Lab 14 Solutions

1. Show that TSP is NP-complete. (Hint: use the relationship between TSP and HamiltonianCycle discussed in the slides. You may assume that the HamiltonianCycle problem is NP-complete.)

   Solution:
   First show TSP is in NP.
   Suppose R is an NP problem. We must show that R $\xrightarrow{\text{poly}}$ TSP. Notice that
   $$R \xrightarrow{\text{poly}} HC \xrightarrow{\text{poly}} TSP$$
   The first is because HC is NP-complete; the second is shown in the lecture.

   So R $\xrightarrow{\text{poly}}$ TSP.

2. Below is another variation of the Knapsack problem.
   Given a set $S = \{s_0, s_1, \ldots, s_{n-1}\}$ of items, weights $\{w_0, w_1, \ldots, w_{n-1}\}$, values $\{v_0, v_1, \ldots, v_{n-1}\}$, a max weight W, and a min value V, find a subset T of S whose total value is no less than V and total weight is at most W.

   Show that the SubsetSum problem is polynomial reducible to this Knapsack problem.

   Solution:

   Let SS be a SubsetSum instance of size n consisting of the set $S = \{s_0, s_1, \ldots, s_{n-1}\}$ of positive integers and a non-negative integer k. We show how to transform S into a Knapsack instance KN in polynomial time. We must provide a set of items with weights and values, and maximum weight W and minimum value V.

   We define KN as follows: Let the set S from SS be the KN set of items. For each i, let $w_i = s_i$ and let $v_i = s_i$. Let W = k and let V = k. These definitions specify an Knapsack instance of size O(n). We must show that a solution to SS yields a solution to KN, and conversely.

   Verification: Solution to SS ⇒ Solution to KN

   Suppose T is a solution to SS (T is a subset of S whose sum is k). We show T is also a solution to KN. We must show
   $$\sum_{s_i \in T} w_i \le W \text{ and } \sum_{s_i \in T} v_i \ge V.$$

   Since T is a solution to SS, we know

$$\sum_{s_i \in T} s_i = k.$$

It follows that

$$\sum_{s_i \in T} w_i = \sum_{s_i \in T} s_i = k = W$$

and

$$\sum_{s_i \in T} v_i = \sum_{s_i \in T} s_i = k = V$$

as required. We have shown that a solution to SS yields a solution to KN.

Verification: Solution to KN ⇒ Solution to SS

Suppose T is a solution to KN (T is a subset of S the sum of whose weights is ≤ W and sum of whose values is ≥ V). We show T is also a solution to SS. We must show that

$$\sum_{s_i \in T} s_i = k.$$

Since T is a solution to KN we have

$$\sum_{s_i \in T} s_i = \sum_{s_i \in T} w_i \leq W = k$$

and also

$$\sum_{s_i \in T} s_i = \sum_{s_i \in T} v_i \geq V = k$$

which, together, establish the desired result. We have shown that a solution to KN yields a solution to SS.

3. Show that the worst case for VertexCoverApprox can happen by giving an example of a graph G which has these properties:
    a. G has a smallest vertex cover of size *s*
    b. VertexCoverApprox outputs size 2*s as its approximation to optimal size.

Solution:

Consider the following disconnected graph with 2 edges and 4 vertices:

A — B
C — D

The smallest vertex cover has size 2 – an example of such a vertex cover is {A, C}. However, the output of the VertexCoverApprox algorithm is {A, B, C, D}, a cover that has exactly twice ths size of a minimal cover.

4. Show that the VertexCover decision problem belongs to NP.

Solution. Assume a solution U of vertices is given as a solution to the VertexCover problem with input G = (V, E), k. To verify that U is correct, we must

(1) Show that $U \subseteq V$
(2) Show that U is a vertex cover – verify that each edge in E has an endpoint in U
(3) Show that U.size $\leq$ k

(1) requires O(n) (using hashtable for storing sets); (2) requires O(m) (where m is the number of edges). And (3) is (under reasonable assumptions) O(1). Therefore time to check correctness is $O(n^2)$ and hence VertexCover is in NP.