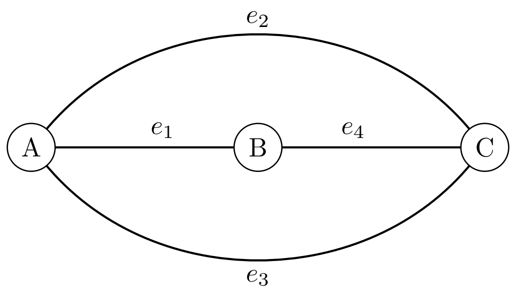
An Introduction to Graph Theory

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 $e_4 = \{B, C\}$.

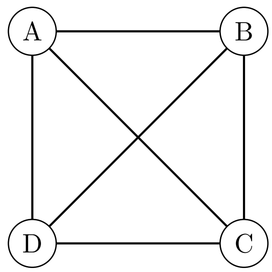
Problem 1:
Draw an undirected graph that has the vertices A, B, C, D, and E and the edges $\{A, B\}$, $\{A, C\}$, $\{A, D\}$, $\{A, E\}$, $\{B, C\}$, $\{C, D\}$, and $\{D, E\}$ in the space below.

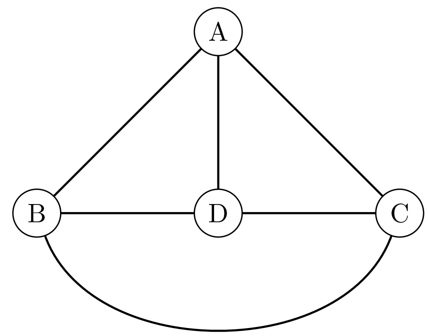
Graphs are useful for solving many different kinds of problems. Most situations that involve some kind of "relation" between elements can be represented by a graph.

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.

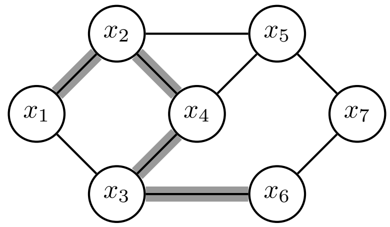
Problem 5:
One girl tells another, "There are 25 kids in my class. Isn't it funny that each of them has 5 friends in the class?" "This cannot be true," immediately replies the other girl. How did she know?

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?

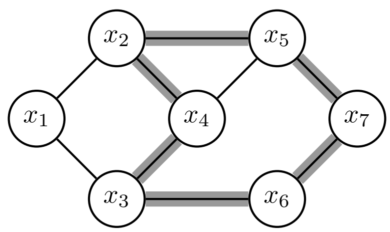
Part 2:
So how did the second girl know right away?

Part 2: Paths and cycles

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



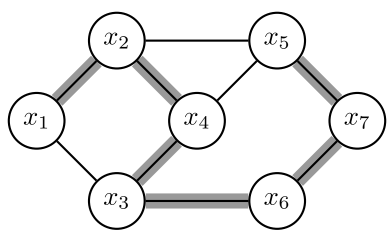
A *cycle* is a path that starts and ends on the same vertex:



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.

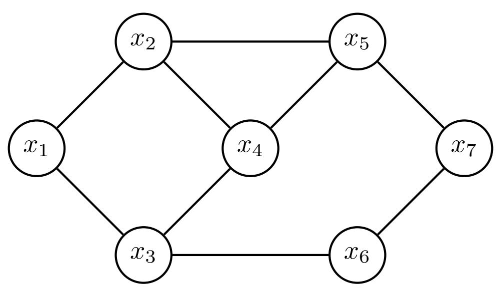


*Pronounced "oiler". These terms are named after a great Swiss mathematician, Leonhard Euler (1707-1783), considered by many as the founder of graph theory.

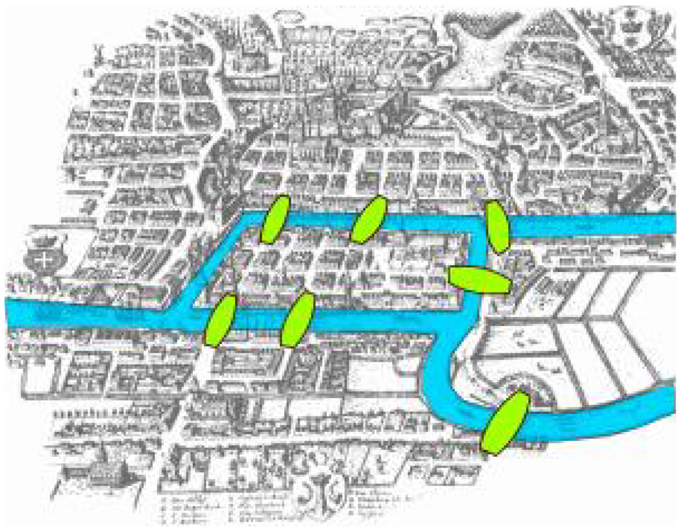
Definition 2:
We say a graph is *connected* if there is a path between every pair of its vertices. A graph is called *disconnected* otherwise.

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.



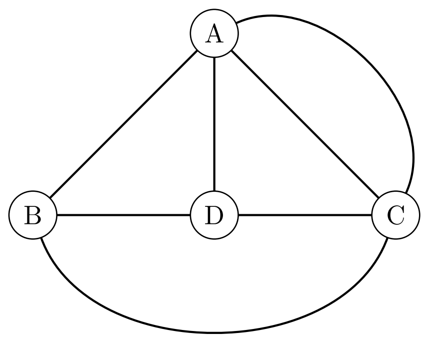
During his stay in the city of Königsberg, then the capital of Prussia, Euler came up with and solved the following problem:
Can one design a walk that crosses each of the seven bridges in Königsberg once and only once? A map of Königsberg in Euler's time is provided below.



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?

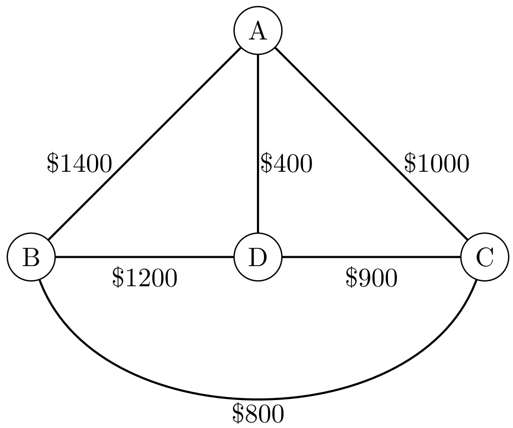
Problem 10:
Find a Eulerian path in the following graph.



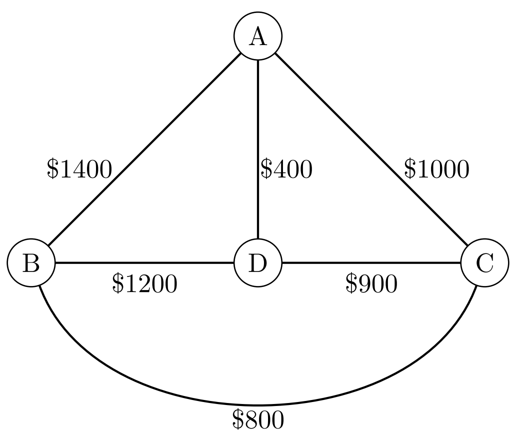
Problem 11:
Does the above graph contain a Eulerian cycle? Why or why not?

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.

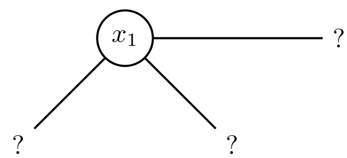
Part 1:
Show that the number of students in the class and the number of problems on the test are the same.

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.

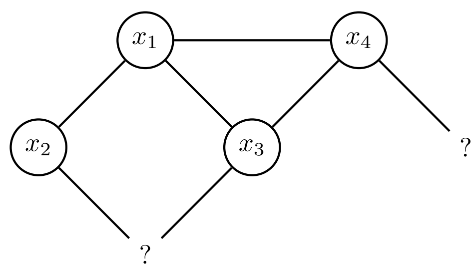
Part 3: Traversing Graphs

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.

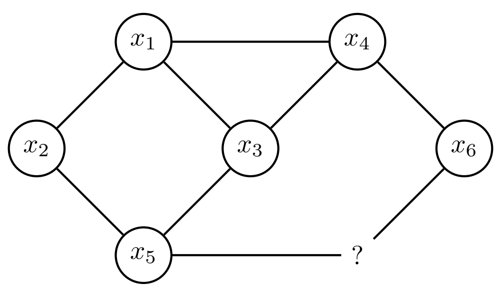
First, let's look at ways to *traverse* a graph. Say we're given a single node†, and can only "see" the edges directly connected to it. We want to explore the whole graph. How can we do so?



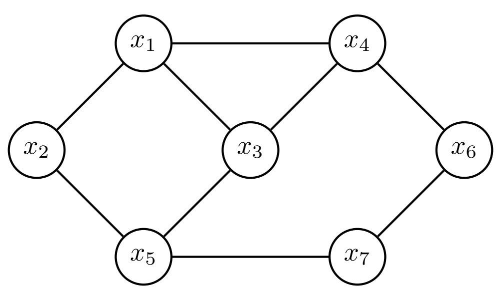
One way to go about this is an algorithm called *breadth–first search.* Starting from our first node, we'll explore the nodes directly connected to it, then the nodes connected to *those*, one at a time, and so on.
First, we explore x_2, x_3, x_4, and find that they have a few edges too:



Then we explore x_5 and x_6:



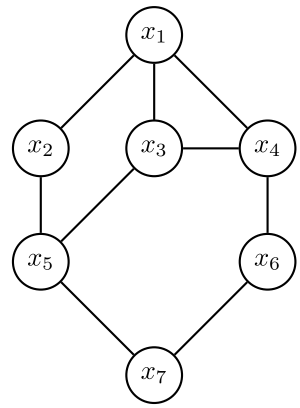
And finally, we explore x_7, and we're done.



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

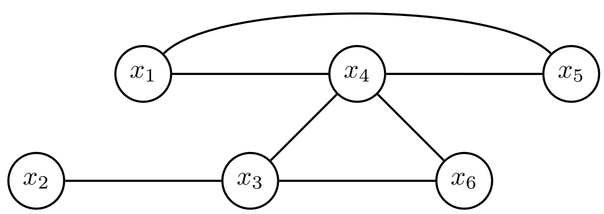
While running a breadth-first search, we can arrange our nodes in "layers." The first layer consists of our starting node, the second, of nodes directly connected to it, and so on.
For example, we get the following if we do this with the graph above:

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



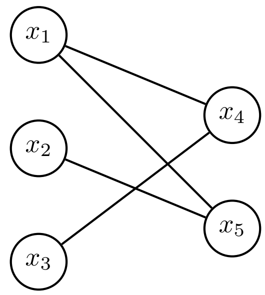
We'll call this resulting graph a *bfs graph*[‡] of G.

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

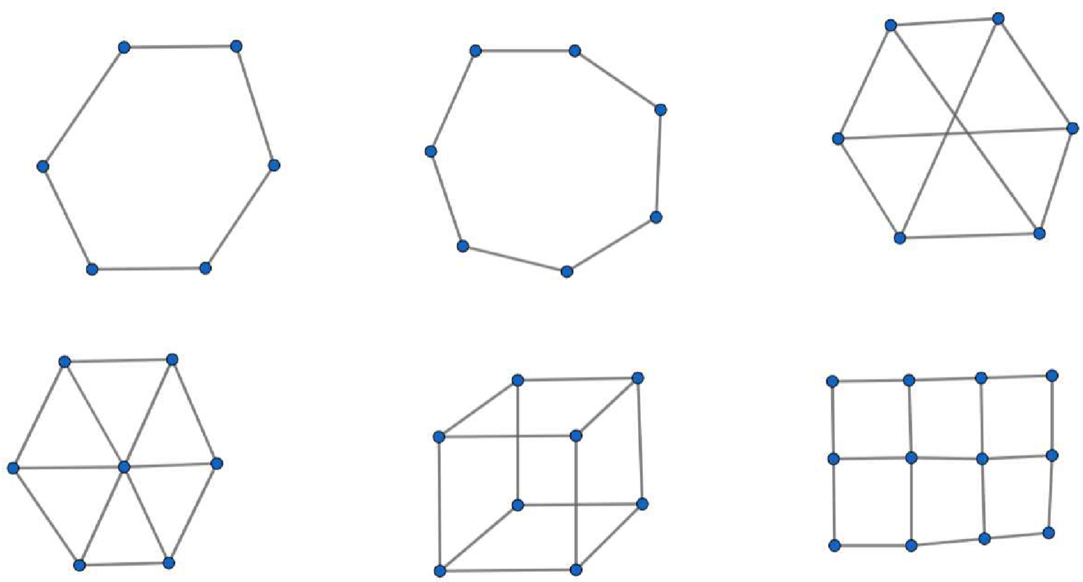


[‡]That is, a **b**readth-**f**irst **s**earch graph

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?

Graph Theory Challenge Problems

Sanjit Dandapanthula and Kason Ancelin

November 2, 2022

We're going to start with a proof of (part of) Kuratowski's theorem. First, we'll start with a definition. Two graphs G_1 and G_2 are **isomorphic** if we can rename the vertices of G_1 to get G_2. For instance, the triangle graph K_3 is isomorphic to the cycle graph C_3 with 3 vertices because they have the same fundamental structure.

Exercise 1. *Draw all non-isomorphic graphs with four vertices. Ask your instructors if you need any help with this definition.*

A **planar** graph is a graph that we can draw on the page with non-overlapping edges.

Exercise 2. *Show that K_2, K_3, and K_4 are planar. Show that cyclic graphs (graphs that are just one large cycle) are planar.*

A **complete bipartite graph** is denoted by $K_{n,m}$. It has $n+m$ vertices and is created by the following process: split the set of vertices into a group of n and m vertices respectively and draw an edge from each vertex of the first group to every vertex of the second group.

Exercise 3. *Show that $K_{2,2}$ is planar. Is it isomorphic to any of the graphs we've discussed earlier? Show that $K_{2,3}$ is planar.*

Exercise 4. *Convince yourself that $K_{3,3}$ and K_5 are not planar by trying to draw them on the page.*

Kuratowski's theorem states that these are the only cases that cause non-planarity. Namely, a graph is non-planar precisely when it contains a **subgraph** isomorphic to K_5 or $K_{3,3}$. Note that a **subgraph** of a graph G is a new graph obtained by considering only a subset of the vertices and edges of G.

Exercise 5. *If H is non-planar and H is a subgraph of G, then show that G is non-planar (hint: don't overthink it).*

Exercise 6. *We call G a **subdivision** of H if G can be obtained from H by adding vertices of degree two in the middle of the vertices of H. Show that if G is a subdivision of a non-planar graph H then G is also non-planar.*

Exercise 7. *Use the results shown in the previous two problems to show one direction of Kuratowski's theorem – namely, show that if G has a subgraph isomorphic to K_5 or $K_{3,3}$, then G is non-planar.*

The other direction of Kuratowski's theorem (if a graph is non-planar, it must contain a subgraph isomorphic to K_5 or $K_{3,3}$) is decidedly more difficult, so we will not prove this here.

Exercise 8. *The six color theorem states that any map can be colored with six or fewer colors so that no adjacent territories receive the same color. Frame this problem as a graph where each territory is a vertex and edges represent territories sharing a border. Using induction on the number of vertices in a graph,* *prove the six color theorem. Note: There exist a four and five color theorem as well, however, their proofs are more complicated.*

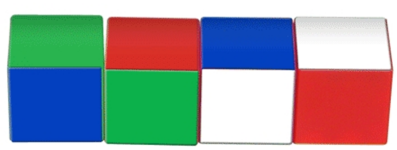
Exercise 9. *Design an algorithm to find the shortest path between any two vertices. Now suppose each edge of the graph has some weight representing maybe the difficulty of traversing the edge or time to traverse the edge. Assume you want to go from one vertex to another with the minimum weight possible for your route, where the total weight of your route is simply the sum of the weights of the edges you traverse. Design an efficient algorithm to do this. In what cases does your algorithm work well or poorly? Discuss your algorithm with an instructor and ask for their feedback on the strengths and weaknesses of your algorithm. Note: This problem won't have a "correct" answer as much as it will have "better" answers than others.*

The following is part of a texbook created by Prof. Oleg Gleizer (director of the ORMC!) containing an interesting application of graph theory to the game Insanity.

Instant Insanity

Instant Insanity is a popular puzzle, created by Franz Owen Armbruster, currently marketed by the *Winning Moves* company, and sold, among other places, on *Amazon.com*. It is advisable to have the puzzle in front of you before reading this chapter any further.

The puzzle consists of four cubes with faces colored with four colors, typically red, blue, green, and white. The objective of the puzzle is to stack the cubes in a row so that each side, front, back, upper, and lower, of the stack shows each of the four colors.



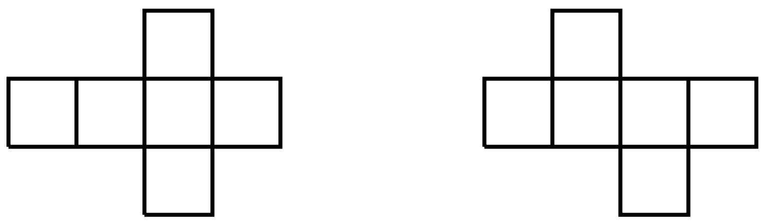
There exist 41,472 different arrangements of the cubes. Only one is a solution. Finding this one by trial and error seems about as likely as winning a lottery jackpot. However, we have witnessed a few LAMC students doing just that. Those were some truly extraordinary children!

Problem 1 *Try to solve the puzzle.*

To approach a task this formidable, the more ordinary people, like the authors of this book, need to forge some tools.

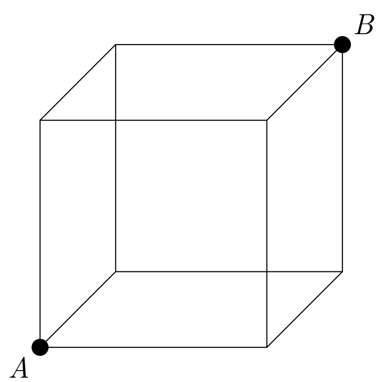
Cubic nets

A *cubic net* is a 2D picture simultaneously showing all the six sides (a.k.a. faces) of a 3D cube, please take a look at the examples below.



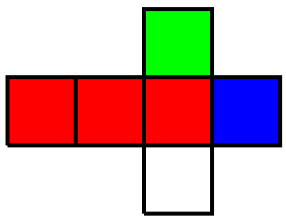
Problem 2 *Draw a cubic net different from the two above.*

Problem 3 *An ant wants to crawl from point A of a cubic room to the opposite point B, please see the picture below.*

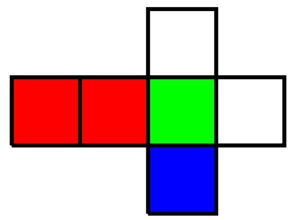


The insect can crawl on any surface, a floor, ceiling, or wall, but cannot fly through the air. Find at least two different shortest paths for the ant (there is more than one). Hint: use a cubic net.

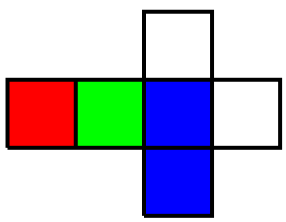
Now we have the means to take a better look at the cubes from the puzzle, the cubic nets!



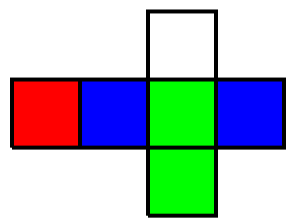
Cube 1



Cube 2



Cube 3



Cube 4

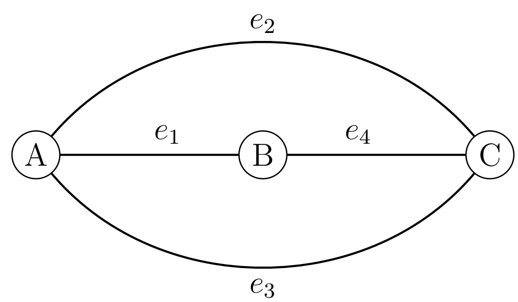
We can see that all the four cubes of the puzzle are different. Cube 1 is the only one having three red faces. Cube 2 uniquely possesses two red faces. Cube 3 is the only one having two adjacent blue faces. Cube 4 also has two blue faces, but they are opposite to each other. Finally, Cube 4 is the only one having two green faces.

Cubic nets are great for visualizing a single cube, but they are not as efficient at describing various configurations of all the four of them. We need one more tool.

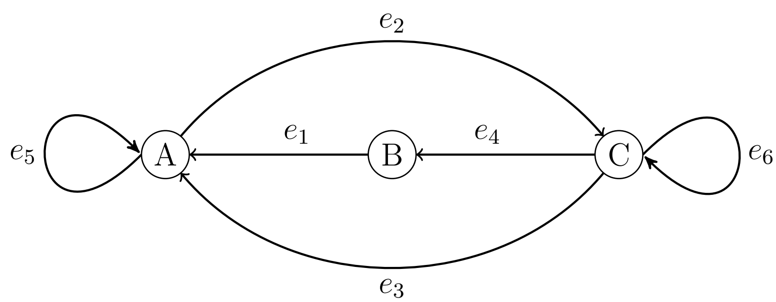
Graphs

A *graph* is a set of vertices, $\mathcal{V} = \{v_1, v_2, \ldots\}$, connected by edges, $\mathcal{E} = \{e_1, e_2, \ldots\}$. If an edge e connects the vertices v_i and v_j, then we write $e = \{v_i, v_j\}$. If the order of the vertices does not matter, the graph is called *undirected*. Typically, the word *graph* means an undirected graph. A graph is called a *directed graph*, or a *digraph*, if the order of the vertices does matter.

For example, the (undirected) graph below has three vertices, A, B, and C, and four edges, $e_1 = \{A, B\}$, $e_2 = \{A, C\}$, $e_3 = \{A, C\}$, and $e_4 = \{B, C\}$.



An edge connecting a vertex to itself is called a *loop*. For example, the digraph below has two loops, $e_5 = (A, A)$ and $e_6 = (C, C)$, in addition to the edges $e_1 = (B, A)$, $e_2 = (A, C)$, $e_3 = (C, A)$, and $e_4 = (C, B)$.



Note that we use different notations for an edge of a graph and digraph. An edge of a graph, $e = \{A, B\}$, is a set of the two vertices it connects. In this case, the order does not matter, $\{A, B\} = \{B, A\}$ as sets. An edge of a digraph, $e = (A, B)$ is a list (an ordered set) of the vertices it connects. The order does matter now, $(A, B) \neq (B, A)$.

The endpoint of a directed edge e is called its *head* and denoted $h(e)$. The starting point on an edge e is called its *tail* and denoted $t(e)$. For example, $h(e_4) = B$ and $t(e_4) = C$ for the digraph above.

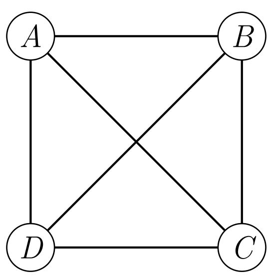
In the book, we use the letters $\mathcal{V}$ and $\mathcal{E}$ for the sets of vertices and edges in a graph. We use the letters V and E for the numbers of the vertices and edges. In other words, V is the number of elements in the set $\mathcal{V}$, E is the number of elements in the set $\mathcal{E}$.

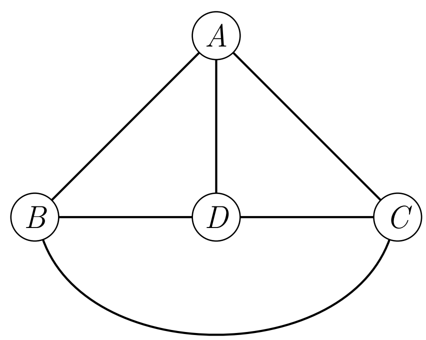
Problem 4 *Given a graph with V vertices and E edges that has no loops, how many ways are there to orient the edges so that the resulting digraphs are all different?*

Problem 5 *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 $\{D, E\}$.*

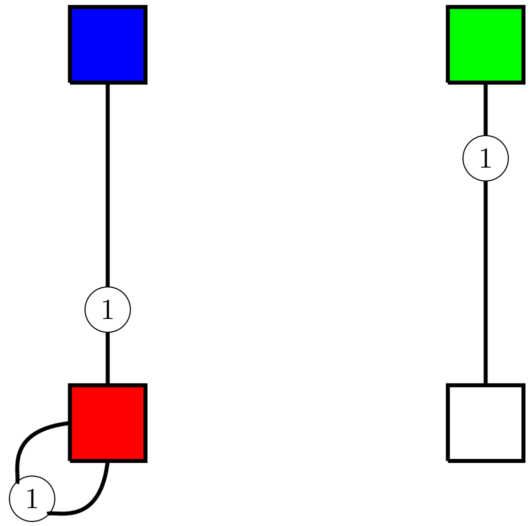
Two different pictures of a graph can look very dissimilar.

Problem 6 *Prove that the two pictures below represent the same graph by comparing the sets of their vertices and edges.*

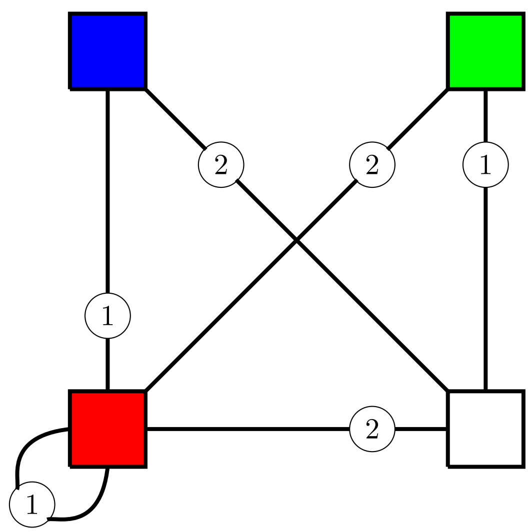




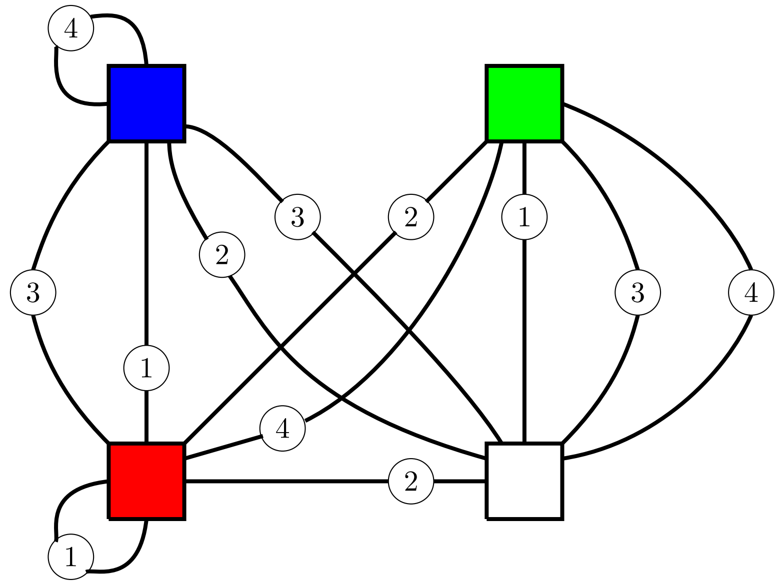
Getting back to the puzzle, let us represent Cube 1, see page 4, by a graph. The vertices will be the face colors, Blue, Green, Red, and White, $\mathcal{V} = \{B, G, R, W\}$. Two vertices will be connected by an edge if and only if the corresponding faces are opposing each other on the cube. Cube 1 has the following edges, $e_1 = \{B, R\}$, $e_2 = \{G, W\}$, and the loop $e_3 = \{R, R\}$. To emphasize that all the three edges represent the first cube, let us mark them with the number 1.



Cube 2 has the following pairs of opposing faces, $\{B, W\}$, $\{G, R\}$, and $\{R, W\}$. Let us add them to the graph as the edges e_4, e_5, and e_6.



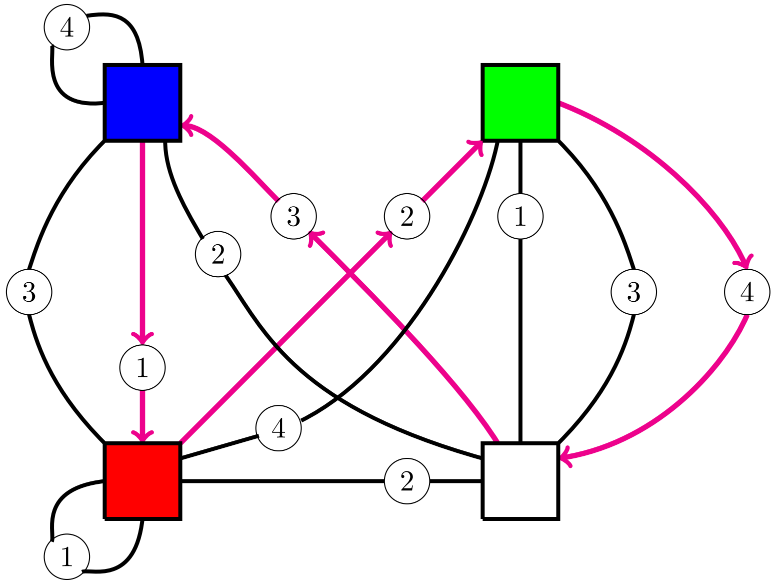
Let us now make the graph represent all the four cubes.



Problem 7 *Check if the above representation is correct for Cubes 3 and 4.*

With the help of the above graph, solving the puzzle becomes as easy as a walk in the park, literally. Imagine that the vertices of the above graph are the clearings and the edges are the paths. An edge marked by the number i represents two opposing faces of the i-th cube. Let us try to find a closed walk, a.k.a. a cycle, in the graph that visits each clearing once and uses the paths marked by the different numbers, $i = 1, 2, 3, 4$. If we order the front and rear sides of the cubes accordingly, then the front and rear of the stack will show all the four different colors in the order prescribed by our walk.

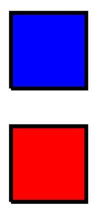
For example, here is such an (oriented) cycle, represented by the magenta arrows on the picture below.



The first leg of the walk tells us to take Cube 1 and to make sure that its blue side is facing forward. Then the red side, opposite to the blue one, will face the rear.

Front:

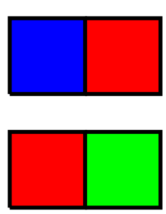
Rear:



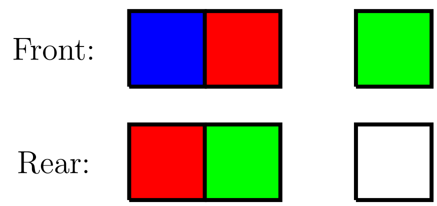
The next leg of the walk tells us to take Cube 2 and to place it in such a way that its red side faces us while the opposing green side faces the rear. Since we go in a cycle that visits all the colors one-by-one, neither color repeats the ones already used on their sides of the stack.

Front:

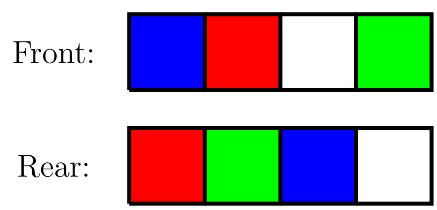
Rear:



The third leg of the walk tells us to take Cube 4, not Cube 3, and to place it green side forward, white side facing the rear.



Finally, the last leg of the walk tells us to take Cube 3 and to place it the white side facing forward, the opposite blue side facing the rear.



Now the front and rear of the stack are done. If we manage to find a second oriented cycle in the original graph that has all the properties of the first cycle, but uses none of its edges, we would be able to do the upper and lower sides of the stack and to complete the puzzle. Using the edges we have already traversed during our first walk will mess up the front-rear configuration, but there are still a plenty of the edges left!

Problem 8 *Complete the puzzle.*