Problem 1
The bin packing problem is as follows: given a set of $n$ items, each with a weight, $w_i$, and a set of bins, $B$, each with a fixed capacity, $c$, find the minimum number of bins needed to pack all the items.

Prove that bin packing is NP-Hard, but not NP-Complete.

Solution:

First things first, we need to rephrase bin packing as a decision problem. Let BIN-PACK be defined as:

$$\text{BIN-PACK} = \{\langle W, c, b \rangle \mid \text{there is a minimum of } b \text{ disjoint subsets } S_x \text{ such that } S_1 \cup S_2 \cup ... \cup S_b = W \text{ and for each } S_x, \text{ and all } \langle i, w_i \rangle \in S_x, \sum w_i \leq c\}$$

Where each $i$ is an item and $w_i$ is its weight.

Next we can attempt to show that BIN-PACK is in NP. The following algorithm accepts if $\langle W, c, b \rangle \in \text{BinPack}$:

1. Non-deterministically select $b$ disjoint $S$.
2. Check that, for each $S_x$, and all $\langle i, w_i \rangle \in S_x, \sum w_i \leq c$.
3. Use the same procedure to determine if $W$ can be packed into $b - 1$ bins of size $c$. If so, REJECT. If not, ACCEPT.

The first thing to note is that if we removed the word “minimal” from the language requirement, we could eliminate Step 3 from the above procedure and have a valid non-deterministic polynomial decider that leverages a valid deterministic polynomial verifier. The resulting NP language would be:

$$\text{BIN-PACK-NP} = \{\langle W, c, b \rangle \mid \text{there are } b \text{ disjoint subsets } S_x \text{ such that } S_1 \cup S_2 \cup ... \cup S_b = W \text{ and for each } S_x, \text{ and all } \langle i, w_i \rangle \in S_x, \sum w_i \leq c\}$$

The second thing to note is that, in class, we learned that the complement of a problem in NP, coNP, is not necessarily in NP. That is to say, the “negative space” of an NP language is harder to decide than the positive space.

The issue with BIN-PACK is that, by including that “minimal” requirement, we’ve snuck a negative space question into a seemingly positive space language. We must decide that $\langle W, c, b - 1 \rangle$ is NOT in BIN-PACK-NP in order to decide that $\langle W, c, b \rangle$ is in BIN-PACK. Thus, BIN-PACK is NOT in NP.
Next, to show that BIN-PACK is NP-HARD, we would like to do a reduction from SUBSET-SUM, aka SUBSET-SUM \( \leq_P \) BIN-PACK. However, this reduction procedure is not immediately obvious. Instead, we’re going to insert an intermediary step SUBSET-SUM \( \leq_P \) PARTITION \( \leq_P \) BIN-PACK.

** For reference, SUBSET-SUM is defined as follows:

** SUBSET-SUM = \( \{ \langle S, t \rangle \mid S = \{x_1, x_2, ..., x_k\}, \text{ and for some } \{y_1, y_2, ..., y_l\} \subseteq \{x_1, x_2, ..., x_k\}, \sum y_i = t \} \)

We’ll begin by defining PARTITION and showing that PARTITION \( \leq_P \) BIN-PACK.

PARTITION = \( \{ \langle A \rangle \mid \text{there are disjoint subsets } B, C, \text{ such that } B \cup C = A \text{ and } \sum_{b \in B} b = \sum_{c \in C} c \} \)

We will construct \( \langle W, c, b \rangle \) from \( \langle A \rangle \) as follows:

Enumerate each element \( a_i \) of \( A \) and pair each element with its corresponding integer to create \( W \):

\[
W = \{\langle 1, a_1 \rangle, \langle 2, a_2 \rangle, ..., \langle k, a_k \rangle\}
\]

If we again take \( n = \sum_{a \in A} a_i \), we can set \( c = \lfloor \frac{n}{2} \rfloor \), and \( b = 2 \).

Since this procedure simply expands \( \langle A \rangle \) with additional information, if runs in \( O(n^2) \) time.

** If \( \langle A \rangle \in \text{PARTITION} \), then \( \langle W, c, b \rangle \in \text{BIN-PACK} \): If \( \langle A \rangle \) is in PARTITION, then there is a way to partition \( A \) into two disjoint sets \( B \) and \( C \) such that \( B \cup C = A \) and \( \sum_{b \in B} b = \sum_{c \in C} c = \frac{n}{2} \).

So, given the values as weights, there is a way to pack them into two bins of size \( \lfloor \frac{n}{2} \rfloor \). Therefore \( \langle W, c, c \rangle \in \text{BIN-PACK} \).

** If \( \langle A \rangle \notin \text{PARTITION} \), then \( \langle W, c, b \rangle \notin \text{BIN-PACK} \): If \( \langle A \rangle \notin \text{PARTITION} \), then there is no way to partition \( A \) into two disjoint sets \( B \) and \( C \) such that \( B \cup C = A \) and \( \sum_{b \in B} b = \sum_{c \in C} c = \frac{n}{2} \).

So, given a bin size of \( \lfloor \frac{n}{2} \rfloor \), the weights of any \( X \subseteq W \) will sum to \( p \) such that \( p > \lfloor \frac{n}{2} \rfloor \) or \( p < \lfloor \frac{n}{2} \rfloor \). If \( p > \lfloor \frac{n}{2} \rfloor \), then \( X \) cannot be packed into a single bin, and if \( p < \lfloor \frac{n}{2} \rfloor \) then \( W - X \) cannot be packed into a single bin. Therefore \( \langle W, c, c \rangle \notin \text{BIN-PACK} \).
Now we can move on to the second part of our reduction: \( \text{SUBSET-SUM} \leq_p \text{PARTITION} \).

\( \text{PARTITION} = \{ (A) \mid \text{there are disjoint subsets } B, C, \text{ such that } B \cup C = A \text{ and } \sum_{b \in B} b = \sum_{c \in C} c \} \)

We begin by pointing out that the \( \text{SUBSET-SUM} \) problem \( \langle S, t \rangle \) is equivalent to the \( \text{SUBSET-SUM} \) problem \( \langle S, n - t \rangle \), where \( n \) is the sum of all members of \( S \). We will thus assume that \( t \geq \frac{n}{2} \), because if it were not, we could instead solve the problem \( \langle S, n - t \rangle \), where \( n - t \geq \frac{n}{2} \).

We will construct \( \langle A \rangle \) from \( \langle S, t \rangle \) as follows. Again, let \( n \) be the sum of all members of \( S \). We will construct \( A \) as \( A = S \cup \{ |n - 2t| \} \). This reduction can clearly be accomplished in \textbf{polynomial time}.

\textbf{If} \( \langle S, t \rangle \in \text{SUBSET-SUM}, \text{ then } \langle A \rangle \in \text{PARTITION} \): If \( \langle S, t \rangle \in \text{SUBSET-SUM} \), then there is a subset \( x \) whose elements add up to \( t \) (\( t \geq \frac{n}{2} \)) and a disjoint subset \( y \) whose elements add up to \( |n - t| \) (\( n - t \leq \frac{n}{2} \)). The elements of \( y \cup \{ |n - 2t| \} \) add up to \( t \). Therefore, \( A \) can be partitioned into two equal parts that sum to \( t \).

\textbf{If} \( \langle S, t \rangle \notin \text{SUBSET-SUM}, \text{ then } \langle A \rangle \notin \text{PARTITION} \): If \( \langle S, t \rangle \notin \text{SUBSET-SUM} \), then there is no subset of \( S \) whose elements add up to \( t \). Regardless, the elements of \( S \cup \{ |n - 2t| \} \) sum to \( 2t \). Since there is no subset of \( S \) that sums to \( t \), and \( \{ |n - 2t| \} \) can only be included on one side of the partition, there is no partition of \( S \cup \{ |n - 2t| \} \) such that the elements on both sides sum to \( t \).

Since \( \text{SUBSET-SUM} \leq_p \text{PARTITION} \leq_p \text{BIN-PACK} \), \( \text{BIN-PACK} \) is \textbf{NP-Hard}. 