of mathematics which studies mathematical structures called *graphs*.

# What is a Graph?

## Definition

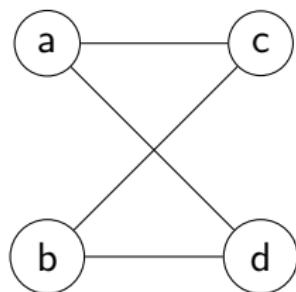
A **graph** consists of two sets,  $V$  and  $E$ .  $V$  is a set of *vertices*.  $E$  is a set of unordered pairs of vertices, called *edges*.

# What is a Graph?

## Definition

A **graph** consists of two sets,  $V$  and  $E$ .  $V$  is a set of *vertices*.  $E$  is a set of unordered pairs of vertices, called *edges*.

## Example



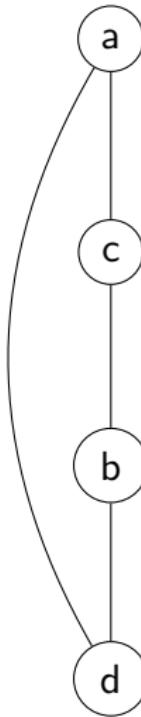
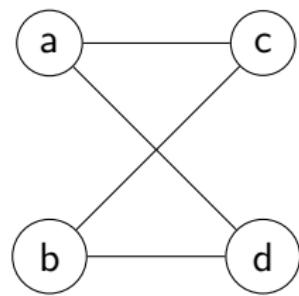
Graph  $G$  has a vertex set  $V$  and edge set  $E$ .

$$V = \{a, b, c, d\}$$

$$E = \{\{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}\}$$

$$E = \{ac, ad, bc, bd\}$$

# Graph Representations



# Simple, Multi, and Pseudographs

- ▶ A **simple graph** contains no duplicate edges or loops. A loop is when a vertex connects to itself.

# Simple, Multi, and Pseudographs

- ▶ A **simple graph** contains no duplicate edges or loops. A loop is when a vertex connects to itself.
- ▶ If we allow multiple edges between a pair of vertices, we get a **multigraph**.



# Simple, Multi, and Pseudographs

- ▶ A **simple graph** contains no duplicate edges or loops. A loop is when a vertex connects to itself.
- ▶ If we allow multiple edges between a pair of vertices, we get a **multigraph**.

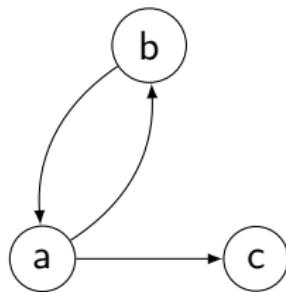


- ▶ If we allow loops, we get a **pseudograph**.



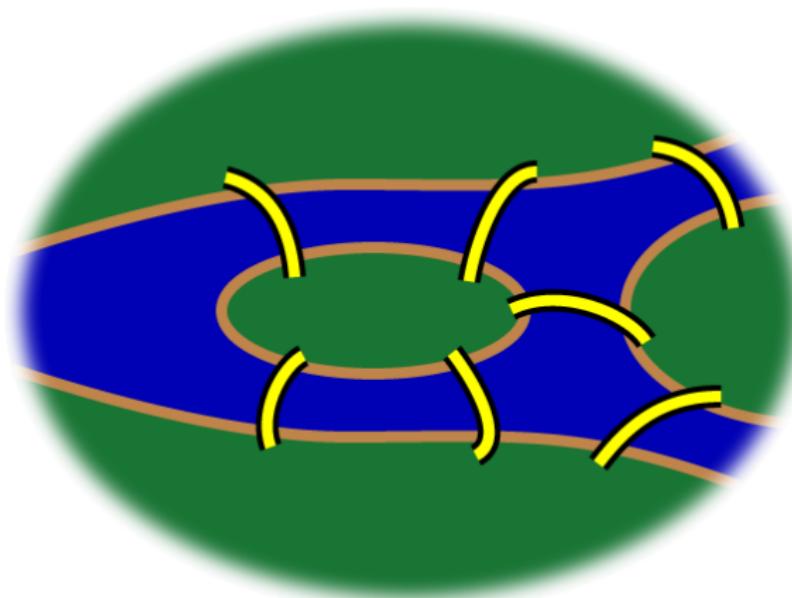
# Directed Graphs

If we make the edges of the graph a set of ordered pairs, we get a **directed graph**, or digraph.



$$E = \{(a, b), (b, a), (a, c)\}$$

# Seven Bridges of Königsberg



Find a walk through the city that crosses each bridge exactly once.

# Three Utilities Problem



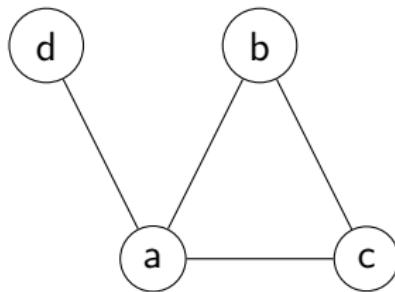
Connect every house to water, gas, and electric without crossing any lines.

# Escaping a Maze



## Adjacency and Degree

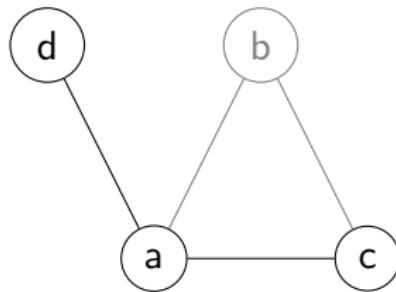
- ▶ Two vertices  $u$  and  $v$  are said to be **adjacent** if there is an edge  $uv$  connecting them. If there is no edge  $uv$ , then they are nonadjacent.
- ▶ The **degree** of a vertex  $u$  is the number of other vertices adjacent to  $u$ .



$$\deg(a) = 3$$

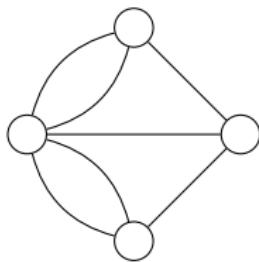
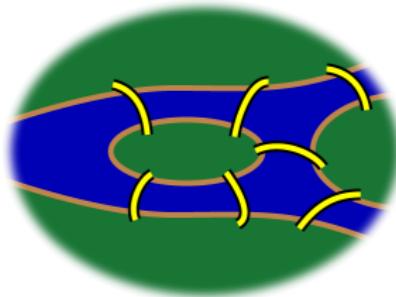
## Walks, Paths, and Trails

- ▶ A **walk** in a graph is a sequence of vertices such that each consecutive pair of vertices in the sequence are adjacent. The vertices need not be distinct.
- ▶ A walk with distinct vertices is called a **path**.
- ▶ A walk with distinct edges is called a **trail**.

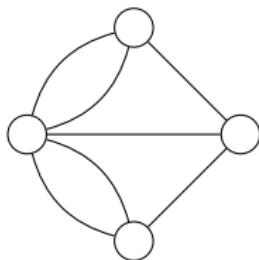
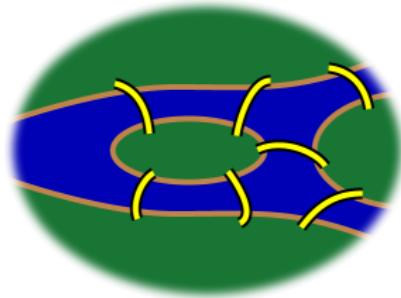


$d, a, c$

# Seven Bridges of Königsberg Revisited

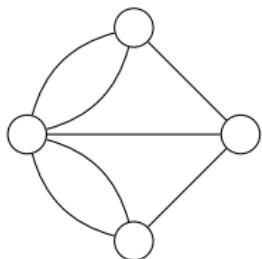
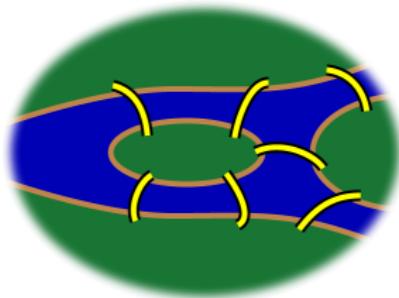


# Seven Bridges of Königsberg Revisited



- ▶ Is there a trail which includes every edge?
- ▶ Known today as an Euler walk or Eulerian trail
- ▶ Leonhard Euler (1707–1783)

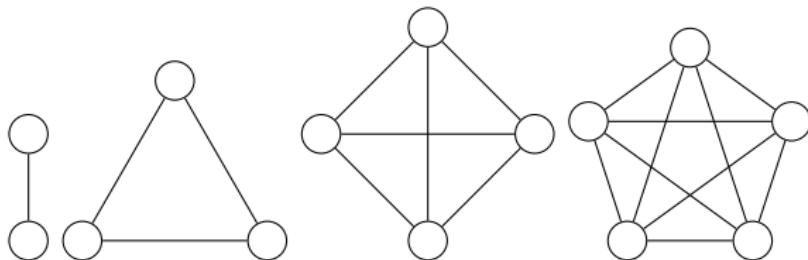
# Seven Bridges of Königsberg Revisited



- ▶ Is there a trail which includes every edge?
- ▶ Known today as an Euler walk or Eulerian trail
- ▶ Leonhard Euler (1707–1783)
- ▶ An undirected graph has an Euler walk if and only if exactly zero or two vertices have odd degree.

# Complete Graphs

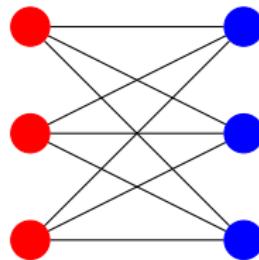
A graph is **complete** if every vertex is adjacent to every other vertex.



$K_2$ ,  $K_3$ ,  $K_4$ , and  $K_5$

## Bipartite Graphs

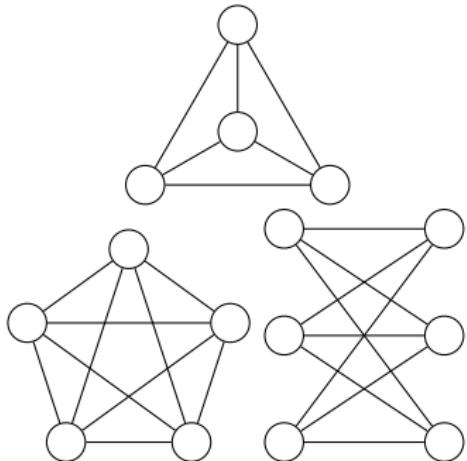
A graph is **bipartite** if all vertices can be colored red or blue such that every edge connects a red vertex to a blue vertex.



$$K_{3,3}$$

The complete bipartite graph of 3 red vertices and 3 blue vertices.

# Graph Planarity



A graph is **planar** if it can be drawn on a plane in such a way that edges do not intersect, except at vertices.

## Kuratowski's Theorem

*A graph  $G$  is planar if and only if it contains no subdivision of  $K_5$  or  $K_{3,3}$ .*

# Labyrinth of Daedalus



- ▶ Graph = maze  
Vertex = intersection  
Edge = passage
- ▶ How to search for the exit in a maze without getting lost and wandering in circles?
- ▶ Legend of Theseus and the Minotaur

# Graph Search

## Trémaux Exploration

1. *Follow any unmarked passage, unspooling string behind you*
2. *Mark all passages and intersections when first visiting them*
3. *Backtrack by rewinding the string when you approach a marked intersection*
4. *Backtrack when no unvisited options remain at an intersection*
5. *Repeat until the exit is found, or the entire maze is explored*

# Graph Search

## Trémaux Exploration

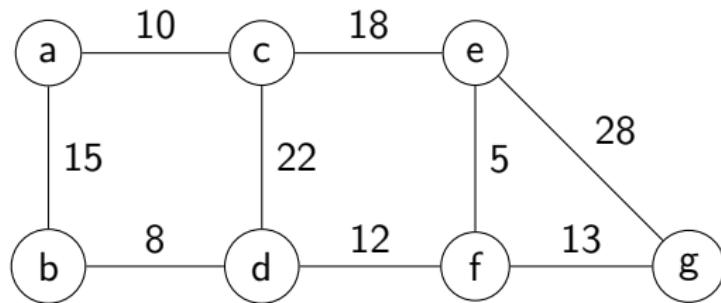
1. Follow any unmarked passage, unspooling string behind you
2. Mark all passages and intersections when first visiting them
3. Backtrack by rewinding the string when you approach a marked intersection
4. Backtrack when no unvisited options remain at an intersection
5. Repeat until the exit is found, or the entire maze is explored

## Depth-First Search

1. Mark vertex as visited
2. Recursively visit all adjacent unmarked vertices

# Shortest Path

An **edge-weighted graph** is a graph where each edge has a weight or cost.



A *shortest path algorithm* finds a path between two vertices with the minimum cost.

# Real-World Problems

- ▶ Maps
- ▶ Webpages
- ▶ Electric Circuits
- ▶ Social Networks
- ▶ Scheduling
- ▶ Commerce

Arbitrage : shortest-path algorithm

## References

-  Harris, John M., Hirst, Jeffry L., and Michael J. Mossinghoff. 2008. *Combinatorics and Graph Theory*. New York: Springer.
-  Sedgewick, Robert, and Kevin Wayne. 2011. *Algorithms*. Upper Saddle River: Pearson Education.