Problem 1. Implications of polynomial-time reductions. Remember that Problem \( Y \) is polynomial-time reducible to Problem \( X \) if there is an algorithm for solving Problem \( Y \) that looks like this:

```java
solveY(yInput)
  Construct xInput
  \( \triangleright \) polynomial time
  foo = solveX(xInput)
  \( \triangleright \) polynomial number of calls
  return yes/no based on foo
  \( \triangleright \) polynomial number of time
```

This means we can solve any instance of Problem \( Y \) using a black-box solver for Problem \( X \) and at most a polynomial amount of additional work.

Last week we showed that \textsc{Bipartite-Matching} is polynomial-time reducible to \textsc{Network-Flow} (we would write this as \textsc{Bipartite-Matching} \( \leq_p \text{ Network-Flow} \)). We know that there is a polynomial-time algorithm for \textsc{Network-Flow} (an efficient variant the Ford-Fulkerson algorithm). What does this imply about \textsc{Bipartite-Matching}?

(a) There is a polynomial-time algorithm for \textsc{Bipartite-Matching}.
(b) There is no polynomial-time algorithm for \textsc{Bipartite-Matching}.
(c) Nothing

Now consider a different pair of problems: \textsc{Multipartite-Batching} and \textsc{Fletwork-Know}, where, like their counterparts above, \textsc{Multipartite-Batching} is polynomial-time reducible to \textsc{Fletwork-Know} (that is, \textsc{Multipartite-Batching} \( \leq_p \text{ Fletwork-Know} \)). Suppose now that you prove that there is \textit{no} polynomial-time algorithm for \textsc{Multipartite-Batching}. What does this imply about \textsc{Fletwork-Know}?

(a) There is a polynomial-time algorithm for \textsc{Fletwork-Know}.
(b) There is no polynomial-time algorithm for \textsc{Fletwork-Know}.
(c) Nothing

Explain your answer.

Problem 2. Interval Scheduling. K&T Chapter 8, Exercise 1. For each of the questions below, decide whether the answer is (i) “Yes”, (ii) “Unlikely, because it would show that an NP-complete problem can be solved in polynomial time, which would prove that \( P = NP \)”. Explain your answer.

Let’s define the decision version of the Interval Scheduling Problem from Chapter 4 as follows: Given a collection of intervals on a time-line, and a bound \( k \), does the collection contain a subset of nonoverlapping intervals of size at least \( k \)?

(Hint: you may use the fact that Vertex Cover and Independent Set are NP-complete. Also, recall that reductions are transitive: if \( Y \leq_p X \) and \( X \leq_p U \), then \( Y \leq_p U \).)

1. Question: Is it the case that Interval Scheduling \( \leq_p \) Independent Set?
2. Question: Is it the case that Interval Scheduling \( \leq_p \) Vertex Cover?
3. Question: Is it the case that Independent Set \( \leq_p \) Interval Scheduling?
Problem 3. Diverse Subset. K&T Chapter 8, Exercise 2. A store trying to analyze the behavior of its customers will often maintain a two-dimensional array $A$, where the rows correspond to its customers and the columns correspond to the products it sells. The entry $A[i, j]$ specifies the quantity of product $j$ that has been purchased by customer $i$.

Here’s a tiny example of such an array $A$.

<table>
<thead>
<tr>
<th></th>
<th>detergent</th>
<th>beer</th>
<th>diapers</th>
<th>cat litter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raj</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Alanis</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Chelsea</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
</tbody>
</table>

One thing that a store might want to do with this data is the following. Let us say that a subset $S$ of the customers is diverse if no two of the of the customers in $S$ have ever bought the same product (i.e., for each product, at most one of the customers in S has ever bought it). A diverse set of customers can be useful, for example, as a target pool for market research.

We can now define the Diverse-Subset Problem as follows: Given an $m \times n$ array $A$ as defined above, and a number $k \leq m$, is there a subset of at least $k$ of customers that is diverse?

Show that Independent Set $\leq_P$ Diverse-Subset (read: Independent Set is polynomial-time reducible to Diverse-Subset).

(Since Independent-Set is NP-complete, this can be used to show that Diverse-Subset is also NP-complete.)