COMP3721 Homework 4 Solutions

1 1.1 True

1.2 Unknown

1.3 False. Some NP-hard problems are more difficult than NP-complete problems.

1.4 True

2 We can prove it by contradiction. Assume $P = NP$, so for each NP language L, there is a polynomial time deterministic TM M decide it. Then $\bar{L}$ can also be decided in polynomial time, since we can modify M by changing the *yes* (*no*) halting states to *no* (*yes*) halting states, and the new TM will decide $\bar{L}$. This means $P = NP = coNP$, which is a contradiction.

3 It is easy to prove Knapsack is in NP. The certificate is a set of objects. One need to verify the total weight of the objects is no larger than the capacity of the knapsack and the total value is no less than the given value. This trivially can be done in polynomial time.

We show how to reduce the SP problem to Knapsack. Given any instance of SP, which is a set S of integers. We set $T = \sum_{x \in S} x$, if it is odd, we can construct any no instance for $Knapsack$, since S is a no instance for SP. If T is even, we construct a Knapsack instance as follows. For each $x \in S$, there is an object with weight and value both x, and the capacity of the knapsack is $T/2$ and $k = T/2$. We next prove S is a yes instance iff the constructed instance for Knapsack is a yes instance.

For the only if direction, if the constructed instance is a yes instance for Knapsack, i.e., the total weight of the selected objects and their total value are both equal to $T/2$, it is easy to see that it implies the set S is a yes instance for SP problem.

For the if direction, if S is a yes instance for SP, i.e. there is a subset S' of S such that $\sum_{x \in S'} x = T/2$, then in the Knapsack problem, we just select the objects corresponding to elements in S'. The total weight and value of the selected objects are both $T/2$ which is a yes instance for the Knapsack problem.

4 (a) $DDS \in NP$

The certificate C is a set of k vertices that dominates all other vertices. Checking that every other vertex is dominated by at least one vertex in C can be easily done in polynomial time.

(b) $DSC \leq_p DDS$

Given an input to a Decision Set Cover Problem: $X = \{x_1, x_2, \cdots, x_m\}$, and a collection of sets $F = \{S_1, \cdots, S_n\}$, where $S_i \subseteq X$, we first check if every x_i is contained in at least one of the S_i's. If not, then we construct any no instance for the DDS problem. Otherwise, we construct an input (G, k') to the DDS problem so that there exists a collection of k sets in F that cover X iff G has a dominating set of size k'.

Transformation: We create a vertex v_i for each x_i, and create a vertex u_i for each S_i. For each $S_i \in F$, and for each $x_j \in S_i$, we create an edge (u_i, v_j). For every two vertices u_i, u_j, create two edges (u_i, u_j) and (u_j, u_i). Set $k' = k$. Obviously the transformation could be done in polynomial time.

Proof: Suppose the DSC input is a yes-input, i.e., there exists a collection of k sets in F that cover X. Then we can pick the corresponding vertices u_i in G. This will dominate all other vertices. Thus DDS is also "yes".

Suppose G has a dominating set D of size k. If D contains any vertex v_i, because it has no outgoing edges and has at least one incoming edge, we can replace it with any vertex that points to v_i without leaving any vertex undominated. So we can transform D into D' with the same number of vertices. Then we pick the corresponding sets, which will cover the entire X.