CMPSCI 575/MATH 513

Combinatorics and Graph Theory

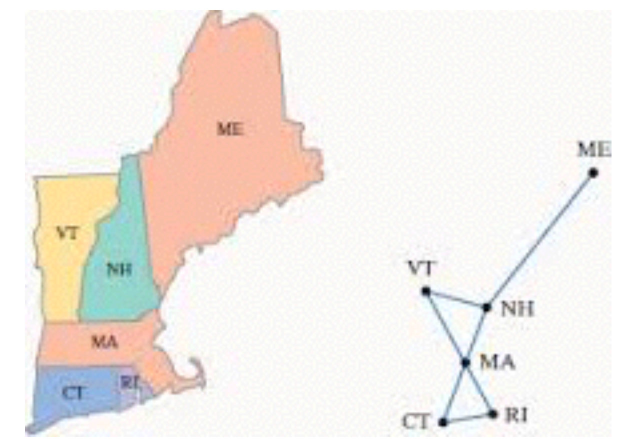
Lecture #6: Graph Coloring
(Tucker Section 2.3)
David Mix Barrington
19 September 2016

Graph Coloring

- Colorings and Chromatic Number
- Basic Examples
- Coloring a Wheel
- NP-Completeness Overview
- The 3-Colorability Problem
- Application to Garbage Trucks
- Scheduling Round-Robin Tournaments

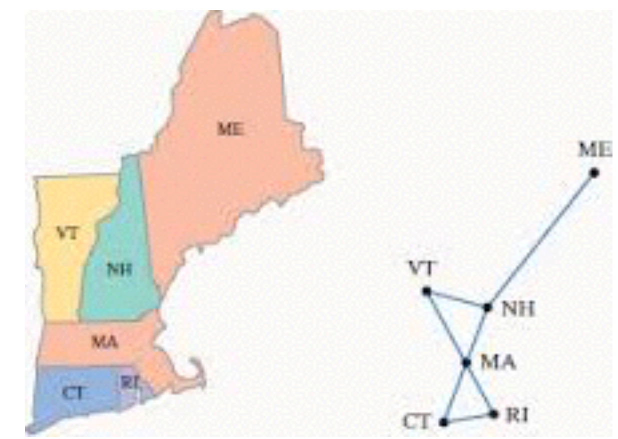
Chromatic Number of a Graph

- As we mentioned earlier, coloring the regions of a map, without neighboring regions sharing a color, corresponds to a graph problem.
- We make a node for each region and an edge between neighboring regions.
- The New England map can be done with 3 colors but not 2.

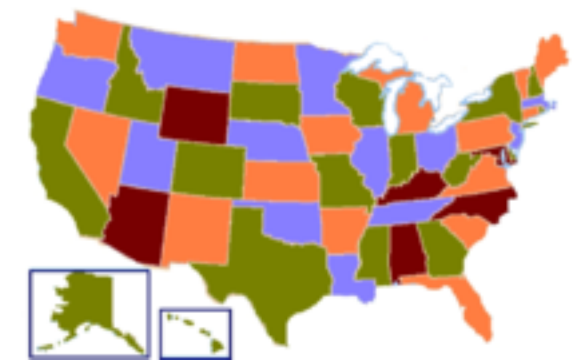


Chromatic Number of a Graph

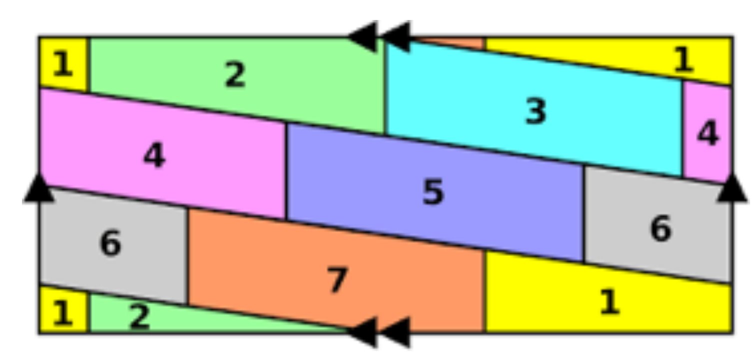
- A **vertex coloring** of a graph is a labeling of the vertices with colors such that no vertices that share an edge also share a color.
- An **edge coloring** is a labeling of the edges with colors such that no two edges of the same color share a vertex.
- The **chromatic number** of a graph is the minimum number of colors in any vertex coloring.



Chromatic Number of a Graph



- Appel and Haken proved in 1976 that ever planar graph can be 4-colored, confirming a conjecture from 1852. Their proof used a computer and an analysis with 1936 cases to check.



- It turns out that for graphs embedded on a **torus**, seven colors are necessary and sufficient. The graph of the map at left is K_7, if we identify the boundary edges as shown.

Applications of Coloring

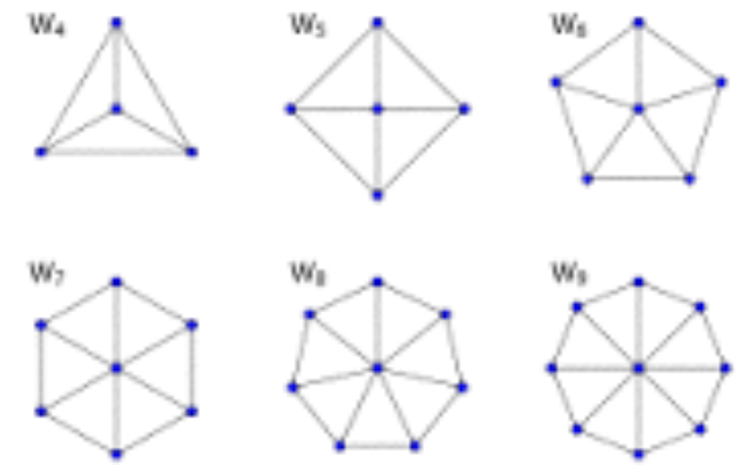
- We already had an example in Section 1.1 where we reduced a combinatorial problem to graph coloring.
- We wanted to schedule meetings of as many committees as possible, without giving any member two meetings at the same time.
- We have a node for each committee and an edge for every pair of committees that share a member. The chromatic number of this graph is the total number of time slots needed.

Basic Examples

- The only connected graph with chromatic number of 1 is the graph with one node and no edges.
- We've already seen that a graph can be 2-colored if and only if it has no odd cycle.
- The graph K_n has chromatic number n.
- In a k-coloring, any vertex of degree $< k$ is easy to color whatever happens elsewhere.

Coloring a Wheel

- The wheel graph W_n has n-1 nodes in a cycle and the n^{th} node connected to each of the others.
- It is easy to see that W_n has chromatic number 3 if n is odd and 4 if n is even.
- The map of the continental USA cannot be 3-colored because it contains a W_6 around Nevada.



NP-Completeness Overview

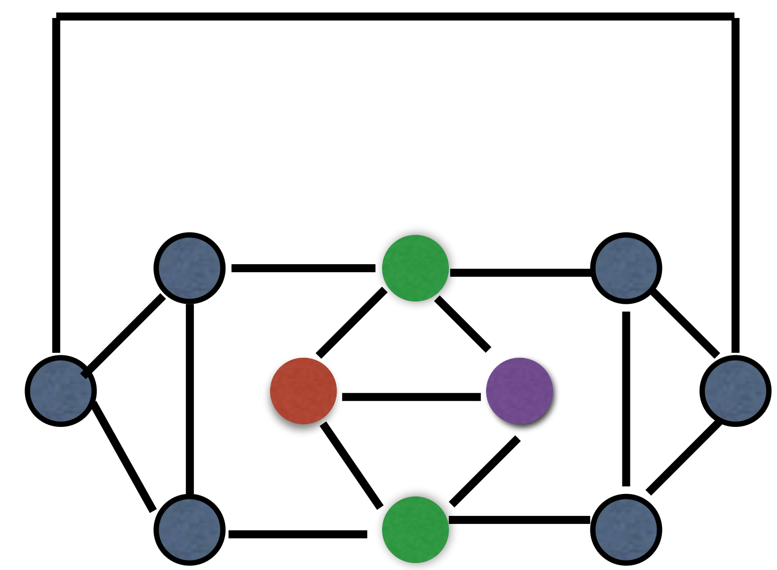
- Appendix A.5 of Tucker gives an informal overview of **NP-completeness**, a concept dealt with here in CS 311 and CS 501.
- The NP-complete problems have no solutions in polynomial time, unless the statement P = NP is true, meaning that all problems whose solutions are verifiable in poly-time are actually solvable in poly-time. Few people believe this is true, but proving so is one of the most famous open problems in math.

NP-Completeness Overview

- The way we prove something NP-complete is to take a known NP-complete problem and show that a poly-time solution to our problem with also solve that.
- The original NP-complete problem is called **satisfiability**. We'll look at a variant of it called **3-SAT**.
- Given n boolean variables and a number of **clauses** of the form $L_1 \vee L_2 \vee L_3$, where the L_i's are variables or negated variables, can we set the variables to make all the clauses true?

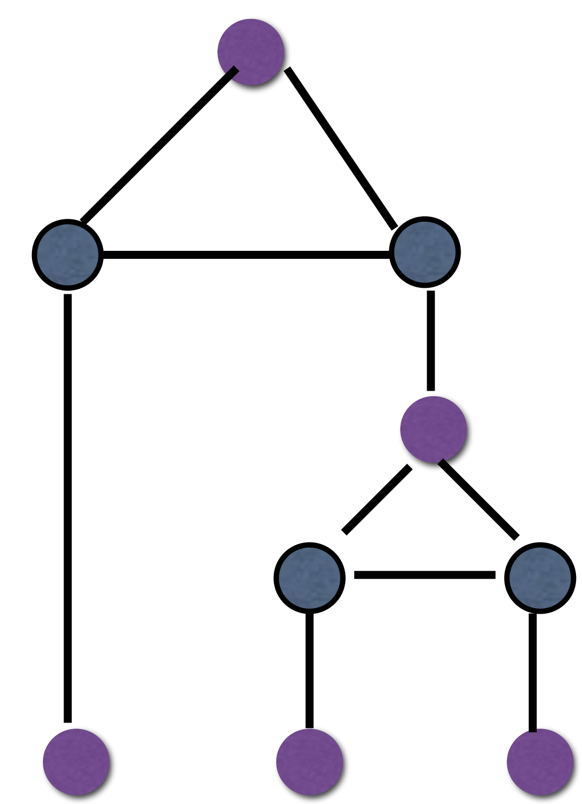
3-Colorability

- Here's Example 2 of section 2.3 in Tucker, a non-3-colorable graph.
- In any 3-coloring, the two middle nodes must have different colors, so the two nodes I have in green must be the same color.
- That forces the two side points to also be green, but they are connected to one another.



3-Colorability

- Given n variables, we make a node for each literal and force each of those nodes to be red or blue.
- Then, for each clause, we connect a copy of the gadget at right to the three literals of that clause. We force all the purple nodes to be red or blue.
- If all three lower nodes become red, the top node cannot be blue. We can finish only if the clause is satisfied.



Application to Garbage Trucks

- In Example 5 of Section 2.3, Tucker explains a problem on which he consulted for the NYC government. They wanted routes for garbage trucks that did not conflict and met certain constraints.
- They could construct short legal tours, and wanted to combine them into longer ones, but needed to assign each tour to a day of the week to avoid conflicts.

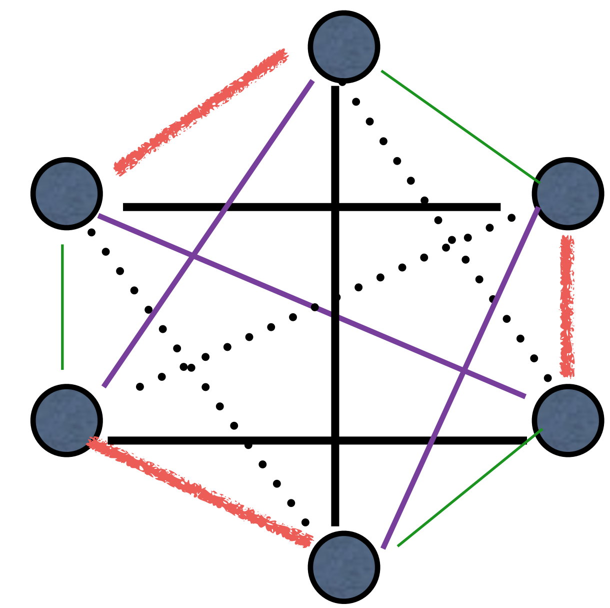
Application to Garbage Trucks

- Assigning tours to days meant six-coloring a graph where the nodes were tours and edges were pairs of tours that shared a point.
- Combining two tours into one meant collapsing an edge to merge two vertices. Their algorithm did this as much as it could, while leaving the remaining graph 6-colorable.

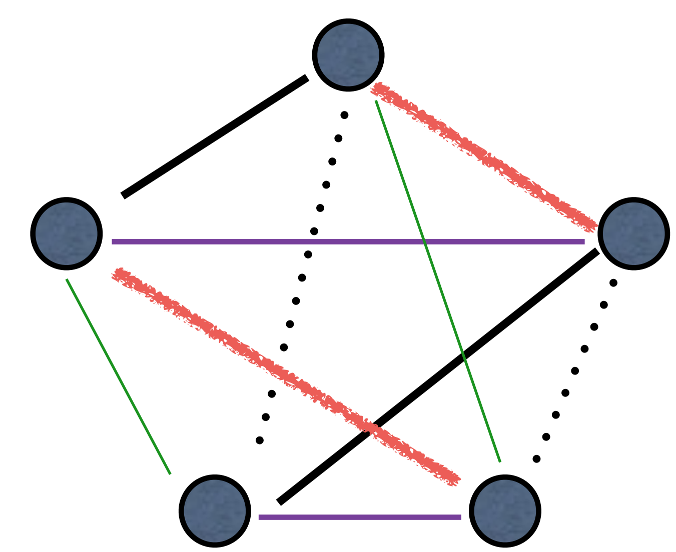
Scheduling Tournaments

- Our last application involves round-robin tournaments again. We have n teams, each of which plays each of the others once, and we want to assign each match a day so that no team plays twice on the same day.
- This means giving an edge coloring of the complete graph K_n. The total number of edges to color is $n(n-1)/2$.
- We cannot have more than $n/2$ edges of a single color, since each team plays only once.

Scheduling Tournaments



- For even n, n(n-1)/2 edges with n/2 to a color means n-1 colors. With n = 6, we have a 5-coloring at right, and similar colorings are possible for any even n.



- For odd n, we have at most (n-1)/2 edges per color and so need at least n colors, which is doable for any odd, n similar to this picture.