Math 3322: Graph Theory[1] Mikhail Lavrov

Lecture 1: What are graphs?

August 13, 2024 *Kennesaw State University*

1 Examples of graphs

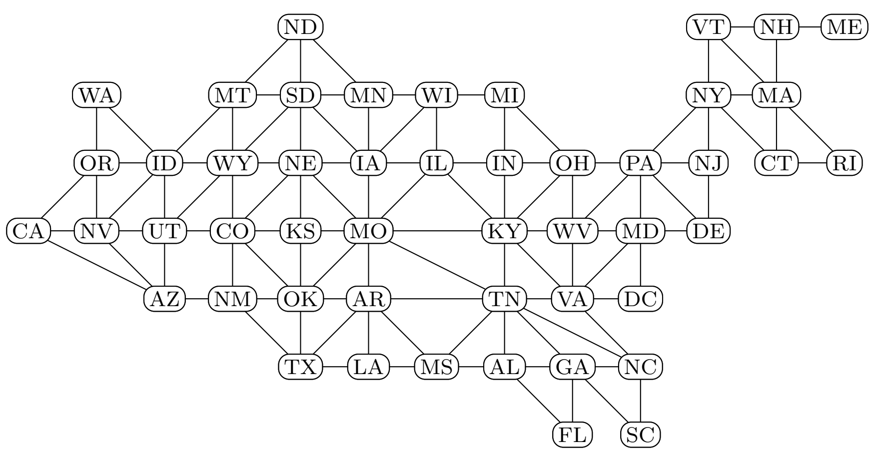
1.1 Tour of the US

Suppose that you decide to take a tour of the 48 contiguous US states, by car. To make things extra challenging for yourself, you add a condition: you cannot visit any state more than once. We can ask several questions:

- Can you visit all 48 states?
- Can you visit all 48 states, starting from Georgia?
- Can you visit all 48 states, and end adjacent to where you started?

We will not answer any of these questions today (though you can think about them on your own, if you like). Instead, we will talk about graphs: the setting where we can ask this kind of question.

If you want to solve this problem, you do not need all the information on a map of the US. It might be equally convenient (or even more convenient) to represent the US by the following diagram:[2]



(Here, a line is drawn between any two states that are directly connected by at least one drivable road.)

[1]This document comes from an archive of the Math 3322 course webpage: http://misha.fish/archive/3322-fall-2024

[2]Weisstein, Eric W. "Contiguous USA Graph." From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/ContiguousUSAGraph.html

Most of the features of the diagram are irrelevant: it does not actualy matter that Maine is drawn in the top right and California is the leftmost state, and it does not mean anything that the line between Tennessee and Montana is longer than most other lines. We could encode all the information we *need* to solve the problem as follows:

1. Make a list of all the states we want to visit: AL, AZ, AR, ..., WY.
2. Make a list of all pairs of states with a road between them, such as $\{\mathrm{AL}, \mathrm{FL}\}$ or $\{\mathrm{MO}, \mathrm{TN}\}$.

This is exactly what a graph is!

Formally, a **graph** G is a pair (V, E) where

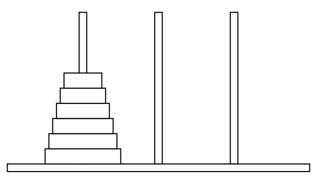
- V is a set of objects called **vertices**. (These can be anything.)
- E is a set of **edges**; each edge is a pair $\{v, w\}$ of vertices $v, w \in V$ and tells us that v and w are **adjacent**.[3]

We will often denote the edge between v and w as vw instead of $\{v, w\}$. This is unordered: vw and wv are the same edge.

It is often convenient to represent a graph by a diagram like the one on the previous page, where the edges are drawn as lines connecting the vertices. Sometimes, when we don't need to keep track of the names of the vertices, we'll draw the vertices as simple dots.

1.2 The Towers of Hanoi puzzle

In the Towers of Hanoi puzzle, you have three pegs, and some number of disks of different sizes stacked on the pegs. Initially, all the disks are placed on one peg, sorted by size (with the smallest disk on top):



You are allowed to make the following moves in this puzzle: lift the top disk on a peg, and put it down on another peg. However, you cannot place a larger disk on top of a smaller one.

The goal of the puzzle is to move all the disks from one peg to another.

To model this puzzle as a graph $G = (V, E)$, we can do the following:

- Let V be the set of all possible states of the puzzle.
- Put an edge between two vertices (two states of the puzzle) if it's possible to get from one to the other by a single move. (This is a symmetric relation: if we can move a disk from one peg to another, we can always move it back.)

[3] Many people informally say that the edge "connects v to w", or that v and w are "connected". I will avoid this terminology, and I encourage you to avoid it, because it is easily confused with the notion of "connected graphs" and "connected components" which we will learn about in tomorrow's lecture.

As a problem, the Towers of Hanoi puzzle looked very different from our driving-around-the-US question. However, as graphs, we are asking some similar questions: in both cases, we are "moving around" the vertices of the graph, traveling along the edges.

1.3 Circuit board layouts

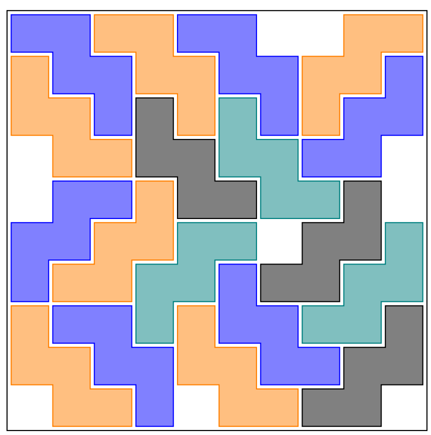
A circuit board is a plastic sheet with many components on it, some of which are connected by conductive copper tracks. When we are initially designing a circuit, we keep track of what the components are, and which ones should be connected by wires. This is a graph: the components are the vertices, and the wires are the edges.

When we print the circuit board, we have an additional constraint: we don't want the conductor tracks to cross!

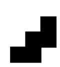
A graph does not keep track of layout: it is the same no matter how you represent it as a diagram. However, graph theory does study the question: does a graph *have* a diagram in which no two edges cross? This is the problem of finding a **plane embedding** of a graph.

1.4 Placing zigzag tiles

You have an infinite supply of  shaped tiles. How many of them can you place on a 10×10 grid without overlap? One very good solution (which I suspect to be optimal) is given below:



How did I find this solution? I did it by encoding the problem as a graph and using some tools in Mathematica's graph theory library. It's not obvious how to represent this as a graph theory problem, but here is what I ended up doing:

- Let the vertices be all ways to place a *single*  tile on the 10×10 grid.
- Put an edge between two vertices if the tile placements they represent are incompatible: the tiles would overlap.

A solution to the puzzle is a set of vertices. Edges represent conflicts, so a conflict-free solution is a set of vertices with no edges between them. This is called an **independent set** in a graph, and we will look at the problem of finding these later on in the class.

There are 256 vertices: the center of the tile can go anywhere in an 8×8 square of the grid, because it cannot lie along an edge, and there are 4 ways to rotate the tile, once you know where its center goes. This is still a lot for a human, but not out of the reach of computer algorithms.

There is a second graph-theoretical problem here (which I solved by trial and error). The tiles in my illustration of the solution are colored orange, blue, gray, and teal so that tiles that share a border are given different colors. Is this possible? And could we reduce the number of colors needed?

For this question, a different graph is necessary. The vertices of this graph are going to be the tiles we *actually* placed in the grid. We put an edge between two vertices if the two tiles share a border (and should not be the same color). To color the illustration, we should define a function f from the vertex set V to the set $\{\text{orange}, \text{blue}, \text{gray}, \text{teal}\}$ so that for all edges vw, $f(v) \neq f(w)$.

This function is called a **coloring** of the graph even in applications that are not about colors. (The origin of the problem precedes graph theory as a field: it was posed in the 19$^{\text{th}}$ century by mapmakers.) The least number of colors necessary to color a graph is called the **chromatic number** of a graph.

2 Walks and connectedness

Let's end by getting started introducing some of the concepts we'll use to solve our first few problems. To do this, I want to introduce one final problem we can model with graphs: the three cups puzzle. It is similar to the Towers of Hanoi puzzle in how we will model it, but it is much simpler.

In the three cups puzzle, you have three cups lined up in a row. In one move, you are allowed to take two consecutive cups (the first and second cup, or the second and third) and flip both of them over. If a cup was already upside down, flipping it will make it right side up again.

The goal of the puzzle is to flip all three cups upside down. But after drawing the diagram of all 8 states, and the adjacencies between them, we quickly see that there's no solution:

$$\begin{tikzcd}
\text{UUU} \arrow[r, dash] \arrow[d, dash] & \text{DDU} \arrow[d, dash] & \text{DDD} \arrow[r, dash] \arrow[d, dash] & \text{UUD} \arrow[d, dash] \\
\text{UDD} \arrow[r, dash] & \text{DUD} & \text{DUU} \arrow[r, dash] & \text{UDU}
\end{tikzcd}$$

(I am using U and D to denote a cup facing Up and a cup facing Down, respectively.)

There are no edges leading from the left part of the diagram to the right part. UUU is on the left, and DDD is on the right, and so we cannot get from UUU to DDD.

We see the same thing show up in other problems. For example, suppose we return to the car-driving problem, but this time we want to take a world tour by car. We run into an obstacle: cars cannot drive on water. So our graph will, once again, have several pieces corresponding to different landmasses. We can get from one vertex (location) to another if they are on the same landmass, but we have no hope if they are on different landmasses.

Our goal for the next lecture is to formalize this idea, which will give us the notions of **connected graphs** and **connected components** of graphs. For now, we want to begin by defining formally what it means to "get from one vertex to another".

We say that a $v - w$ **walk** in a graph G is a sequence of vertices $v_0, v_1, v_2, \ldots, v_\ell$ where $v_0 = v$, $v_\ell = w$, and for $i = 0, 1, \ldots, \ell - 1$, $v_i v_{i+1}$ is an edge. For example, in the cups puzzle, the sequence UUU, DDU, DUD is a UUU – DUD walk.

The number ℓ is the **length** of the walk. (The length is the number of edges used; one fewer than the number of vertices.) The walk UUU, DDU, DUD has length 2, which matches our intuition that it takes 2 steps to get from UUU to DUD.

A special kind of walk is a $v - w$ **path**. A path is a walk in which no vertices are repeated. The above walk is also a path, but UUU, DDU, DUD, UDD, DUD is a UUU – DUD walk which is not a path: the vertex DUD occurs twice.

Initially, we will be more interested in walks, rather than paths. Looking at paths will be interesting to us later on, when we want to say things like "there are two ways to get from UUU to DUD". Indeed, there are only two UUU – DUD paths in the three cup graph; however, there are infinitely many UUU – DUD walks, because we can take any number of redundant steps.

3 Practice problems

These are not homework problems; they are only here so that you can get extra practice with the material.

1. In a very silly version of the cup puzzle, you have three cups in a row, all upside down. In one step, you can flip any individual cup (from upside down to right side up, or back).

   (a) Let G be the graph whose vertices are possible states of this puzzle, with an edge between states that are one step apart. Draw a diagram of G.

   (b) What is the **order** of G: the number of vertices? What is the **size** of G: the number of edges?

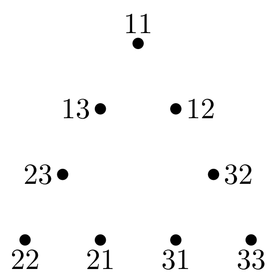
   (c) Find a UUU − DDD walk of any length.

   (d) Find a UUU − UUU walk which visits every vertex.

2. In this problem, we'll explore the graph for the Towers of Hanoi puzzle.

   We can describe a state of the game by going through the disks, from largest to smallest; for each disk, write down 1, 2, or 3 to record whether it's on the 1^{st}, 2^{nd}, or 3^{rd} peg. For example, the initial state earlier in the notes is "11111". If we move the top disk to the last peg, it will become "11113".

   (a) Complete the diagram of the $n = 2$ Towers of Hanoi graph given below by drawing the edges corresponding to possible moves:

   (b) Find a $11-22$ walk that is as short as possible. Then, find one that is as long as possible, but without visiting the same vertex more than once.

   (c) How many vertices are in the n-disk puzzle? Trickier question: how many edges are there? (Check your answer by applying it to the $n = 2$ case, for which you have the diagram.)

3. The zigzag tile graph from today's lecture is too large for you to draw by hand, so let's look at a much simpler problem of the same type.

   Suppose we are trying to place 2×2 square tiles on a 4×4 grid without overlap.

   (a) Draw a diagram of the graph where the vertices are ways to place a single 2×2 tile on the grid (there should be 9 vertices), with an edge between two vertices when the tile placements they represent are incompatible.

(b) Find an independent set in the graph representing the "boring" solution, which places four 2×2 square tiles covering the entire grid.

(c) Find some other interesting non-overlapping tile placement in the grid, and find the independent set in your graph that corresponds to it.

4. Suppose that you have a circuit in which any two components must have a wire between them. How many components can you have before it becomes impossible to put the components and the wires on a circuit board without the wires crossing?

5. An **interval graph** is a graph whose vertices are intervals of the real line, with an edge between two vertices when the two intervals overlap.

   (a) Draw a diagram of the interval graph with vertices $[8, 13]$, $[9, 11]$, $[10, 14]$, $[12, 15]$, $[16, 18]$, $[17, 19]$.

   (b) Suppose that the intervals represent times of day that certain events are happening (for example, TV shows you want to watch). What would an independent set in this graph correspond to?

6. Consider the map of the contiguous USA graph. Remember that we were trying to drive around the US without visiting any graph more than once. In other words, we are looking at paths in this graph, not walks.

   (a) Suppose you decide to take a tour of the 48 states, starting from Florida (FL). You do not have to worry about whether this is possible; but *if* it is, where will your tour end?

   (b) Suppose you decide to take a tour of the 48 states, starting from Georgia (GA). In fact, this is impossible; you will not be able to visit all three of Florida (FL), South Carolina (SC), and the endpoint state from your previous part. Why not?

   (c) Suppose you want something simpler: starting from Georgia (GA), you want to get to Vermont (VT) and then return to Georgia. You still do not want to visit any state other than Georgia more than once. Why is this also impossible?

7. Find a way to color this tiling using only 3 colors so that no two tiles that share a border have the same color.