1. Show that 
\[ f(n) \text{ is } \Theta(g(n)) \]
implies 
\[ g(n) \text{ is } \Theta(f(n)). \]

2. Suppose we have a computational problem X and two algorithms for X, A and B. Suppose that the complexities of A and B are \( f(n) \) and \( g(n) \) respectively. Given each of the following pieces of information, which algorithm is more efficient (in the sense of its complexity having a lower rate of growth), and what is the strongest conclusion we can reach (in terms of \( O \) and \( \Theta \)) about the problem’s complexity?

(a) \( f(n) \) is \( O(g(n)) \) but \( f(n) \) is not \( \Theta(g(n)) \).
(b) \( f(n) \) is \( \Theta(n^2) \) and \( g(n) \) is \( O(f(n)) \).
(c) \( f(n) \) is \( O(g(n)) \) and the complexity of the problem is \( \Theta(g(n)) \).

3. Show that if \( b \in \mathbb{N} \) and \( t(n) \geq n, t(n) * b^{t(n)} \) is \( 2^{O(t(n))} \).

(The above is used in the proof of a theorem in Lecture 22.)

**Note:** A function \( f(n) \) is \( 2^{O(t(n))} \) if there exist \( c > 0 \) and a positive integer \( M \) s.t. \( f(n) \leq 2^{ct(n)} \) for all \( n \geq M \).

4. Let SUBSET\_SUM be the problem of deciding if, given a collection of integers \( x_1, \ldots, x_n \) and a target number \( t \), there exists a subcollection \( \{y_1, \ldots, y_l\} \subseteq \{x_1, \ldots, x_n\} \) that adds up to \( t \):

\[ \sum_{i \in \{1, \ldots, l\}} y_i = t \]

Show that SUBSET\_SUM is in NP.