CS 344 HW 4

Fall 2019

1. (DPV 8.1) *Optimization versus search.* Recall the traveling salesman problem:

   TSP

   Input: A matrix of distances; a budget b

   Output: A tour which passes through all the cities and has length $\leq b$, if such a tour exists.

   The optimization version of this problem asks directly for the shortest tour.

   TSP-OPT

   Input: A matrix of distances

   Output: The shortest tour which passes through all the cities.

   Show that if TSP can be solved in polynomial time, then so can TSP-OPT.

2. (DPV 8.2) *Search versus decision.* Suppose you have a procedure which runs in polynomial time and tells you whether or not a graph has a Hamiltonian path. Show that you can use it to develop a polynomial-time algorithm to return the actual path, if it exists.

3. (DPV 8.17) Show that for any problem Π in NP, there is an algorithm which solves Π in time $O(2^{p(n)})$, where n is the size of the input instance and $p(n)$ is a polynomial (which may depend on Π).

4. (DPV 8.13) Determine which of the following problems are NP-complete and which are solvable in polynomial time. In each problem you are given an undirected graph $G = (V, E)$, along with:

   - A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves includes the set L.
   - A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves is precisely the set L.
   - A set of nodes $L \subseteq V$, and you must find a spanning tree such that its set of leaves included in the set L.
   - An integer k, and you must find a spanning tree with k or fewer leaves.
   - An integer k, and you must find a spanning tree with k or more leaves.
   - An integer k, and you must find a spanning tree with exactly k leaves.

   (Hint: All the NP-completeness proofs are by generalization, except for one.)

   (Generalization: To reduce $A \rightarrow B$, you may be able to note that A is a special case of B, so the f and h functions to transform from one to the other are trivial. For example, we can easily reduce 3-SAT $\rightarrow$ SAT, since a 3-SAT instance is already a valid SAT instance!)