Problem Set #5

You should try to solve these problems by yourself. I recommend that you start early and get help in office hours if needed. If you find it helpful to discuss problems with other students, go for it. **You do not need to turn these problems in. The goal is to be ready for the in class quiz that will cover the same or similar problems.**

Problem 1: Graph Theory

A connected, undirected graph is *vertex biconnected* if there is no vertex whose removal disconnects the graph. A connected, undirected graph is *edge biconnected* if there is no edge whose removal disconnects the graph. Give a proof or counterexample for each of the following statements:

(a) A vertex biconnected graph where $|E| > 1$ is edge biconnected.

(b) An edge biconnected graph is vertex biconnected.

Problem 2: 2-Colorable Graphs

An undirected graph $G = (V, E)$ is said to be k-colorable if all of the vertices of G can be colored one of k different colors such that no two adjacent vertices are assigned the same color. Design an algorithm based on BFS that either colors a graph with 2 colors or determines that two colors are not sufficient. Argue that your algorithm is correct.

Problem 3: Depth First Search

(a) During the execution of depth first search, we refer to an edge that connects a vertex to an ancestor in the DFS-tree as a *back edge*. Either prove the following statement or provide a counter-example: if G is an undirected, connected graph, then each of its edges is either in the depth-first search tree or is a back edge.

(b) Suppose G is a connected undirected graph. An edge whose removal disconnects the graph is called a *bridge*. Either prove the following statement or provide a counter-example: every bridge e must be an edge in a depth-first search tree of G.

Problem 4: Discipline in Groups of Children

Your job is to arrange n rambunctious children in a straight line, facing front. You are given a list of m statements of the form "i hates j". If i hates j, then you do not want to put i somewhere behind j because then i is capable of throwing something at j.

(a) Give an algorithm that orders the line (or says it's not possible) in $O(m + n)$ time.

(b) Suppose instead that you want to arrange the children in rows such that if i hates j, then i must be in a lower numbered row than j. Give an efficient algorithm to find the minimum number of rows needed, if it is possible.