Coloring a Graph
Graph Coloring

Given an undirected graph, can we assign a color to each vertex such that no adjacent vertices have the same color?

– If we have \(|V|\) colors, then yes.
– What if we have 2 colors? 3? k?

We actually did the 2-coloring problem as a homework problem. What was its runtime?

How much harder do you think deciding whether 3-coloring can be done will be?
Coloring a Graph

Let’s consider the following algorithm for coloring a graph:
– Number your vertices from 1 to $|V|$.
– Assign color 1 to vertex 1.
– for $i=2$ to $|V|$ {color vertex $i$ with the lowest color number that has not been assigned to one of its neighbors}

See how many colors you used…
– What is the runtime of this?
– Is this guaranteed to be an optimal coloring of any given graph in terms of the number of colors used?
Coloring a Graph Differently

Let’s consider the following modified version of that algorithm which I will call *GreedyAppxColor* for coloring a graph:

– Sort the vertices in descending order based on their degree and then number them from 1 to |V| where vertex 1 has the highest degree.
– Assign color 1 to vertex 1.
– for i=2 to |V| {color vertex *i* with the lowest color number that has not been assigned to one of its neighbors}

• What is the runtime of this?
• Is *this* guaranteed to be an optimal coloring of any given graph?
3-Color

The 3-coloring problem is NP-Complete!

We will soon discuss exactly what this means...
Specific types of graphs…

There are proofs and conjectures about certain types of graphs and the ability to color them with various numbers of colors…

A valid proof closes the question.

A conjecture is really just a guess. It might be a reasonable-sounding guess made by a well-respected person, but it is still a guess…
How could you convert a Sudoku game into a graph problem?

– What are the vertices?
– What are the edges?