CS41 Lab 4

September 23, 2019

In typical labs this semester, you'll be working on a number of problems in groups of 3-4 students. You will not be handing in solutions; the primary purpose of these labs is to have a low-pressure space to discuss algorithm design. However, it will be common to have some overlap between lab exercises and homework sets.

The goal of this lab session is to gain more practice designing graph algorithms. Do not expect to complete all parts of all problems by the end of the lab. Work on the problem(s) that sound the most interesting to your group.

1. **k-coloring.** A graph is *k-colorable* if it's possible to color each vertex using one of k colors such that the endpoints of each edge have different colors. Design and analyze an $O(n+m)$-time algorithm that colors a graph using $\Delta + 1$ colors, where Δ is the largest degree of a vertex.

2. **Testing Tripartiteness.** Call a graph $G = (V, E)$ *tripartite* if V can be partitioned into disjoint sets A, B, C such that for any edge $(u, v) \in E$, the vertices u, v lie in different sets. Design and analyze an algorithm to test a graph for tripartiteness.

3. **Strongly Connected Components.** Let $G = (V, E)$ be a directed graph. Vertices u and v are *strongly connected* if there are $u \rightsquigarrow v$ and $v \rightsquigarrow u$ paths in G. A *strongly connected component* is a set of vertices $C \subseteq V$ such that u, v are strongly connected for all $u, v \in C$ (and no other vertices are strongly connected to a vertex $u \in C$.)

   Design and analyze an algorithm to identify all strongly connected components in G. What is the runtime of your algorithm?