Your homework submissions need to be typeset (hand-drawn figures are OK). See the course web page for suggestions on typing formulas. Solution to each question needs to be uploaded to CMS as a separate pdf file.

Remember that when a problem asks you to design an algorithm, you must also prove the algorithm's correctness and analyze its running time. The running time must be bounded by a polynomial function of the input size.

If A and B are two decision problems, proving A $\leq$ B often just consists of a polynomial-time algorithm that takes an instance of A and outputs an instance of B. To show that such a reduction is correct, you must prove two things. First, that it transforms YES instances of A to YES instances of B. Second, that it transforms NO instances of A to NO instances of B.

Problems 1 and 3 are asking you to prove a problem NP-complete. The list of problems we proved NP-complete in class so far are SAT, Independent Set, Vertex-Cover. You can use in your proof that these problems, as well as the Clique problem defined below, are NP-complete.

(1) *(10 points)* Consider a strong version of SAT where we would need each clause to have at least 2 true variables. We say that a SAT formula is *very satisfiable* if there is a truth assignment so that there are at least two different true variables in each clause. The VERY-SAT problem is to decide if a given SAT formula Φ has a very satisfying truth assignment.

Show that the VERY-SAT problem is NP-complete.

(2) *(10 points)* In class, we stated without proof the important fact that SAT is NP-hard, which means that every problem in NP reduces to SAT in polynomial time. We have seen more of these reductions, that SAT is at least as hard as matching and also independent set.

To gain more intuition about how these reductions work, this exercise asks you to give a polynomial-time reduction from a particular NP problem to SAT. **For this exercise, your solution *may not* use the fact that** SAT (or any other problem) **is NP-hard.**

The GRAPH ISOMORPHISM problem is defined by two undirected graphs $G = (V, E)$ and $G' = (V', E')$ with $|V| = |V'|$, and asks the question if the two graphs are isomorphic: that is, is there a one-to-one mapping of $\pi : V \to V'$, so that for all pairs of nodes v, w in V, the edge $(v, w) \in E$ if and only if $(\pi(v), \pi(w)) \in E'$. For example, the two graph on the figure below are isomorphic with the numbering showing how to identify the vertices.

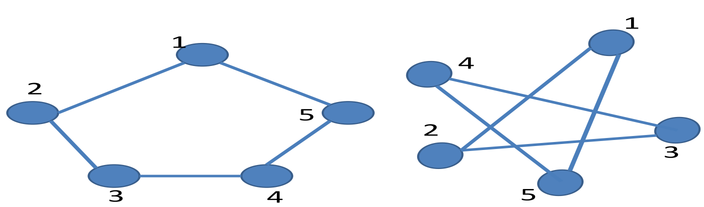

Figure 1: Two isomorphic graphs.

Give a polynomial-time reduction from GRAPH ISOMORPHISM to SAT, i.e., show that
Graph Isomorphism $\leq$ SAT.

Comment. You may find it interesting to hear about a recent breakthrough of a sub-exponential graph isomorphism algorithm `http://arxiv.org/abs/1512.03547`. While this is not a polynomial time algorithm, it suggests that graph isomorphism may really be easier than NP-complete.

(3) *(10 points)* Consider the (undirected) graph of friendships $G = (V, E)$, where nodes correspond to people, and two nodes are connected by an edge if they are friends (we will assume friendships are symmetric). You are also one node in the graph, let's call this node y. You would like to organize a giant party to subset of friend, i.e., a subset of the neighbors of your node in the graph. Ideally you want all pairs of people invited to be friends. Such a group of nodes with all edges between them is called a *clique* in the graph. It turns out that this would make your party too small. So you decide that it is OK to invite any pair of people, who have at least one friend in common beyond yourself, where this other shared friend $x \neq y$ is not necessarily one of your friends. For this problem, we will call such a set of nodes an *almost clique*, that is, a set $S \subset V$ all of which are neighbors of one node y (representing you in the graph), such that for all pairs of nodes $v, w \in S$, either $(v, w) \in E$ or there is a third node $s \neq y \in V$ (doesn't have to be in S or even be connected to y) such that both $(v, s) \in E$ and $(w, s) \in E$. Given a desired size k, and a graph G with special node y the ALMOST CLIQUE problem is to decide if the graph has an almost-clique among y's neighbors of size k.

Show that the ALMOST CLIQUE problem is NP-complete.

Hint: We have seen that the INDEPENDENT SET problem is NP-complete. A similar problem is the CLIQUE problem: given a graph G and a target size k, does the graph have a clique of size k? Notice that the CLIQUE problem is also NP-complete (as the clique problem on graph G is the same as the independent set problem on the complement graph of G, where the edges in the complement are exactly the pairs that are not edges in G). You can use that the CLIQUE problem is NP-complete without writing down a proof for this.