Exercise Solution: NP-Completeness

1. Let DOUBLE-SAT $= \{\langle\phi\rangle \mid \phi$ has at least two satisfying assignments$\}$.

   Show that DOUBLE-SAT is NP-complete.

   We first show that DOUBLE-SAT is in NP. For any yes-instance ϕ of DOUBLE-SAT, there exists an certificate, which is an satisfying assignment to every variable. The assignment needs at most $O(n)$ bits to encode, where n is the number of variables. With this certificate, a verifier can use at most $O(n)$ rounds to scan through the Boolean formula ϕ and replace the truth values of the literals in ϕ. Then, it only takes linear time to scan through the formula and check if it is true. Therefore, it takes polynomial time to verify the yes-instances.

   Next, we show that DOUBLE-SAT is NP-hard by reducing from SAT in polynomial time.

   For any instance ϕ of SAT, we construct an instance of DOUBLE-SAT $\phi' = \phi \wedge (d \vee \bar{d})$, where d is a dummy variable. The new instance size is polynomial in the size of ϕ since there are only one new variable and one new clause. Therefore, the construction can be done in polynomial time in the instance size ϕ.

   Now, we show that ϕ' is satisfiable if and only if ϕ is satisfiable. First, assume that ϕ is satisfiable. We can construct two satisfying assignments A to ϕ':

   - $x_1, x_2, \cdots, x_n$ have the same truth value in A and $d =$ TRUE, and
   - $x_1, x_2, \cdots, x_n$ have the same truth value in A and $d =$ FALSE.

   Therefore, ϕ' is a yes-instance of DOUBLE-SAT.

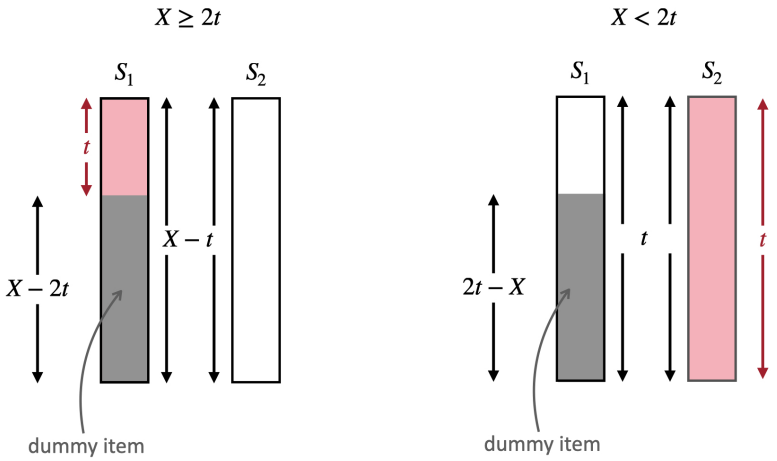
   Next, assume that ϕ' is satisfiable. That is, there exist at least two satisfying assignments to variables $x_1, x_2, \cdots, x_n, d$, where x_i's are the variables in ϕ. Since the only extra variable d only appears in the extra clause $(d \vee \bar{d})$, the truth values of x_i's in one of the satisfying assignments satisfy the Boolean formula ϕ. Therefore, ϕ is a yes-instance.

2. Show that PARTITION is NP-complete. (Hint: By reduction from SUBSETSUM.)

   SUBSETSUM $= \{\langle S, t\rangle \mid$ There is a subset $T \subseteq S = \{x_1, \cdots, x_n\}$ such that the sum of values in T is equal to $t\}$.

   PARTITION $= \{\langle S\rangle \mid S = \{x_1, \cdots, x_n\}$ can be partitioned into S_1 and S_2 such that the sum of elements in S_1 is equal to the sum of elements in $S_2\}$.

   For reducing SUBSETSUM to PARTITION, we need to transform every input of SUBSETSUM, $\langle S, t\rangle$, to an input of PARTITION, $\langle S'\rangle$. We do this by adding a dummy item d to S' such that there is a subset in S with sum t if and only if $S' = S \cup \{d\}$ can be partitioned into two subsets with equal sums. See the following figure.



To determine if $\langle S, t\rangle$ is in the language SUBSETSUM, let X be the sum of values in S (that is, $X = \sum_i x_i$). We construct S' by adding an extra number $|X - 2t|$ to S. That is, $S' = S \cup \{|X - 2t|\}$. The construction of S' can be done in polynomial time.

There are two cases of the value $X - 2t$, $X \geq 2t$ or $X < 2t$. Recall that X is the sum of values in S. If $X \geq 2t$, the sum of values in S' is $X + (X - 2t) = 2X - 2t$. If $X < 2t$, the sum of values in S' is $X + (2t - X) = 2t$. In the following we show that there is a subset in S with sum t if and only if $S' = S \cup \{d\}$ can be partitioned for each case.

Suppose that $X \geq 2t$. The sum of S' is $2X - 2t$. If there is a subset $C \subseteq S$ with sum t, there is a partition $C \cup \{|X - 2t|\}$ with sum $X - t$. Hence, $C \cup \{|X - 2t|\}$ and $S' \setminus (C \cup \{|X - 2t|\})$ form a partition in S'. For the other direction, suppose that there is a partition of S', S'_1 and S'_2. Without loss of generality, we assume that $|X - 2t| \in S'_1$. Since S'_1 and S'_2 form a partition of S', the sum of S'_1 is $X - t$ since $X \geq 2t$. The sum of the elements in S'_1 excepts $|X - 2t|$ is $(X - t) - (X - 2t) = t$ and the subset $S'_1 \setminus |X - 2t|$ is a subset of S with sum t.

Suppose that $2t \geq X$. The sum of S' is $X + (2t - X) = 2t$. If there is a subset $C \subseteq S$ with sum t, C and $S' \setminus C$ form a partition in S'. For the other direction, suppose that there is a partition of S', S'_1 and S'_2. Without loss of generality, we assume that $|X - 2t| \in S'_1$. Since S'_1 and S'_2 form a partition of S', the sum of S'_2 is t. Since $|X - 2t| \in S'_1$ (and not in S'_2), S'_2 is a subset of S with sum t.

3. The language 2WAYPARTITION is defined as $\{\langle S, t\rangle \mid S = \{x_1, x_2, \cdots, x_n\}$ where exists a subset $T \subset S$ such that $\sum_{y_i \in T} y_i = \sum_{y_i \in S \setminus T} y_i$ and $|T| \leq t$ and $|S \setminus T| \leq t\}$. Show that 2WAYPARTITION is NP-hard.

   We show the NP-hardness of 2WAYPARTITION by reduction from PARTITION. For any instance of PARTITION, set S, we construct an instance of 2WAYPARTITION, set $S' = S$, and $t = |S'|$. This instance transformation can be done in polynomial time.

   For any yes-instance (S', t) of 2WAYPARTITION, there exists a subset T of S' such that the sum of elements in T is equal to the sum of elements in $S' \setminus T$, $|T| \leq t$, and $|S' \setminus T| \leq t$. The partition T is also a partition in S. Therefore, the corresponding instance S is a yes-instance of PARTITION.

   For any yes-instance S of PARTITION, there exists an equal-sum partition of S. Moreover, each of the two parts T and $S \setminus T$ has cardinality at most $|S| = |S'| = t$. Therefore, the corresponding instance (S', t) is a yes-instance of 2WAYPARTITION.

4. Given a graph $G = (V, E)$, an *independent set* is a subset U of vertices in V such that there is no edge between any two vertices in U. In the *Maximum Independent Set* problem, we aim to find the maximum independent set in the given graph.

(a) Give the decision version of the Maximum Independent Set, IndepSet

IndepSet = Given a graph G, is there an independent set in G with size at least k?
(Alternative: IndepSet = $\{\langle G, k\rangle \mid$ There is an independent set in G with size at least $k\}$)

(b) Show that the decision version of the Maximum Independent Set is NP-complete.

First, we prove that IndepSet is in NP. Let an independent set of size k, c, be a certificate.

A = "On input $\langle\langle G, k\rangle, c\rangle$:
1. Test whether c is a set of k nodes in G.
2. Test whether G there is no edge between any pair of vertices in c
3. If both 1 and 2 pass, *accept*; otherwise, *reject*."

Step 1 takes at most $|c| = k$ times of scanning through the input. Step 2 takes at most $|c|^2$ times of scanning through the input. Hence, A runs in polynomial time in the input length.

Now, we prove that IndepSet is NP-hard by polynomial-time reduction from Clique. Given an instance $G = (V, E)$ and k of Clique problem, we construct a graph $G' = (V', E')$ and an integer k' for IndepSet as follows. We set $V' = V$ and $k' = k$. That is, For any two vertices u and v, we have $(u, v) \in E'$ if and only if $(u, v) \notin E$. The construction can be done in polynomial time.

Now we show that there is a clique in G with size at least k if and only if there is an independent set with size at least k' in G'. Assume that W' is a subset of V' that is an independent set with size at least k. By the construction, any two vertices in W' are adjacent in G (since they are not adjacent in G'). Thus, the vertices in W' are a clique of G with size at least k.

Assume that W is a subset of V that is a clique with size at least k. By the construction, any two vertices in W are not adjacent in G' (since they are not adjacent in G). Thus, the vertices in W are an independent set of G with size at least $k = k'$.

5. Given a graph $G = (V, E)$, a *vertex cover* is a subset C of vertices in V such that for any edge $(u, v) \in E$, $\{u, v\} \cap U \geq 1$. In the *Minimum Vertex Cover* problem, we aim at finding the minimum vertex cover in the given graph.

(a) Give the decision version of the Minimum Vertex Cover, VC

VC = Given a graph G, is there a vertex cover in G with size at most k?
(Alternative: VC = $\{\langle G, k\rangle \mid$ There is a vertex cover in G with size at most $k\}$)

(b) Show that the decision version of the Minimum Vertex Cover, VC, is NP-complete.

Proof. First, we prove that VC is in NP. Let a vertex cover C of size k be the certificate. The verifier first checks if C has less than or equal to k vertices. Next, the verifier checks for every edge to see if it has at least one endpoint in C. The verification takes linear time ($O(\min\{|C|, |V|\})$) for checking the size of C. For checking if the elements in C are in V, it takes $O(|V|^2)$ time. The final checking takes $O(|E| \cdot |D|)$ time. Hence, the verification can be done in polynomial time in the input length of $G = (V, E)$.

Next, we show that VC is NP-hard by Polynomial time reduction from IndepSet. Given an instance of the IndepSet problem, $G = (V, E)$ and integer k, we construct an instance of the VC, G' and integer k' where $G' = G$ and $k' = |V| - k$. The reduction takes polynomial time since G' is a copy of G, and k' can be calculated in constant time.

Now, we want to show that the reduction works by proving that there is an independent set in G of size at least k if and only if there is a vertex cover in G' of size at most k'.

Suppose that G has an independent set $I \subseteq V$ with $|I| \geq k$. We claim that $V \setminus I$ is a vertex cover in G'. Since I is an independent set in G, for any edge $(u, v) \in E$, the endpoints u

and v cannot be both in I. That is, one of u or v is in $V \setminus I$. Hence, the set of vertices $V \setminus I$ is a vertex cover, which has size $|V| - |I| \leq |V| - k$.

Suppose that there is a vertex cover C in G' where $|C| \leq |V| - k$. For all edge $(u, v) \in E'$, at least one of u or v is in C. That is, for any pair of u and v that are not in C, $(u, v) \notin C$. Therefore, the set $V \setminus C$ forms an independent set, which has size $|V| - |C| \geq k$. □

6. The weighted vertex cover problem is defined as follows. Given a graph $G = (V, E)$, where each vertex $v \in V$ has a positive weight w_v, find a subset of V with minimum total weight such that this subset forms a vertex cover of G. Show that the weighted vertex cover problem is NP-complete.

   (Note that you need to provide formal proof; saying "since its special case vertex cover is NP-hard, it is NP-hard" is insufficient.)