CS 8803 GA
Solutions for NP Practice Problems

Problem 1: [DPV] Problem 8.1

Solution: We want to show that TSP-OPT $\rightarrow$ TSP, i.e., we will design a polynomial time algorithm for solving TSP-OPT assuming that we have an algorithm $A_{\text{TSP}}(b)$ that solves TSP problem with a budget b in polynomial time. The algorithm we will use is the binary search algorithm. Consider an input to TSP-OPT, where D is the matrix of distances and b is the budget. (We assume the entries of D are integers for simplicity.) Let B be the total sum of the distance matrix D, i.e., $B = \sum_{i,j} D_{i,j}$. We will perform binary search in the range $[0, B]$. In the first round, the algorithm will call $A_{\text{TSP}}(B/2)$ and if the result is "YES" then we will try the range $[0, B/2]$ in the next round and recurse; if the result is "NO", then we will try the range $(B/2, B]$ and continue until the range is a single integer number. The total number of times that our algorithm calls A_{TSP} is at most $O(\log B)$. And note that the input size of the TSP-OPT problem is at least $\max\{\Omega(\log B), \log n\}$. Therefore, our algorithm's running time is polynomial in the input size.

Problem 2: [DPV] Problem 8.3

Solution: First we will show that it is in NP. Given a satisfying assignment σ of F, we can just check clause by clause that σ satisfies every one, which is a linear time algorithm. Then, we just count how many variables are set to be TRUE in σ and decide whether the number is larger than k or not.

We will show that SAT $\rightarrow$ STINGY-SAT. Consider an input formula f for SAT with n variables and m clauses. We run STINGY-SAT on f with $k = n$. Every assignment has at most n variables set to true, so the extra restriction that at most k variables are set to true does not constrain the set of solutions. Hence, if STINGY-SAT returns a satisfying assignment, then that assignment is also a satisfying assignment for the SAT problem. And if STINGY-SAT returns NO, there are no solutions, then there are no solutions to this SAT input and we return NO as well for the SAT problem.

Problem 3: [DPV] Problem 8.4

Solution (a): For graph $G = (V, E)$, given any set $S \subseteq V$, it is easy to verify whether S is a clique or not in $O(|V|^2)$ time and also count the size of S in linear time to decide whether $|S| < k$ or not.

Solution (b): The direction of the reduction is wrong. It is showing CLIQUE-3 $\rightarrow$ CLIQUE. In order to show CLIQUE-3 is NP-complete we need to do the reduction from a known NP-complete problem, say CLIQUE. So we need to show that CLIQUE $\rightarrow$ CLIQUE-3.

Solution (c): The statement of "a subset $C \subseteq V$ is a vertex cover in G if and only if the complementary set $V - C$ is a clique in the same graph G" is incorrect. The correct statement is "a subset $C \subseteq V$ is a vertex cover in G if and only if the complementary set $V - C$ is a clique in the complementary graph $\bar{G} = (V, V^2 - E)$".

Problem 4: [DPV] Problem 8.10 part(a)

Prove that the following problem is NP-complete by showing that it is a generalization of some NP-complete problem we have seen.
SUBGRAPH ISOMORPHISM: Given as input two undirected graphs G and H, determine whether G is a subgraph of H (that is, whether by deleting certain vertices and edges of H we obtain a graph that is, up to renaming of vertices, identical to G), and if so, return the corresponding mapping of $V(G)$ into $V(H)$.

Solution:

a) SUBGRAPH ISOMORPHISM is in NP because given the mapping π from $V(G)$ to $V(H)$, we can verify the solution in $O(|E| + |V|)$ time by checking for every edge $(u, v) \in E$ in G it maps to an edge $(\pi(u), \pi(v)) \in E(H)$ in H.

b) We'll prove that: INDEPENDENT SET $\rightarrow$ SUBGRAPH ISOMORPHISM.

Consider an input to the INDEPENDENT SET problem. Denote this input as the graph $H = (V, E)$ and an integer k. Now we'll construct the input to the SUBGRAPH ISOMORPHISM problem. Let $G = (V, E)$ be the graph where $|V'| = k$ and $E' = \emptyset$.

Run SUBGRAPH ISOMORPHISM on G and H. H has an independent set of size k if and only if there is a solution to the SUBGRAPH ISOMORPHISM problem. The solution provides a mapping from V' into V so we can also find the independent set in the original graph H and thus find a solution to the INDEPENDENT SET problem.