

CSE 5311 Lab Assignment 3

Due May 1, 2019

Goals:

Understanding of edge coloring

Requirements:

1. Write (and test) a C/C++ program to *edge color* a provided *bipartite* graph. Your program must compile and execute on at least one of OMEGA or Visual Studio.

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. $n_l \leq n_r \leq 50$ and $m \leq 1000$.
- b. m lines, each giving an edge: left column vertex, right column vertex. Left column vertices are numbered $0 \dots n_l - 1$. Right column vertices are numbered $0 \dots n_r - 1$. 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 .
 - b. A list of the edges (input order) and the final color for each edge.
2. Submit your C/C++ code on Blackboard before 3:45 p.m. on May 1. Please provide only source code files without compressing/zipping. Be sure to include comments regarding how to compile and execute your code.

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 $\Delta + 1$).
2. Assigned colors must be in the range $0 \dots \Delta - 1$. 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.