8 Sept 2014

Math 181

Spotlight 3.4: The naive way of solving the interview scheduling problem is to draw the 14 vertices (companies) and start adding edges between pairs that have conflicting time slots. Once finished, you still have to find a proper coloring using only 5 colors.

Other graph representations can be useful, and I want to show you one such example. Instead of placing 14 "dots" for vertices, let's use the fact that each company is requesting an interval of timeslots and represent the vertices with line segments. This representation is called an *interval graph*, and it is much easier to color.

Greedy Algorithm: In many cases, determining the chromatic number of a graph is NP-complete, which means that for large graphs, it is hopeless even for a computer to solve. If we are not concerned about finding the fewest number of colors, can we find a "good enough" solution faster? If I give you an arbitrary graph, how would you go about trying to properly color it? In terms of vertex valence, what is an upper bound on the number of colors that you will need?

Suppose I visit the vertices one at a time according to some order. When I visit a vertex, I will use the smallest color that does not appear on its neighbors. If M is the largest valence of a vertex, then I will never need more than $M + 1$ colors (M neighbors, plus 1 color for the vertex itself).

Can you think of any graphs where you can do no better? Explicitly, can you think of a graph whose valence is M everywhere, but cannot be properly colored with M colors.

Petersen Graph: One *very* famous graph is the *Petersen Graph*. It is named for Julius Petersen, who was a Danish mathematician and graph theorist in the late 1800s. The graph is special because it frequently shows up as a "special case" to many coloring and structural theorems.