CS41 Lab 4

September 22, 2022

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 learning goals of this lab session are to gain more experience with undirected graphs and to practice algorithm design for graph problems. There should be more problems on this writeup than you have time to complete. Consider the lab a success if you can make good progress on two problems.

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. **Paths in Graphs.** A path P in a graph $G = (V, E)$ is a sequence of vertices $P = [v_1, \ldots, v_k]$ such that for all $1 \leq i < k$ there is an edge $(v_i, v_{i+1}) \in E$. P is *simple* if all v_i's are distinct.

   In this problem, you will examine different graphs and consider how many different paths can exist in the graph.

   (a) Describe a graph G_1 on n vertices where between any two distinct vertices there are zero simple paths.

   (b) Describe a graph G_2 on n vertices where between any two distinct vertices there is exactly one simple path.

   (c) Describe a graph G_3 on n vertices where between any two distinct vertices there are exactly two simple paths.

   (d) Describe a graph $G_4 = (V, E)$ on n vertices and two distinct vertices $s, t \in V$ such that there are $2^{\Omega(n)}$ simple $s \rightsquigarrow t$ paths.

3. **Testing Bipartiteness.** A graph $G = (V, E)$ is *bipartite* if the vertices V can be partitioned into two sets $A, B \subseteq V$ such that for any edge $(u, v) \in E$, the vertices u, v lie in different sets. In other words, a graph G is bipartite if it is 2-colorable.

   Design an analyze a $O(n+m)$ algorithm that takes a graph G as input and outputs YES if G is bipartite, and outputs NO otherwise.

4. **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. In other words, a tripartite graph is three-colorable.

   (a) Design and analyze an algorithm which takes as input an undirected graph $G = (V, E)$ and returns YES if G is three-colorable, and NO otherwise.

   (b) Design and analyze an efficient algorithm which takes as input a *three-colorable* graph $G = (V, E)$ and colors the vertices of the graph using $O(\sqrt{n})$ colors. (Note: while the input graph *is* three-colorable, it does not mean that we know what that coloring is!)