Problem Set: Complexity Theory

Note: NP-Complete and NP-Hard are used as names of the sets of NP-complete and NP-hard problems in this document.

Problem 1. Prove that if a problem $A \in P$, then $A \in NP$.

Problem 2. Let $A \in P$ and $B \leq_p A$. Prove that $B \in P$.

Problem 3. Prove that " $\leq_p$ " is transitive, i.e if $A \leq_p B$ and $B \leq_p C$, then $A \leq_p C$.

Problem 4. Assuming $P \neq NP$, prove or give a counter example for the following statements.

1. NP-Complete $= NP$
2. NP-Complete $\cap P = \emptyset$
3. NP-Hard $= NP$

Problem 5. Let A be a NP-Complete problem and B and C are any other problems (may or may not be in NP). Suppose that B is polynomial time reducible to A and A is polynomial-time reducible to C. Prove whether or not the following statements are true.

1. C is NP-complete.
2. B is NP-Hard.
3. C is NP-Hard

Problem 6. Prove that if any NP-complete problem is polynomial-time solvable, then $P = NP$.

Problem 7. Prove that the clique problem is NP-complete. *Hint:* Show that 3SAT is polynomial time reducible to Clique problem.
The Clique Problem: Given a graph G, the clique problem asks to find the largest clique in G, (A clique of order k is a complete graph on k vertices).
Decision Version: Given a graph G and an integer k, is there a clique of size at least k in G?

Problem 8. Prove that Vertex Cover problem is polynomial time reducible to Dominating Set problem. *Hint:* Replace every edge (u, v) in G with a triangle (u, v, w) to form G', where $w \in G'$ and $w \notin G$ (see Figure 1).
The Vertex Cover Problem: Given a graph G and a number k, decision version of the vertex cover problem asks if there is a subset of size at most k in $V(G)$ that covers all edges (i.e. every edge in G intersects the set subset).
Dominating Set Problem Given a graph $G(V, E)$ and a number k, decision version of

Dominating set problem asks if there is a dominating set of size k in $V(G)$. Dominating set is a subset $A \subset V$ such that each vertex is either in A or has a neighbor in A.

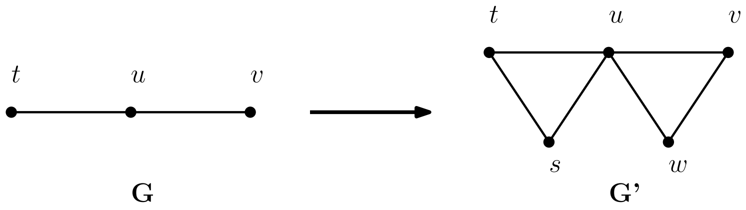

Figure 1: Vertex Cover input G transformed to Dominating Set input G'

Problem 9. Prove that 3-SAT problem is polynomial time reducible to 3-coloring problem.
k-Coloring Problem Given a graph G, is there a coloring of the nodes with k colors such that the endpoints of every edge are colored differently?
Hint: For every variable x_i. create two nodes $x_i, \overline{x_i}$ and connect them. Make three special nodes $\{Base, True, False\}$ and connect them to form triangle. Now connect every variable node to Base node, as shown in Figure

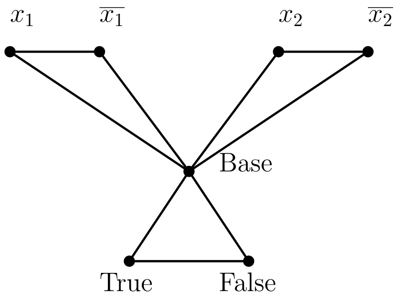

Figure 2: 3-SAT input transformed to 3-coloring input

Problem 10. Prove that Subset Sum problem is NP-complete.
Subset Sum Problem Given a set A of integers and an integer k, does there exist a subset of A such that the sum of its elements is equal to k?

Problem 11. Prove that Hamiltonian cycle problem is NP-complete.
Hamiltonian Cycle Problem Given a graph G on n vertices, is there a cycle on n vertices in the graph.

Problem 12. Prove that Hamiltonian Path problem is $NP-$Complete.
Hamiltonian Path Problem: Given a graph G, does G contain a path that visits every node exactly once?

Hint: Prove that Hamiltonian Cycle problem is polynomial time reducible to Hamiltonian Path problem. Pick any edge (u, v) in G, add two new vertices u_1, v_1 such that u_1 is only connected to u and v_1 is only connected to v.

Problem 13. Suppose we are given that the graph has no cycle. Design a polynomial time algorithm to find the longest $s-t$ path. *Hint:* You don't have to design an algorithm, just model is as a problem we already studied.

Longest $s-t$-path Problem: Given a weighted graph $G = (V, E)$ with $w : E \to R$ and two vertices $s \neq t \in V$, called the source and target vertex respectively, find a simple $s-t$ path P of maximum total weight, where weight of a path is the sum of weights of its edges, i.e. $w(P) = \sum_{e \in P} w(e)$.

The decision version of the longest $s-t$-path is given an integer k, is there a $s-t$ path of length at least k in G.