MA/CS-109: Graphs as Models of Conflicts
Leo Reyzin
Thanks to Azer Bestavros for the Slides
What else can graphs model?

- Edges represent a relationship between vertices:
  - For maps, the relationship is “adjacency”

- These relationships do not have to be between physical things:
  - For web networks, the relationship is “linkage”
  - For social networks, the relationship is “friendship”
  - For news networks, the relationship is “subscription”
  - For curricular networks, the relationship is “prerequisite”
  - For neuron networks, the relationship is “interaction”
  - ...

So far, we have seen how a graph is used to model a (subway or internet) map. In such a model, the edges in the graph represented the “adjacency” relationship between the vertices in the graph – two vertices (e.g., subway stops or Internet routers) have an edge between them if they are adjacent to one another. For the subway map, adjacency means that the two stations (modeled by the vertices) are within one stop from one another on a subway line. For the Internet map, adjacency means that the two routers (modeled by the vertices) are directly connected with a physical wire.

In general, the edges of a graph could represent any relationship between the vertices – e.g., linkage between web pages, friendships between users on Facebook, subscribers to a news feed or followers of a user on Twitter, prerequisite relationships between courses, interactions between neurons in a brain, ...
What else can graphs model?

- Edges do not have to represent “harmonious” relationships; they may represent conflicts!

In all the examples we considered so far, the relationship that edges represented was “harmonious” – but that is not necessarily the case. Edges in a graph can also represent conflict relationships!
Graphs as models of conflicts

- Edges do not have to represent “harmonious” relationships; they may represent conflicts!
  - For Enemybook, the relationship is “hate”
  - For predatory networks, the relationship is “eat”
  - For prescription drugs, the relationship is “-ve reaction”
  - For radio frequency, the relationship is “interference”
  - For course scheduling, the relationship is “conflict”
  - …

- Why are “conflict” graphs interesting?

One may use a graph to represent animosity between users (instead of friendship), or a predatory relationship between animals (instead of a symbiotic relationship), …
Resolving conflicts using graphs

- **Example applications:**
  - Transporting live fish to pet stores
  - Seating guests in a wedding reception
  - Assigning broadcast frequencies to local radio stations
  - Coloring of geographic maps
  - Assigning time slots to classes

- **Common problem:**
  - Find the *minimum* grouping of nodes such that no two nodes in a single group conflict
  - Group = fish tank, table at a wedding, broadcast frequency, map color, course time-slot, ...

Many applications require us to reason about such conflicts:
- We would not want to put fish that eat one another in the same tank
- We would not want to seat sworn enemies on the same table at a United Nations function (or one’s wedding for that matter!)
- We would not want to assign the same frequencies to radio stations broadcasting in close proximity from one another
- We would not want two states on the US map to have the same color (the map will be confusing)
- We would not want to schedule exams for two classes in the same time slot if a student is taking both of these classes
Graph (vertex) coloring problem

- What is the minimum number of colors needed to color all vertices in a graph such that no edge would connect vertices of the same color?

One such interesting question/problem is “graph vertex coloring”, namely what is the minimum number of colors needed to color vertices of a graph in such a way that any two adjacent vertices (i.e., vertices connected by an edge) are of different color. Finding an answer to this question is very important for many problems.
Graph (station) coloring problem

The Federal Communications Commission (FCC) prevents interference between radio stations by assigning appropriate frequencies to each station. **Two stations cannot use the same channel when they are within 150 miles of each other.** How many different frequencies are needed for the six stations located at the distances shown in the table?

<table>
<thead>
<tr>
<th></th>
<th>WBUR</th>
<th>WGBH</th>
<th>WROR</th>
<th>WKIS</th>
<th>WFXI</th>
<th>WABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBUR</td>
<td>-</td>
<td>25</td>
<td>202</td>
<td>77</td>
<td>375</td>
<td>106</td>
</tr>
<tr>
<td>WGBH</td>
<td>-</td>
<td>175</td>
<td>51</td>
<td>148</td>
<td>222</td>
<td></td>
</tr>
<tr>
<td>WROR</td>
<td>-</td>
<td>111</td>
<td>305</td>
<td>411</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKIS</td>
<td>-</td>
<td>78</td>
<td>297</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WFXI</td>
<td>-</td>
<td>227</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WABC</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Along the same lines, one can use graph vertex coloring to solve other (quite important) problems, including assigning the minimum number of radio frequencies to various radio transmitters who may interfere with one another – here the radio stations would be the vertices, and if two stations interfere with one another (because of geographical proximity) then we draw an edge between them. Now coloring the graph is akin to assigning frequencies since we would not want to give the same frequency (color) to interfering stations (adjacent vertices).
Graph (station) coloring problem

The Federal Communications Commission (FCC) prevents interference between radio stations by assigning appropriate frequencies to each station. **Two stations cannot use the same channel when they are within 150 miles of each other.** How many different frequencies are needed for the six stations located at the distances shown in the table?

<table>
<thead>
<tr>
<th></th>
<th>WBUR</th>
<th>WGBH</th>
<th>WJOR</th>
<th>WKIS</th>
<th>WFXI</th>
<th>WABC</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBUR</td>
<td>-</td>
<td>X</td>
<td>202</td>
<td>X</td>
<td>375</td>
<td>X</td>
</tr>
<tr>
<td>WGBH</td>
<td>-</td>
<td>175</td>
<td>X</td>
<td>X</td>
<td>222</td>
<td></td>
</tr>
<tr>
<td>WJOR</td>
<td>-</td>
<td>X</td>
<td>365</td>
<td>411</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKIS</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>297</td>
<td></td>
</tr>
<tr>
<td>WFXI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>WABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

By marking in the table the interfering stations (shown with X) we identify the conflict relationships.
Graph (station) coloring problem

Which leads us to the conflict graph that we can now color to figure out the minimum number of frequencies needed.

How many colors (frequencies) would be needed??
Three colors (radio frequencies) will do it! Can you see why it cannot be done in less?
Yet, another example is the problem of arranging guests attending a wedding around tables. Given a list of “irreconcilable differences” between invitees, you task is to find a seating arrangement that uses the least number of tables while avoiding to have two individuals with “irreconcilable differences” sitting around the same table. This is a graph coloring problem! Namely, we can solve the problem by (1) modeling each individual as a vertex in a graph, (2) representing the fact that two individuals have irreconcilable differences by drawing an edge between the vertices corresponding to these individuals, and (3) minimizing the number of colors (tables) used to color the vertices (seat the guests) such that no two vertices connected with an edge (two guests with irreconcilable differences) have the same color (are sitting on the same table).

Along the same lines, one can use graph vertex coloring to solve other (quite important) problems, including assigning the minimum number of radio frequencies to various radio transmitters who may interfere with one another – here the radio stations would be the vertices, and if two stations interfere with one another (because of geographical proximity) then we draw an edge between them. Now coloring the graph is akin to assigning frequencies since we would not want to give the same frequency (color) to interfering stations (adjacent vertices).

Yet, another famous application of graph coloring is “map coloring”. Here our job is to color adjacent states (vertices) on a planar map (e.g., map of the US) with different colors, but use the minimum number of colors. This fairly classical problem was settled by proving that one need no more than 4 colors for such (planar) graphs.
Graph (guest) coloring problem

The attempt on the left shows a coloring (assignment of guests to tables) that uses 4 colors (i.e., we need 4 tables). The one on the right shows a coloring that uses only 3 colors. One can show that for this graph, three is the minimum number of colors. Try to prove it!

Accordingly, we would seat A, B, and E on the first (red) table; D, G, and H on the second (green) table; and C, and F on the third (blue) table.
At least one student is enrolled in “Science” and in “English”.

For example, consider the problem of assigning time-slots to the final exams of seven classes (say in Science, Math, Accounting, English, Music, Economics, and Art). Now consider students enrolling in these classes. If a student is enrolled in two classes (say English and Music) then these two classes cannot be assigned the same time slot for the final exam, since the student taking both classes cannot take both finals in the same time. We can model this “conflict” relationship between classes using a graph and by finding the minimum number of colors for that graph, we would identify the minimum number of exam slots needed (in this case, each color would correspond to one exam slot).
Yet, another famous application of graph coloring is “map coloring”. Here our job is to color adjacent states (vertices) on a planar map (e.g., map of the US) with different colors, but use the minimum number of colors.

Note: You may ask why is it important to use a small number of colors. There are at least two reasons: printing costs increase as the number of colors used increases and fewer colors means we can pick more contrasting ones.
Graph (map) coloring problem

Here is a not-so-good attempt that uses too many colors...
Here is an attempt at a coloring with less colors (using 5 colors)…
Incidentally, this fairly classical problem was settled by proving that one need no more than 4 colors for such (planar) graphs – coloring is shown above.
Graph coloring problem

- **Graph Vertex Coloring**
  - What is the minimum number of colors needed to color all vertices in a graph such that no edge would connect vertices of the same color?
    - For a complete graph with N vertices: N colors (proof?)
    - For any planar graph: No more than 4 colors
    - For arbitrary graphs: No efficient way to figure this out!

- What do we mean by “no efficient way”?

So, how many colors do we need?

Well, the answer is relatively easy for some special graphs:

1. For a complete graph with N nodes (a graph where each pair of vertices are connected with an edge), we would obviously need N colors. One can prove this very easily by contradiction – Assume that one can color a complete graph with less than N colors such that no two adjacent vertices have the same color. Since the graph has N vertices, then it must be the case that at least two vertices have the same color. But since there is an edge between any pair of vertices, it follows that two adjacent vertices have the same color, which is a contradiction.

2. For a ring with N nodes (a graph where each vertex is connected to exactly one other vertex forming a connected cycle). One can prove that if N is even, then one needs two colors, and if N is odd, one needs 3 colors.

3. For a planar graph (a graph that can be drawn on a plane without having any of the graph’s edges intersect), as we mentioned before, it was shown that one needs at most 4 colors...

How about arbitrary graphs? In fact, it turns out we don’t even know an efficient algorithm that will tell whether a graph can be colored with three colors – and yet, if a graph can be colored in three colors, I can be easily convinced of that fact if only someone shows me the correct coloring.

Turns many problems have similar properties, and we now turn to discussing a very special class of problems for which no efficient algorithm is known: so-called NP problems.