Greedy Algorithms: Graph Coloring

Textbook: Mentioned several times, but not covered in-depth. Look in the index under “graph coloring”.


How many colors do you really NEED?

Map coloring

- Problem: Color the regions on a map
  - Regions that share a border must be different colors
  - Meeting at a single point is not a border

- As a decision problem:
  - Can this map be colored with N colors?

- As an optimization problem:
  - What is the minimum number of colors needed to color this map?

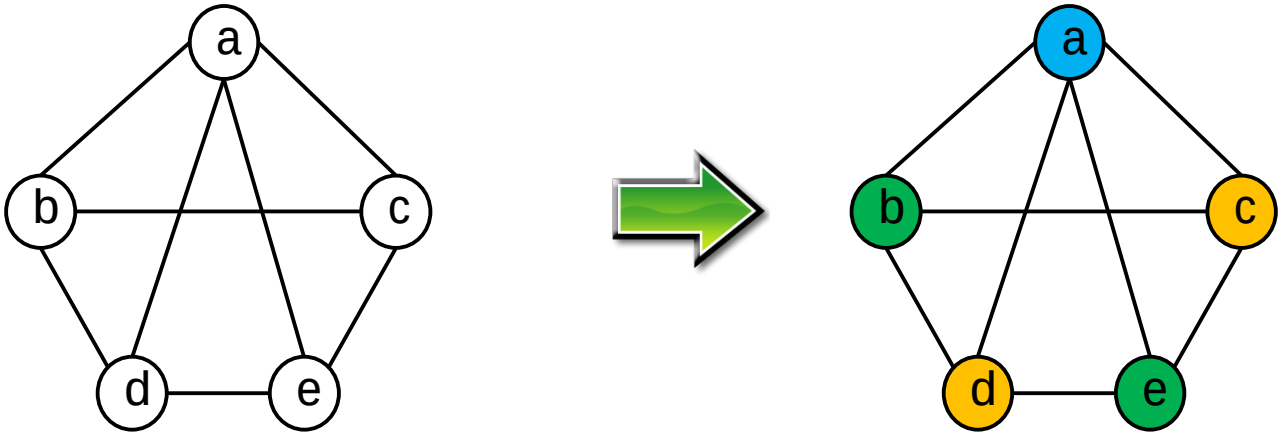
Graph representation

- One vertex for each region
- Edge between regions if they share a border

- Problem re-stated as a graph problem:
  - Assign colors to the vertices of a graph so that no adjacent vertices are the same color

Graph coloring problem

- Color a graph with as few colors as possible such that no two adjacent vertices are the same color
- Example:

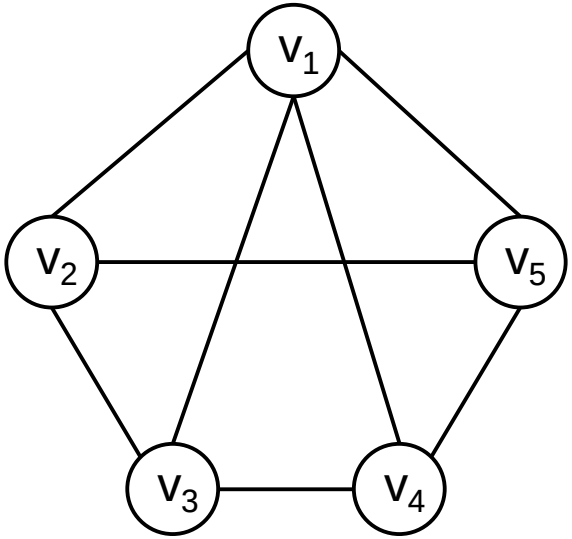


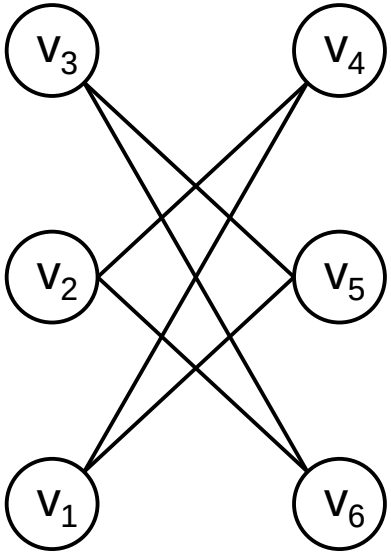
We say that this graph is *3-colorable*

Graph coloring – greedy algorithm

- Start with just one color
- Consider the vertices in a specific order $v_1, \ldots, v_n$
- For each v_i , assign the first available color not used by any of v_i's neighbours
- If all colors are in use by neighbours, add a new color

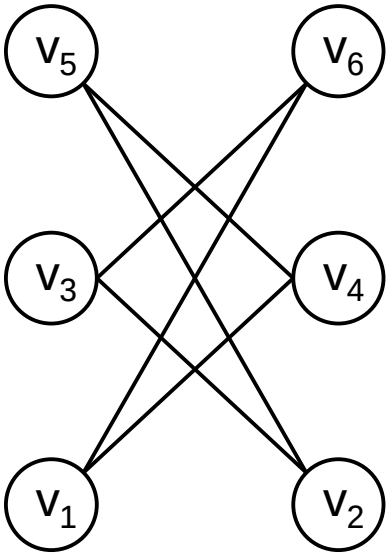
Examples





Is this algorithm optimal?

- Consider the previous graph but with vertices numbered differently



- Needed only two colors before
- The order of considering the vertices matters

- Greedy algorithms do not always yield optimal solutions

- But like brute-force, they are often worth considering because they may be easy to implement

Puzzle – just for fun!

- Make a graph that *requires* 4 colors

*PFG for first person to give a correct answer