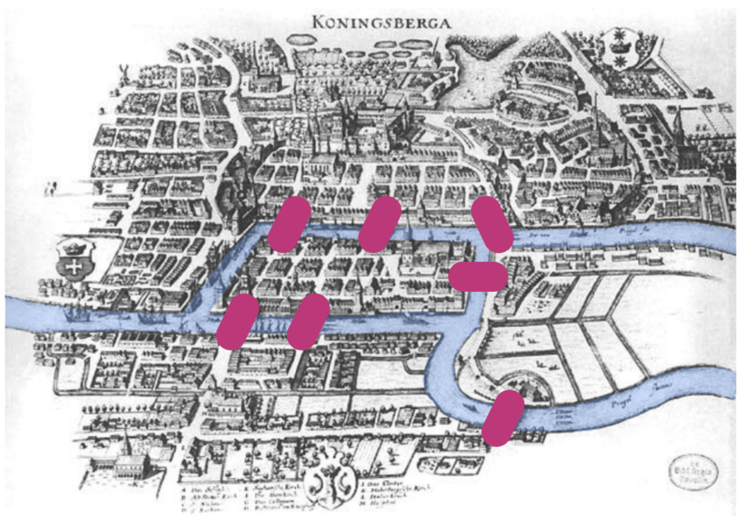
Grade 9/10 Math Circles
October 25, 2023
Graph Theory

Warm-up: Bridges of Königsberg



Wikimedia Commons

Let's explore Königsberg!
Find a walk that allows us to cross each bridge EXACTLY once.
We must stay within the city (no going outside the image).

Why Graph Theory?

Graph Theory has applications to many real-world fields: transportation of supplies, chemistry, traffic flow, social networks, computer networks, knot theory, linguistics, and more! But, above all, graph theory is a very cool branch of mathematics that is fun to explore.

Basic Definitions

Before we get too far in, we need a few definitions.

Definition 1. A **set** is a collection of objects with:

1. No repeats
2. No order

Example: Sets

Each of the following are examples of **sets**:

- $\{a, b, c\}$
- $\{\bigcirc, \square, \triangle, \bigstar\}$
- $\{-1, 1, 2, 4, 5, 0\}$

Note 1: $\{1, 1, 2, 3\} = \{1, 2, 3\}$ **Note 2:** $\{1, 2, 3\} = \{2, 3, 1\} = \{3, 2, 1\}$

Definition 2. A **graph** is a set of vertices paired with a set of unordered pairs of distinct vertices, called edges.

Example: Graph

$G = (V, E)$

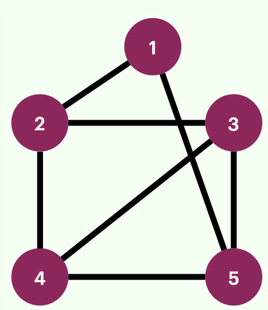
$V = \{1, 2, 3, 4, 5\}$

$E = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{4, 5\}\}$

Draw a visual representation below!

Definition 3. A vertex is a **neighbour** of another vertex if there is an edge between them.

Example: Neighbours



In the graph, vertex 1 and vertex 2 are neighbours. Vertex 1 and vertex 3 are not.

Exercise 1

Create a 'word graph', where the vertices are:

- Bat
- Cot
- Pan
- Pin
- Pot
- Rot
- Cat
- Mat
- Pat
- Pit
- Rat
- Sit

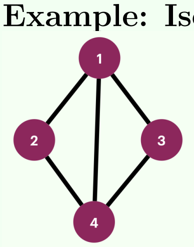
and two vertices are neighbours if they differ by only one letter.

Note 3: We are only going to consider *simple* graphs today:

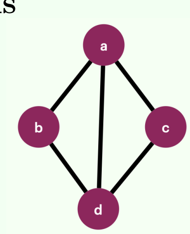
- No edge direction
- No duplicate edges
- No loops (edges which go back to the same vertex)

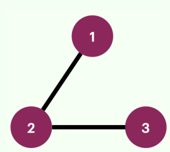
Definition 4. Two graphs G_1 and G_2 are **isomorphic** if it is possible to relabel G_1 to get G_2.

Example: Isomorphic Graphs

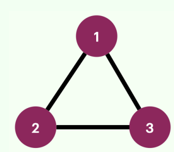


is isomorphic to



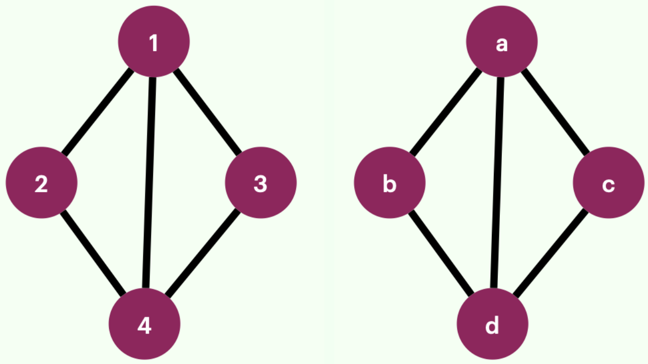


is not isomorphic to



Definition 5. An **isomorphism** is a map which defines the relabelling of a graph.

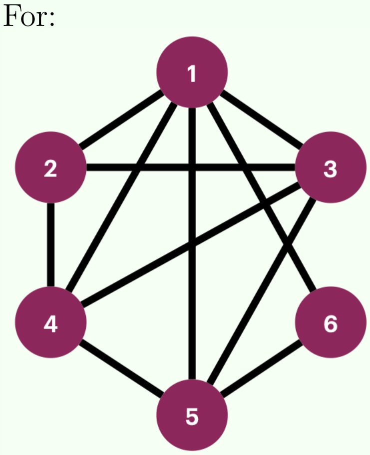
Example: Graph Isomorphisms
For:



A corresponding isomorphism is $1 = a$, $2 = b$, $3 = c$, $4 = d$.

Definition 6. The **degree** of a vertex is the number of edges which connect to the vertex.

Example: Degree
For:



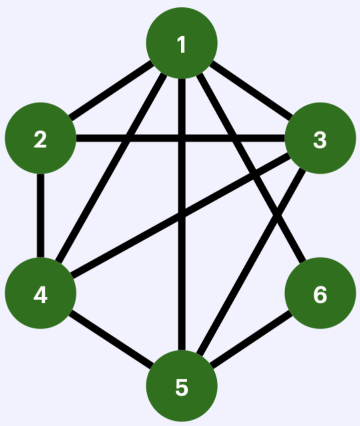
$\deg(1) = 5$, $\deg(2) = 3$, $\deg(3) = \deg(4) = \deg(5) = 4$ and $\deg(6) = 2$

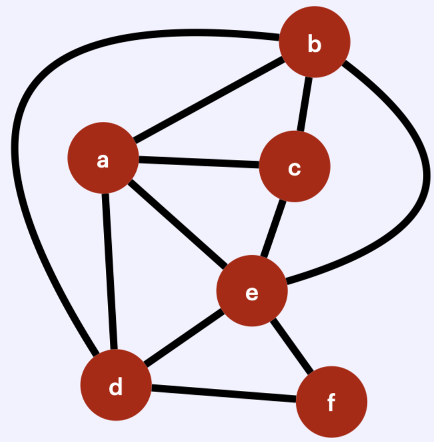
Finding an isomorphism:

- Check the number of vertices is the same
- Check vertex degree: a vertex can only be paired with a vertex of same degree
- Check already re-labelled neighbours: a vertex must have the same (corresponding) neighbours after re-labelling

Exercise 2

Show that the following two graphs are isomorphic

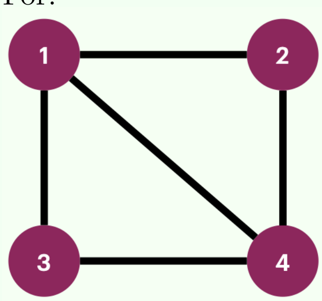




Theorem 7. ***Handshaking Lemma.*** *For any graph G, the sum of the degrees of all vertices is equal to 2 times the number of edges.*

Example: Handshaking Lemma

For:



$\deg(1) = 3$, $\deg(2) = 2$, $\deg(3) = 2$ and $\deg(4) = 3$.

Then, the sum of the degrees is $10 = 2 * 5 = 2$ times the number of edges.

Proof: Note that each edge has two ends. When we sum the degrees, each edge end is counted once, meaning that each edge is counted twice. ☐

Example: Applying Handshaking Lemma 1
Find the number of edges in a graph with vertices $\{a, b, c, d, e\}$ who have degrees 3, 4, 3, 2, 2, respectively.
Then:

$$3 + 4 + 3 + 2 + 2 = 2 * (\text{number of edges}) \qquad \text{(by HL)}$$
$$\implies 14 = 2 * (\text{number of edges})$$
$$\implies \text{number of edges} = 7$$

Example: Applying Handshaking Lemma 2
Find the number of vertices in a graph where all vertices have degree 3 and the graph has 51 edges.
Then, $3 * (\text{number of vertices}) = \text{sum of degrees}$.

$$\implies 3 * (\text{number of vertices}) = 2 * 51 \qquad \text{(by HL)}$$
$$\implies 3 * (\text{number of vertices}) = 102$$
$$\implies \text{number of vertices} = \frac{102}{3} = 34$$

Exercise 3

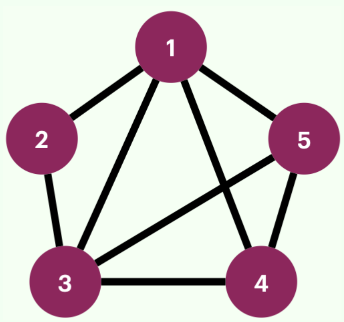
Find a graph with vertex set $\{a, b, c, d\}$ where $\deg(a) = 1$, $\deg(b) = 3$, $\deg(c) = 2$, $\deg(d) = 1$.

Theorem 8. *The number of vertices of odd degree in a graph is even.*

Definition 9. A **walk** is a sequence of vertices and edges that lead from one vertex to another.

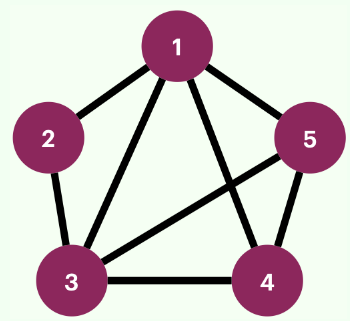
Definition 10. A **path** is a walk where we do not return to the same vertex twice.

Example: Walk
For:



$1 \to 2 \to 3 \to 1 \to 4 \to 5$ is a walk 1 to 5.

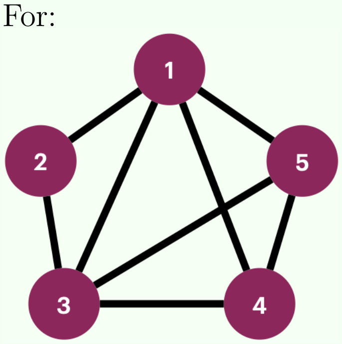
Example: Path
For:



$1 \to 2 \to 3 \to 5$ is a path from 1 to 5.
$1 \to 3 \to 1 \to 5$ is not a path.

Definition 11. A **cycle** is a path which starts and ends at the same vertex.

Example: Cycle
For:



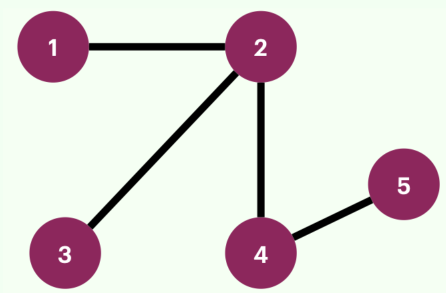
$1 \to 3 \to 5 \to 1$ is a cycle.

What are the other cycles in this graph?

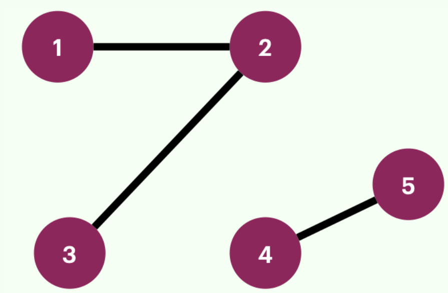
Definition 12. The **tree** is a graph that is **connected** (all vertices have paths to each other) and has no cycles.

Example: Tree

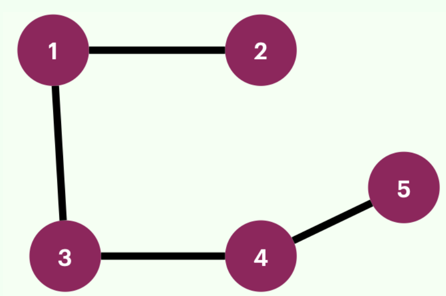
Tree:



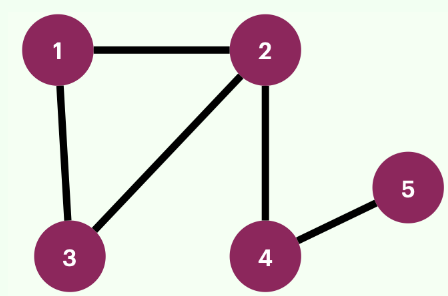
Not a Tree:



Tree:

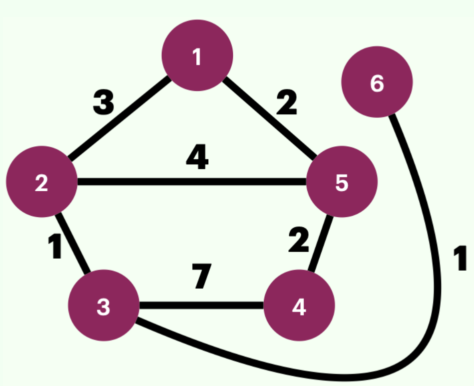


Not a Tree:



Definition 13. A **weighted graph** is a graph where each of the edges have weights or cost assigned.

Example: Weighted Graph



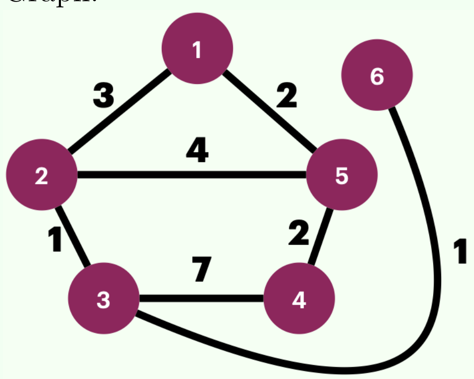
How much does it cost to travel from vertex 1 to vertex 6?

$1 \to 2 \to 3 \to 6 : 3 + 1 + 1 = 5$

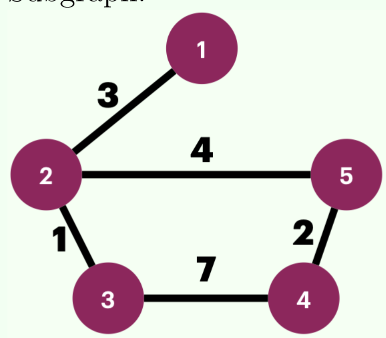
Definition 14. A **subgraph** is a graph whose edges and vertices are all part of a possibly larger graph.

Example: Subgraph

Graph:



Subgraph:



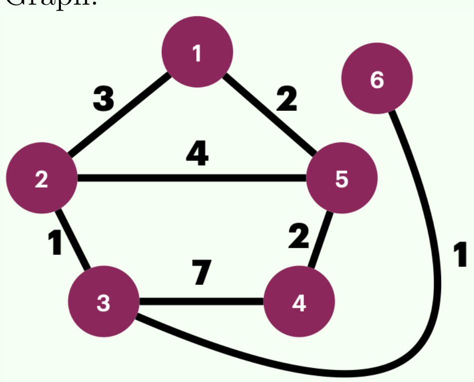
Definition 15. A **spanning tree** is a subgraph with the following properties:

- it is a tree
- all vertices from the original graph must be included

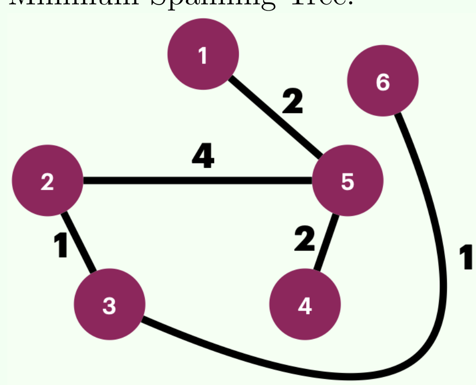
Definition 16. A **minimum spanning tree (MST)** is a spanning tree of lowest cost.

Example: MST

Graph:



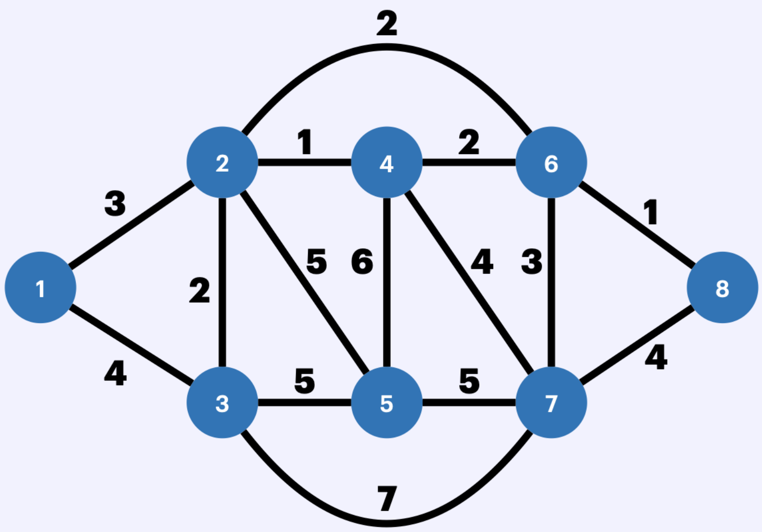
Minimum Spanning Tree:



MST with cost $2 + 4 + 1 + 2 + 1 = 10$

Exercise 4

Find an MST of the graph below:



Prim's Algorithm

Prim's Algorithm is a way of finding an MST of a graph.

1. Pick a vertex in your graph - it is the start of our spanning tree
2. Consider all edges that are connected to exactly one vertex in the current tree
3. Pick the edge with the smallest weight and add it, along with the new vertex, to the tree
4. Repeat steps 2-3 until all vertices are in the tree

Exercise 5

Use Prim's Algorithm to find a different MST of the graph below: