Assignment 6: Graph Colouring

Due: Mon Apr 9th, 23:59. Worth 3% of your course grade.

Overview: You will write a program for “two-colouring” graphs.

Graphs: Recall that a graph is a set of vertices (also called nodes) and a set of edges that connect them. We draw graphs using circles for vertices and lines for edges, for example:

In this assignment, we will work only with graphs whose edges have no direction.

Colouring: Graph colouring is the problem of assigning colours to the vertices in a graph such that no two neighbour vertices (vertices that are joined directly by an edge) have the same colour. Depending on the graph and the number of colours you have, there may be no solution. Clearly, the problem gets easier the more colours you have. A $k$-colouring is one that uses $k$ colours.

All of this may seem rather esoteric, but it is extremely useful because finding graph colourings corresponds to many practical problems. For example, if vertices represent courses and an edge between two courses indicates that there are some students taking both courses, then finding a $k$-colouring of a graph corresponds to choosing time slots for an exam schedule. The $k$ colours correspond to $k$ different time slots.

Program Spec:
Write a program that reads a description of a graph from the standard input and then writes to the standard output either a description of a two-colouring for that graph, or just “no” if no two-colouring could be found.

The graph description will consist of the following:

- On the first line: the number of vertices in the graph, $n$.
  (We will assume that the vertices are numbered from 0 to $n$.)
- On the second line: the number of edges in the graph, $e$.
- On each of $e$ subsequent lines: a pair of integers, separated by a blank. A pair of integers $a$ and $b$ indicates that there is an edge from vertex $a$ to vertex $b$.

When a two-colouring is found, the output should be in this format: For each vertex $v$, print its number followed by a blank and then either 0 or 1 (to indicate one of the two colours). Print each of these pairs on a separate line.

Please follow this input and output format carefully; we will be autotesting this program.
**Program design:** Use good object-oriented design. Write a separate graph class, and make it general. Your graph class must use an adjacency list for storing the graph (see chapter 11 of your text), and each list must be an instance of a general set class. You may reuse your set code from assignment 3 if you wish.

Your two-colouring function must be outside your graph class; that is, it must not be a member function. Use a simple greedy algorithm for colouring. (Recall that we discussed greedy algorithms briefly in the section on "Reasoning about Programs"). Just start at node 0 and give it a colour. That forces all of its neighbours to be the other colour. Continue in this fashion until everything is coloured or you run into a conflict, in which case you may give up. Thought exercise: When you give up in such a case, are you sure there was no solution, or was the algorithm just bad?

**How to submit:** Details will be available on the web site.

Note that we will not require that your program produce the same two-colourings as ours, just that it produce valid two-colourings, printed in the required format.