CSE 5311-001 Lab Assignment 2

Due April 11

Goals:

Understanding of edge coloring

Requirements:

1. Write (and test) a C program to *edge color* a provided *bipartite* graph.

    The input to your program will be:

    a. n_l, n_r, and *m*, the numbers of left column vertices, right column vertices, and edges. $n_l \le n_r \le 50$ and $m \le 1000$.
    b.   *m* lines, each giving an edge: left column vertex, right column vertex. Left column vertices are numbered 0 . . . n_l - 1. Right column vertices are numbered 0 . . . n_r - 1. Duplicate edges will not occur.

    The output from your program will be:

    a.   A trace of the processing. Each edge will be processed by either 1) using a color that is "free" at both incident vertices, or 2) using an alternating (*a*, *b*) path. For (1), simply indicate the free color that is used for the edge. For (2), indicate the colors (e.g. numbers) for *a* and *b* along with the vertices on the path.
    b.   A list of the edges (input order) and the final color for each edge.

2. Submit your C code on Canvas before 3:45 p.m. on April 11. Your code must compile and execute on `omega.uta.edu`.

Getting Started:

1. Bipartite edge coloring is discussed in Notes 11. Do not use the approach for general graphs, since the number of colors you may use is bounded by the degree (Δ) of the bipartite graph (not Δ + 1).

2. Assigned colors must be in the range 0 . . . Δ - 1. You will need to preprocess the edges to determine Δ.

3. Do not use backtracking instead of the alternating (*a*, *b*) path technique.

4. It is useful to have several tables. Rather than having a table of free colors for each vertex, it is useful to have a table that indicates for a given vertex and color the incident edge (if any) with that color.