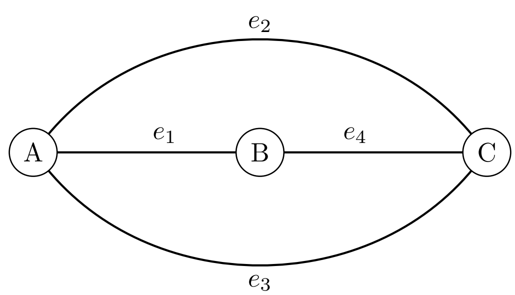
An Introduction to Graph Theory

Prepared by Mark on August 12, 2022
Based on a handout by Oleg Gleizer

Part 1: Graphs

A *graph* is a collection of nodes (vertices) and connections between them (edges). If an edge e connects the vertices v_i and v_j, then we write $e = v_i, v_j$. An example is below.



More formally, a graph is defined by a set of vertices $\{v_1, v_2, ...\}$, and a set of edges $\{\ \{v_1, v_2\}, \{v_1, v_3\}, ...\ \}$.

If the order of the vertices in an edge does not matter, a graph is called *undirected*. A graph is called a *directed graph* if the order of the vertices does matter. For example, the (undirected) graph above has three vertices, A, B, and C, and four edges, $e_1 = \{A, B\}$, $e_2 = \{A, C\}$, $e_3 = \{A, C\}$, and $e_4 = \{B, C\}$.

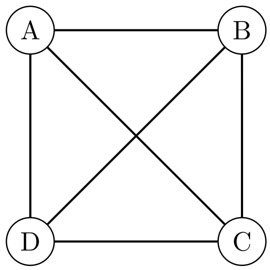
Problem 1:
Draw an undirected graph that has the vertices A, B, C, D, and E and the edges $\{A, B\}$, $\{A, C\}$, $\{A, D\}$, $\{A, E\}$, $\{B, C\}$, $\{C, D\}$, and $\{D, E\}$ in the space below.

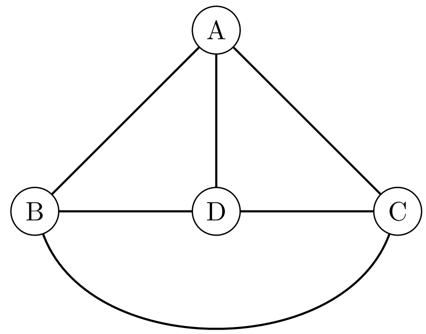
Graphs are useful for solving many different kinds of problems. Most situations that involve some kind of "relation" between elements can be represented by a graph.

Also, note that the graphs we're discussing today have very little in common with the "graphs" of functions you're used to seeing in your math classes.

Graphs are fully defined by their vertices and edges. The exact position of each vertex and edge doesn't matter—only which nodes are connected to each other. As such,two equivalent graphs can look very different.

Problem 2:
Prove that the graphs below are equilvalent by comparing the sets of their vertices and edges.





Definition 1:
The degree $D(v)$ of a vertex v of a graph is the number of the edges of the graph connected to that vertex.

Theorem 1:
For any graph, the sum of the degrees of the vertices equals twice the number of the edges.

Problem 3:
Prove Theorem 1

Problem 4:
Prove the following corollary of Theorem 1:
The number of vertices of odd degree in any graph is even.

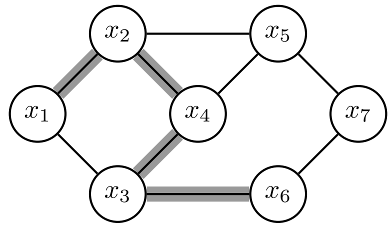
Problem 5:
One girl tells another, "There are 25 kids in my class. Isn't it funny that each of them has 5 friends in the class?" "This cannot be true," immediately replies the other girl. How did she know?

Part 1:
Let us represent the children in the first girl's class as vertices of a graph. Let us represent the friendships as the graph's edges. What is the degree of each vertex?

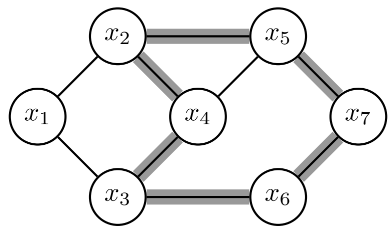
Part 2:
So how did the second girl know right away?

Part 2: Paths and cycles

A *path* in a graph is, intuitively, a sequence of edges: $\{ \{x_1, x_2\}, \{x_2, x_4\}, ... \}$. For example, I've highlighted one possible path in the graph below.



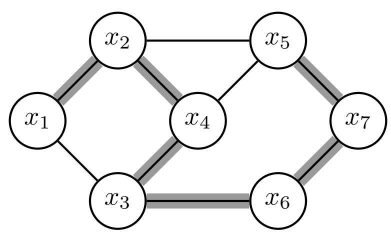
A *cycle* is a path that starts and ends on the same vertex:



A *Eulerian** path is a path that traverses each edge exactly once.
A Eulerian cycle is a cycle that does the same.

Similarly, a *Hamiltonian* path is a path in a graph that visits each vertex exactly once, and a Hamiltonian cycle is a closed Hamiltonian path.

An example of a Hamiltonian path is below.

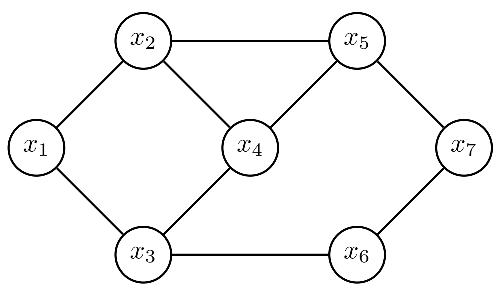


*Pronounced "oiler". These terms are named after a great Swiss mathematician, Leonhard Euler (1707-1783), considered by many as the founder of graph theory.

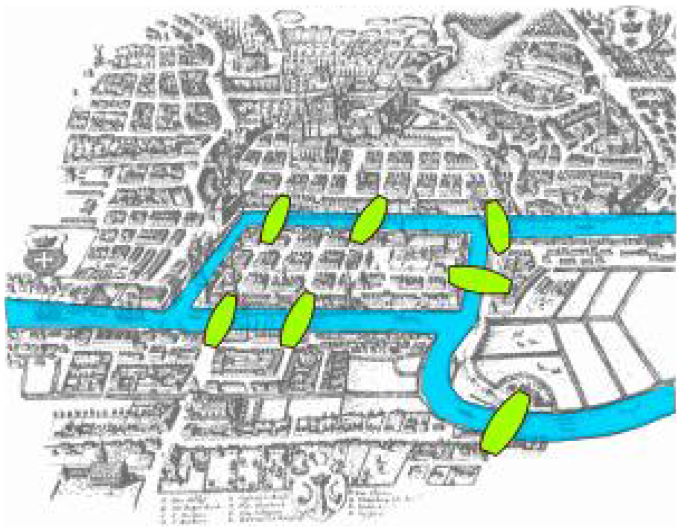
Definition 2:
We say a graph is *connected* if there is a path between every pair of its vertices. A graph is called *disconnected* otherwise.

Problem 6:
Draw a disconnected graph with four vertices.
Then, draw a graph with four vertices, all of degree one.

Problem 7:
Find a Hamiltonian cycle in the following graph.



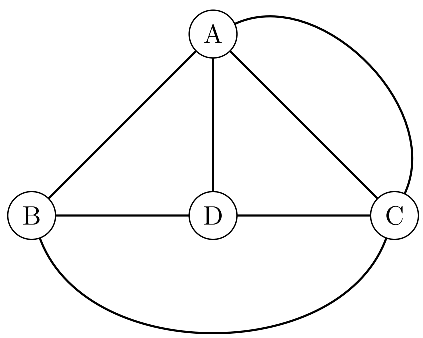
During his stay in the city of Königsberg, then the capital of Prussia, Euler came up with and solved the following problem:
Can one design a walk that crosses each of the seven bridges in Königsberg once and only once? A map of Königsberg in Euler's time is provided below.



Problem 8:
Draw a graph with the vertices corresponding to the landmasses from the picture above and with the edges corresponding to the Königsberg's seven bridges. What are the degrees of each of the graph's vertices?

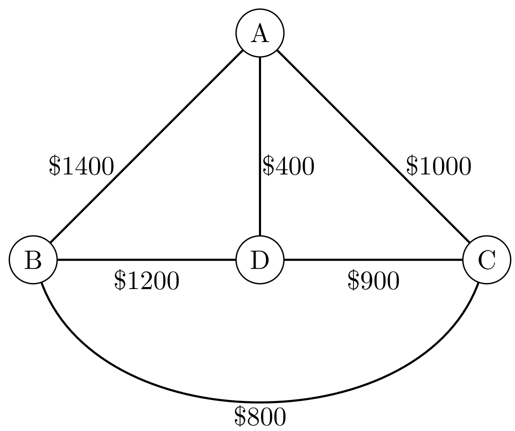
Problem 9:
Is there an Eulerian path in this map of Königsberg? Why or why not?

Problem 10:
Find a Eulerian path in the following graph.

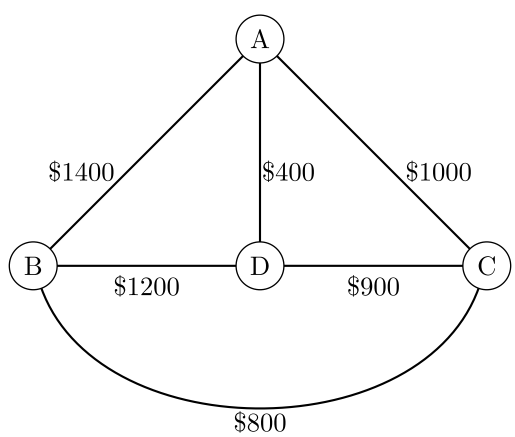


Problem 11:
Does the above graph contain a Eulerian cycle? Why or why not?

Problem 12: A Traveling Salesman
A salesman with the home office in Albuquerque has to fly to Boston, Chicago, and Denver, visiting each city once, and then to come back to the home office. The order in which he visits the cities does not matter. The airfare prices, shown on the graph below, do not depend on the direction of the travel. Find the cheapest route.



Here's an extra copy of the graph.



Problem 13:
On a test every student solved exactly 2 problems, and every problem was solved by exactly 2 students.

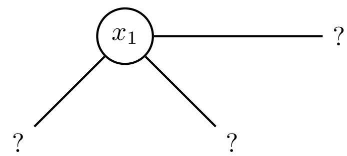
Part 1:
Show that the number of students in the class and the number of problems on the test are the same.

Part 2:
The teacher wants to make every student present one problem they solved at the board. Show that it is possible to choose the problem each student presents so that every problem on the test gets presented exactly once.

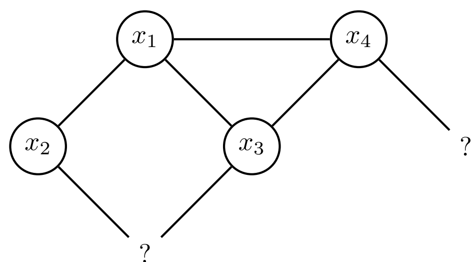
Part 3: Traversing Graphs

As you can imagine, it would be good to have computers help us with problems involving graphs. However, computers can't simply *look* at a graph and provide a solution. If we want a computer's help, we must break our problems down into a series of steps.

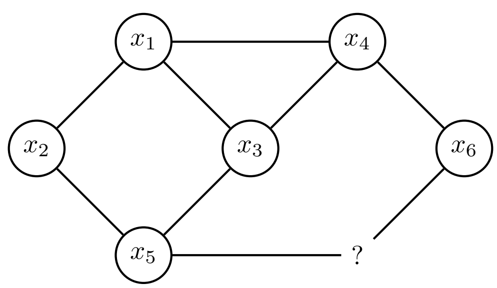
First, let's look at ways to *traverse* a graph. Say we're given a single node[†], and can only "see" the edges directly connected to it. We want to explore the whole graph. How can we do so?



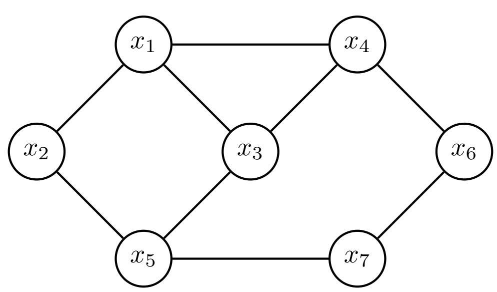
One way to go about this is an algorithm called *breadth–first search.* Starting from our first node, we'll explore the nodes directly connected to it, then the nodes connected to *those*, one at a time, and so on.
First, we explore x_2, x_3, x_4, and find that they have a few edges too:



Then we explore x_5 and x_6:



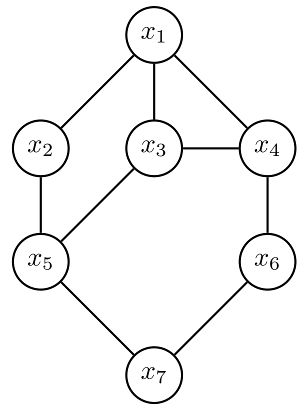
And finally, we explore x_7, and we're done.



[†]In graph theory, the terms "node" and "vertex" are equivalent.

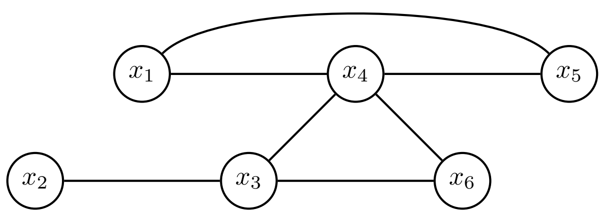
While running a breadth-first search, we can arrange our nodes in "layers." The first layer consists of our starting node, the second, of nodes directly connected to it, and so on.
For example, we get the following if we do this with the graph above:

Layer 1: x_1
Layer 2: x_2, x_3, x_4
Layer 3: x_5, x_6
Layer 4: x_7



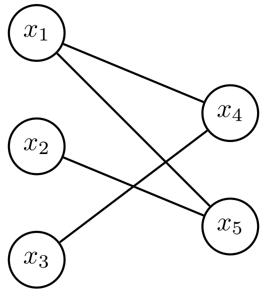
We'll call this resulting graph a *bfs graph*‡ of G.

Problem 14:
Starting from x_1, draw the bfs graph of the following:

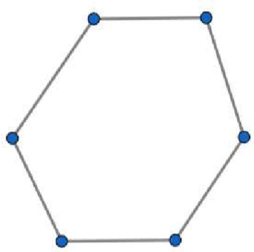
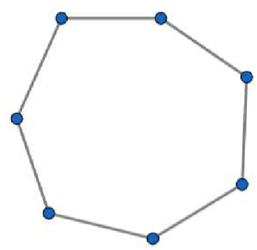
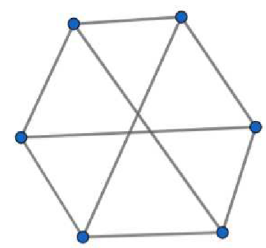
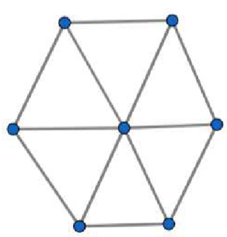
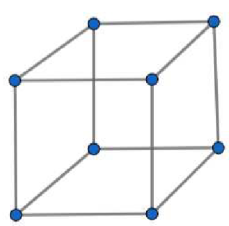
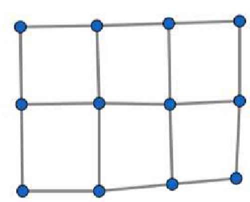


‡That is, a **b**readth-**f**irst **s**earch graph

Definition 3:
We say a graph is *bipartite* if it can be split into two groups so that no two nodes in the same group are connected. For example, the following graph is bipartite, since we can create two groups ($\{x_1, x_2, x_3\}$ and $\{x_4, x_5\}$) in which no nodes are connected.



Problem 15:
Which of the following graphs are bipartite?

Problem 16:
Show that you only need two colors to color the nodes of a bipartite graph so that no two nodes of the same color are connected.

Problem 17:
Given a large graph, how can you check if it is bipartite?