Announcements

- Final exam review tomorrow

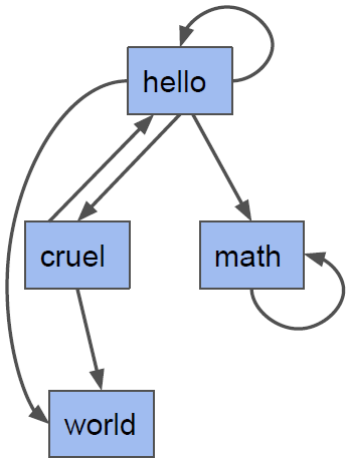
Walks, paths and Eulerian tours

Definition: Let $G = (V, E)$ be a graph (directed or undirected). A **walk** of graph G is a sequence of alternating vertices (in V) and edges (in E) such that

- We start on any vertex and end on any vertex
- A single step in the walk proceeds upon an outgoing edge from the current vertex

Note: A walk has to respect edge direction in a directed graph. In an undirected graph, it doesn't matter.

Example: Consider the directed graph below



Valid walks:

- hello→hello→cruel→hello
- hello
- hello→math→math
- cruel→hello→cruel→world

Invalid walks:

- hello→math→hello
- hello→math→world

Definition: Let $G = (V, E)$ be a graph (directed or undirected). A **path** of a graph is a walk in which no vertex is repeated. The **length** of a path is the number of edges that it traverses.

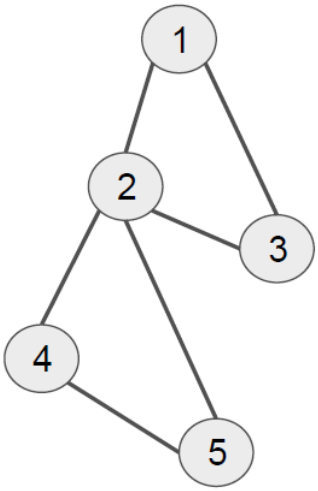
Example: hello→cruel→world is a path of length 2

Definition: Let $G = (V, E)$ be a graph (directed or undirected). An **Eulerian tour** of a graph is a special kind of walk which starts and ends at the same vertex, and traverses each edge exactly once.

Summary of terminology:

- A **walk** is pretty much any traversal, going from vertex to vertex
- A **path** does not repeat vertices
- An **Eulerian tour** is a walk that does not repeat edges and starts and ends at the same vertex

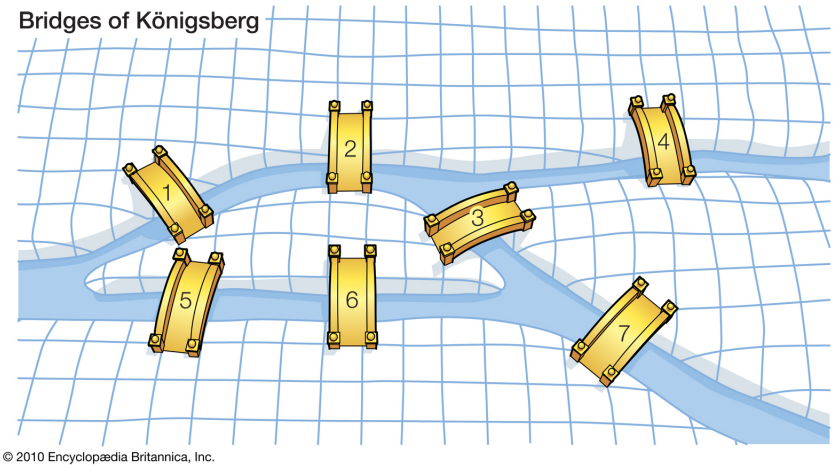
Example: Can there be an Eulerian tour in this graph?



Answer: Yes! (we can repeat vertices, just not edges)

$$2 \to 1 \to 3 \to 2 \to 4 \to 5 \to 2$$

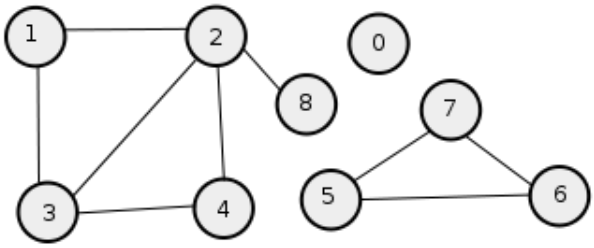
Puzzle: The city of Konigsberg has two islands formed by a river with seven bridges connecting the islands and the mainland. Is there a path that traverses each bridge exactly once?



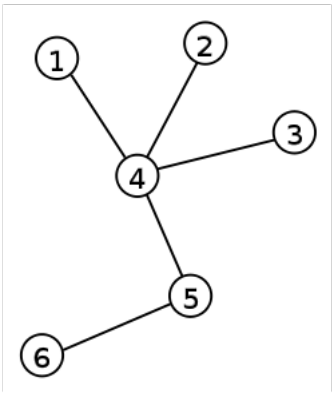
We can represent this as a graph, and solve the riddle by checking if it has an Eulerian tour. First, we need a bit more terminology.

Definition: An undirected graph G is **connected** if there is a path between every pair of distinct vertices in the graph. An undirected graph is called **disconnected** if it is not connected.

Example: The following graph is disconnected



And the following graph is connected



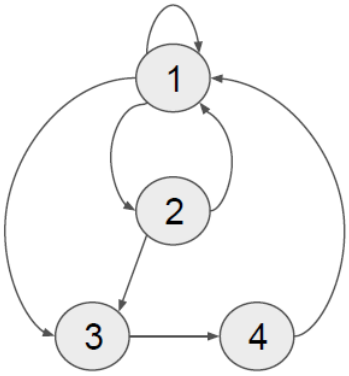
Definition: A subgraph of a disconnected graph G is called a **connected component** of G if it is a "maximally connected" subgraph of G

Example: In the disconnected graph above, $\{1, 2, 3, 4, 8\}$, $\{0\}$, and $\{5, 6, 7\}$ are connected components, but $\{5, 6\}$ is not because it is not maximal.

Connectedness in *directed graphs* is a bit trickier. We define two notions of connectedness:

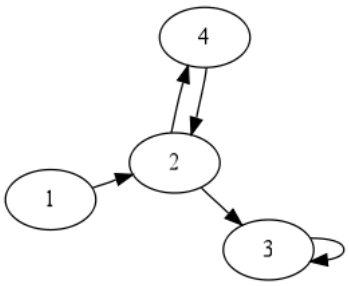
Definition: A digraph G is **strongly connected** if for each pair of distinct vertices a and b there is a path from a to b *and* a path from b to a

Example:



Definition: A digraph G is **weakly connected** if for each pair of distinct vertices a and b there is a path from a to b *or* a path from b to a

Example:

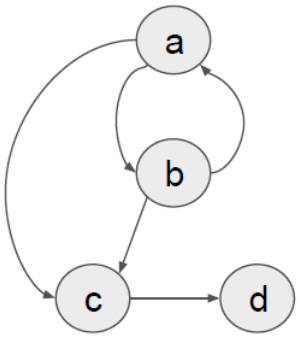


Definition: A subgraph of a digraph G is a **strongly connected component** if it is a maximal strongly connected subgraph of G.

Example: What is the strongly connected component of the weakly connected graph above?

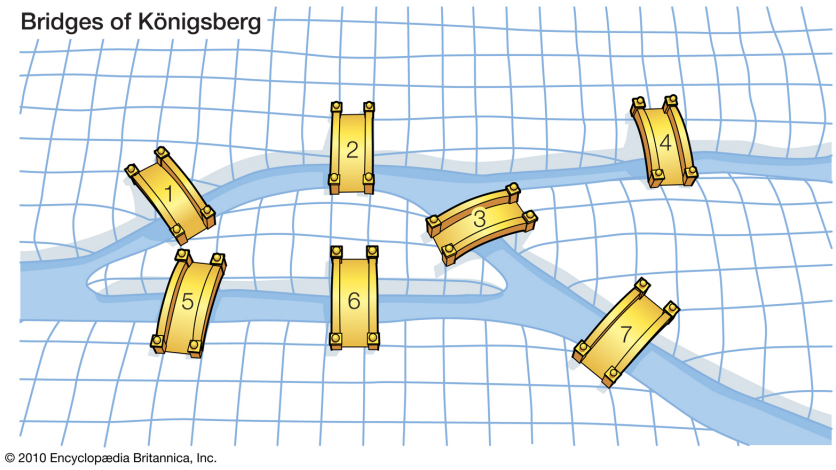
Answer: $\{2, 4\}$

Example: What is the smallest number of directed edges you could add to the following graph to make it a strongly connected graph?

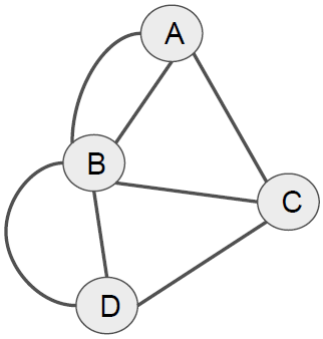


Answer: 1 – make a connection from d to b or from d to a.

Seven bridges of Koenigsberg:

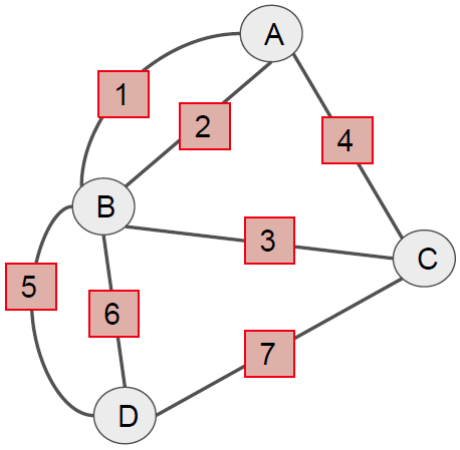


Let's formalize this as a graph. Let the bridges be edges and the land masses be vertices, then we get something like the following.



This is an example of a **multigraph** because pairs of vertices have multiple edges connecting them. We can turn it into a simple graph by making each edge distinct

- Turn each bridge into a vertex
- This amounts to giving them labels, and we want to visit each bridge vertex.



Now, does *this* graph have an Eulerian tour?

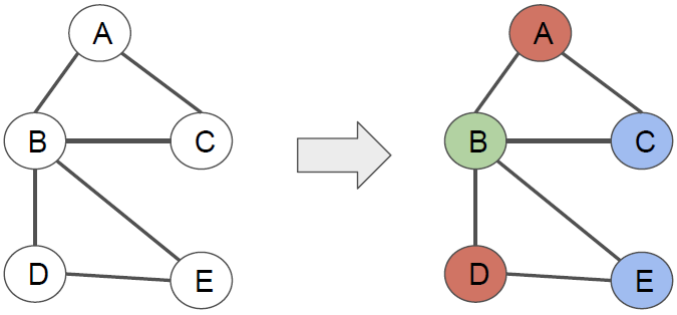
Answer: Nope! Here's why:

- We either start on C or somewhere else.
- Knowing that C has three edges, if we start on C, there will be no way to return to C.
- Similarly, if we start somewhere else, we will arrive at C via one of its edges
- Now there are two more edges to traverse, which means we will leave C then come back to C and be stuck there which means we won't be able to return to where we began.

Theorem: A connected undirected graph G (with no self-loops) has an Eulerian tour *if and only if* each vertex has even degree

Graph coloring

Graph coloring is the task of assigning labels (colors) to vertices so that no vertices of the same color share an edge.



Definition: Let $G = (V, E)$ be a graph (directed or undirected). A **cycle** of a graph is a sequence of alternating vertices and edges that starts and ends on the same vertex but doesn't repeat any other vertices. The number of edges traversed is called the *length* of the cycle.

Examples:

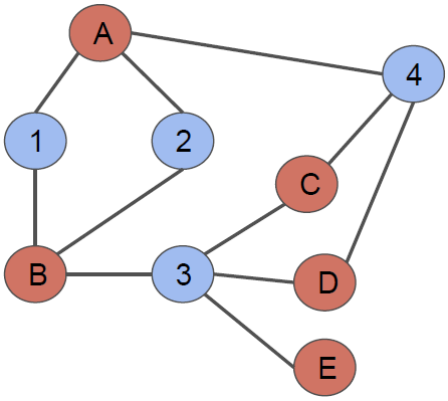
$A \to B \to C \to A$

$B \to E \to D \to B$

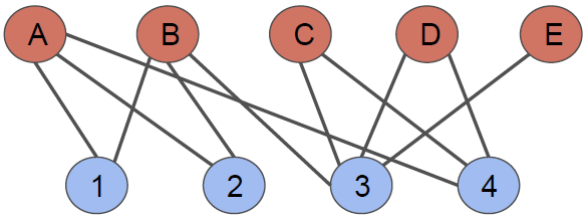
Definition: A graph $G = (V, E)$ is called **bipartite** *if and only if* the vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that

1. $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \varnothing$
2. Any edge in E goes from a vertex in V_1 to one in V_2, and vice versa

Example: The following graph is bipartite, where V_1 are the blue/numerical vertices and V_2 are the red/letter vertices.



Rearranging it, we can see this much more easily



But it's difficult to decide whether a graph is bipartite or not just based on the possibly complicated picture.

Theorem: A graph G is bipartite *if and only if* it has no odd length cycles

Proof. (One direction, by contradiction) Suppose G is bipartite and has an odd length cycle,

$$v_1 \to v_2 \to v_3 \to \cdots \to v_{2n} \to v_{2n+1} \to v_1$$

Since G is bipartite, we must be able to color the vertices along this cycle alternating blue/red. Without loss of generality, we can start with blue:

$$v_1 \to v_2 \to v_3 \to \cdots \to v_{2n} \to v_{2n+1} \to v_1 \quad \Rightarrow\Leftarrow$$

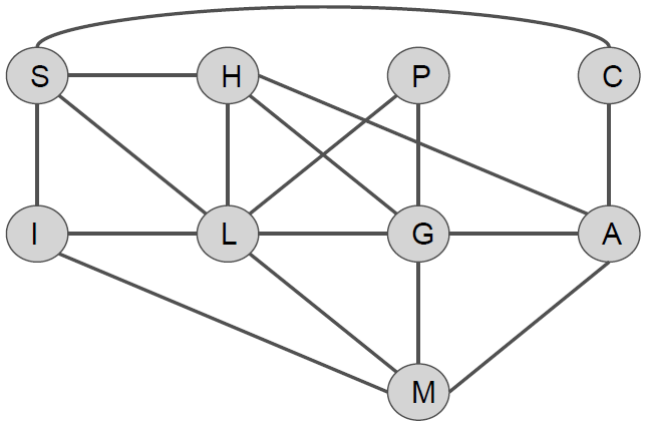
v_1 can't be both red and blue, so if G is bipartite, then it must not have any odd length cycles. □

Example: Suppose you are responsible for scheduling final exams at a university. You want to make sure that any two courses with a common student are scheduled at different times to avoid a conflict.

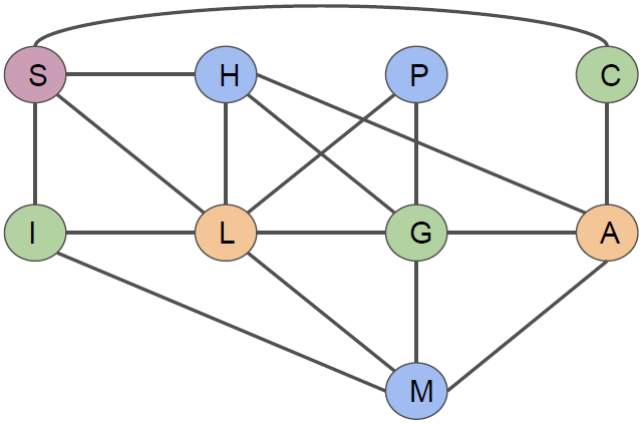
We can encode the information in a graph by assigning each course a vertex, and connecting two vertices by an edge if they have a common student. The adjacency matrix below has an X if there is an edge between two vertices.

Class	A	C	G	H	I	L	M	P	S
Astronomy		X	X	X			X		
Comp Sci	X								X
Greek	X			X		X	X	X	
History	X		X			X			X
Italian						X	X		X
Latin			X	X	X		X	X	X
Math	X		X		X	X			
Physics			X			X			
Spanish		X		X	X	X			

Below is the visual graph.



Now think about finding a coloring of the graph such that no adjacent vertices have the same color. Then associating a final exam time with each color solves the problem. Here is one such coloring of the graph, which requires 4 different colors.



This raises some questions:

- Can you *always* find a coloring of a graph?
- How many colors do you need?
- Are there certain colorings that are "better" than others?

and some answers

- Yes! A graph with n vertices can definitely be colored using n colors
- The scheduling problem motivates us to use a few colors as possible

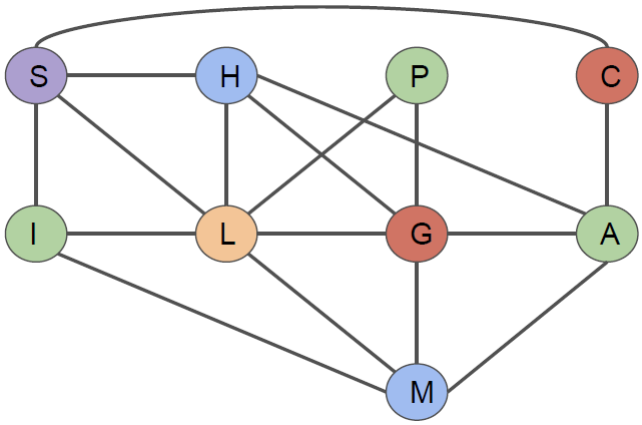
Definition: The **chromatic number** of a graph, $\chi(G)$, is the minimum number of colors required to color a graph.

How can we find a graph coloring?

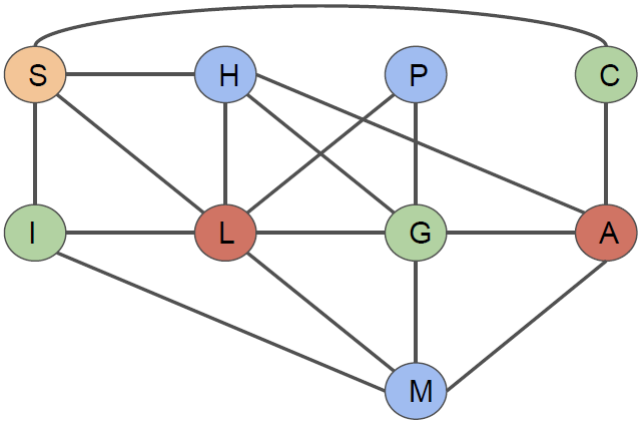
Here is a **greedy algorithm** that is *not* guaranteed to find the minimum coloring of a graph, but *is* guaranteed to find a coloring that works

1. Order the vertices in some arbitrary way
2. Color the first vertex using Color 1
3. Pick the next uncolored vertex v
    a. Color it using the lowest-numbered color that has not been used on any of the vertices adjacent to v
    b. If you run out of colors, then add a new color
4. Repeat step 3 until all vertices are colored

Example: Let's test it out on the exam scheduling problem. Order the vertices as $A, C, G, H, I, L, M, P, S$ and use colors green (1), red (2), blue (3), orange (4), and purple (5)



We used all 5 colors, but we know we can do better! Let's try again with a different vertex ordering. This time we'll use: $G, L, H, P, M, A, I, S, C$



This time we only needed 4 colors. This happens frequently with greedy algorithms.

In practice:

- Run the algorithm a few times
- Use different vertex orderings each time
- Choose the one that gives the minimal coloring

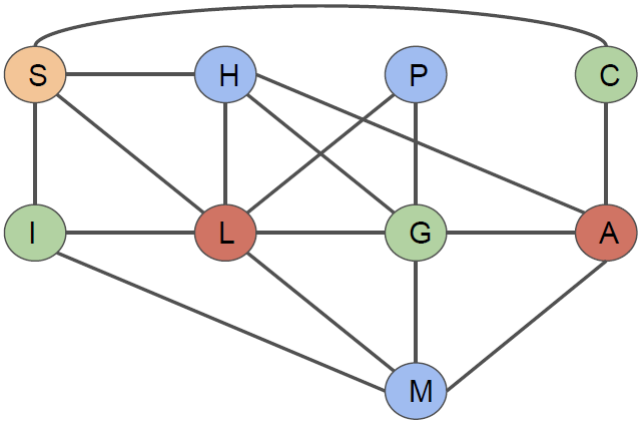
But it would be nice if there were some way we could tell how well we can possibly do. It turns out, we can't easily say what the minimal coloring is. But we *can* say what the worst case coloring is.

Greedy coloring theorem: If d is the largest degree of any vertex in graph G, then G has a coloring with $d+1$ or fewer colors. In other words, the chromatic number of G is at most $d+1$

Example: In the exam scheduling problem, the largest degree is 6 (L), so we need no more than 7 colors.

Definition: A graph is a **planar** graph if it can be drawn in the plane without any edges crossing.

The 4-color theorem: Any **planar** graph can be colored using at most 4 colors ($\chi(G) \leq 4$)



Planar representation: