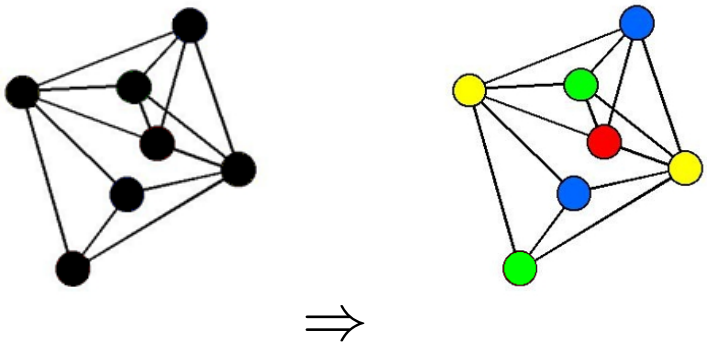
Homework 14 sample solution

Due 10/28/16

October 25, 2016

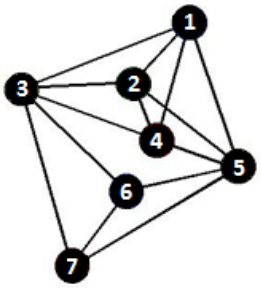
The *graph coloring* problem takes in a graph and asks what is the fewest number of colors you could use to color the graph vertices so that no adjacent vertices are the same color. For example, the graph below requires a minimum of 4 colors to produce a valid coloring (sample coloring on right):



Graph coloring is useful in scheduling, as well as register assignment in compilers (vertices are variables, edges connect variables active at the same time, and colors are registers) and other applications.

The greedy coloring algorithm uses colors numbered 1 to n. It iterates through vertices of the graph in order of vertex ID and assigns each vertex the minimum color that is not already assigned to one of its neighbors. The return value is just the largest color assigned to a vertex.

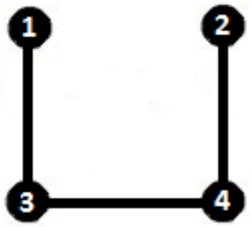
For example, if the graph above had vertex IDs numbered from top to bottom, the algorithm would operate as follows and return 4:



Vertex #	$N(v)$ colors	Color assigned
1	none	1 (blue)
2	1	2 (green)
3	1, 2	3 (yellow)
4	1, 2, 3	4 (red)
5	1, 2, 4	3 (yellow)
6	3	1 (blue)
7	1, 3	2 (green)

Prove that the greedy algorithm is not correct.

Many possible answers; e.g.,



This graph only requires two colors (alternate blue and green), but greedy coloring will use three.