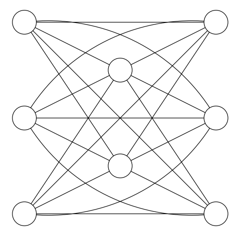
Disclaimer:
These are practice problems for the upcoming final exam. You will be given a sheet of notes for the exam. Also, go over your homework assignments. **Warning**: This does not necessarily reflect the length, difficulty, or coverage of the actual exam.

Problem 1 A graph is tripartite if the vertices can be partitioned into three sets so that there are no edges internal to any set. The *complete* tripartite graph, $K(a, b, c)$, has three sets of vertices with sizes a, b, and c and all possible edges between each pair of sets of vertices. $K(3, 2, 3)$ is pictured below. A *Hamiltonian* cycle in a graph is a cycle that traverses every vertex exactly once.



(a) For which values of n does K(1,1,n) have a Hamiltonian cycle.

(b) For which values of n does K(1,n,n) have a Hamiltonian cycle.

(c) For which values of n does K(n,n,n) have a Hamiltonian cycle.

Problem 2 Let $G = (V, E)$ be an undirected graph. A *triangle* is a set of three vertices such that each pair has an edge.

(a) Give an efficient algorithm to find all of the triangles in a graph.

(b) How fast is your algorithm?

Problem 3 Show that you can convert a formula in Conjuctive Normal Form (CNF) where every clause has *at most three* literals, into a new formula where every clause has *exactly* three literals, so that the new formula is satisfiable if and only if the original formula is satisfiable. No variable may occur twice in the same clause.

Problem 4 A *vertex cover* in a graph $G = (V, E)$ is a subset of vertices such that every edge is incident on at least one vertex of the subset. The *Weighted Vertex Cover Problem* (WVCP) is, given a graph $G = (V, E)$ with integer weights on the vertices, find a vertex cover whose sum of weights is as small as possible. You can assume that the weights are between 1 and n (inclusive).

(a) WVCP is an optimization problem. Define a decision version of WVCP.

(b) Show that the decision version is in **NP**. Make sure to state the certificate and give the pseudo code.

(c) Show that if you could solve the optimization version in polynomial time that you could also solve the decision version in polynomial time.

(d) Show that if you could solve the decision version in polynomial time that you could also solve the optimization version in polynomial time. HINT: First find the weight of an optimal weighted vertex cover.