CS500, Theory of Computation
Homework #4

Note: You may discuss this homework with others in the class. However, you must do your own writeup, and you must clearly state on your homework who you worked with. Due by email in .pdf format by midnight on Wednesday, April 7th.

1. Let G be the undirected graph with 8 vertices and 12 edges formed by the edges of a cube. How many undirected Hamiltonian cycles does G have?

2. Show that Primality is in P if we give the input in unary rather than binary. In other words, show that $L = \{1^p : p \text{ is prime}\}$ is in P.

   Now consider Primality when the input is given in binary. Is it obviously in NP? Is its complement, Compositeness (i.e., the property of not being a prime)? In fact, both of these are in NP, and it was proved two years ago that they are in P; but I want you to tell me which one is *obviously* in NP.

3. Two graphs G and H are *isomorphic* if their structure is the same, except that their vertices are permuted: formally, if there is a 1-1 and onto mapping ϕ from the vertices of G to the vertices of H such that u and v are connected in G if and only if $\phi(u)$ and $\phi(v)$ are connected in H. Prove that the property Graph Isomorphism (that is, the language $\{\langle G, H\rangle : G \text{ and } H \text{ are isomorphic graphs}\}$) is in NP. What do you think about Graph Non-Isomorphism? Is it in NP?

4. Recall that the star operation on a language gives $L^* = \cup_{i\geq 0} L^i$ where L^i is the concatenation of L with itself i times. Show that P is closed under this operation: that is, if $L \in \mathrm{P}$, then $L^* \in \mathrm{P}$. The tricky part is that if I give you an input in L^*, you don't know where the boundaries between words in L are. Hint: build a table of which substrings of the input are in L, and then use dynamic programming.

5. Show that if $L \in \mathrm{NP}$, then $L^* \in \mathrm{NP}$. Hint: this is easier than the previous problem.

6. Consider two problems, Short Path and Long Path. Both take input (G, u, v, k) where G is a graph, u and v are vertices, and k is an integer. Short Path asks whether there is a path in G from u to v of length *at most* k, and Long Path asks whether there is a path of length *at least* k. In both cases paths are not allowed to visit the same vertex twice. Show that Short Path is in P, but Long Path is NP-complete.

7. Recall that Not-All-Equal Satisfiability (NAE-SAT) is like SAT, but where we demand that every clause contains at least one true literal and at least one false one. NAE-3-SAT is the special case where each clause has exactly 3 literals. Show that NAE-3-SAT is NP-complete by reducing 3-SAT to it. Make sure you prove that your reduction works: that is, that a satisfying assignment for the original 3-SAT formula exists if, and only if, a satisfying assignment exists for the resulting NAE-3-SAT formula.

   Hint 1: it might be easier to first reduce 3-SAT to NAE-k-SAT for some $k > 3$ and then show how to break up NAE-k-SAT clauses into NAE-3-SAT clauses.

   Hint 2: unlike for SAT, for NAESAT the complement of a satisfying assignment is also satisfying, and this makes it initially confusing how to reduce SAT to NAESAT. Try "breaking the symmetry" by using one or more additional variables as references to define which value will correspond to True or False in the reduction. You may not use constants like T or F.

8. Show that Monotone NAE-3-SAT, where no variables are negated, is NP-complete. Hint: use two variables in the new formula for each variable of the old. You may repeat variables in a clause.

9. Given a graph G with vertices V, a *cut* is a subset $S \subset V$. The size of the cut is the number of edges with one end in S and the other end in $\overline{S}$. MAX-CUT is the following problem: given a graph G and a number k, does G have a cut of size k or more? Show that MAX-CUT is NP-complete by reducing NAE-3-SAT to it.

   Hint: if the NAE-3-SAT formula has k clauses, represent each variable x with $3k$ vertices labelled x and another $3k$ vertices labelled $\overline{x}$, with $(3k)^2$ edges connecting all the x vertices to all the $\overline{x}$ vertices. Then represent each clause with a triangle connecting an appropriate triple of vertices. Figure out how large the cut corresponding to a solution to the NAE-3-SAT formula is, and prove that a cut of this size exists if and only if the formula is satisfiable.

10. Suppose I have a 3-SAT formula F. Define the *degree* of a variable x_i as the number of clauses it appears in (either positively or negatively). Show that the restricted version of 3-SAT in which every variable has degree at most 2 is in P. What degree do you think we need for NP-completeness?

11. Point out and discuss the fallacy in the following "proof" that P $\neq$ NP: "To see if a 3-SAT formula is satisfiable, we need to look at 2^n possible truth assignments. This takes exponential time, so 3-SAT is not in P. But it is in NP, so P $\neq$ NP."