Exercise 1: The class \( \mathcal{NPC} \) \( (8 \text{ Points}) \)

As this type of exercise is really important for the course we have another one.

A subset of the nodes of a graph \( G \) is a **dominating set** if every other node of \( G \) is adjacent to some node in the subset. Let

\[
\text{DOMINATINGSet} = \{ \langle G, k \rangle \mid \text{has a dominating set with } k \text{ nodes} \}.
\]

Show that DOMINATINGSet is in \( \mathcal{NPC} \). Use that

\[
\text{VERTEXCover} := \{ \langle G, k \rangle \mid \text{Graph } G \text{ has a vertex cover of size at most } k \} \in \mathcal{NPC}.
\]

*Remark: A VERTEXCover is a subset \( V' \subseteq V \) of nodes of \( G = (V, E) \) such that every edge of \( G \) is adjacent to a node in the subset.*

Exercise 2: Complexity Classes: Big Picture \( (2+3+2 \text{ Points}) \)

(a) Why is \( \mathcal{P} \subseteq \mathcal{NP} \)?

(b) Show that \( \mathcal{P} \cap \mathcal{NPC} = \emptyset \) if \( \mathcal{P} \neq \mathcal{NPC} \).

*Hint: Assume that there exists a \( L \in \mathcal{P} \cap \mathcal{NPC} \) and derive a contradiction to \( \mathcal{P} \neq \mathcal{NPC} \).*

(c) Give a Venn Diagram showing the sets \( \mathcal{P}, \mathcal{NP}, \mathcal{NPC} \) for both cases \( \mathcal{P} \neq \mathcal{NP} \) and \( \mathcal{P} = \mathcal{NP} \).

*Remark: Use the results of (a) and (b) even if you did not succeed in proving those.*
Exercise 3: The class \( \mathcal{P} \)  

\textit{(1+2+2+1 Points)}

\textbf{Clique:}

- A \textit{clique} of a graph \( G = (V, E) \) is a subset \( Q \subseteq V \) such that for all \( u, v \in Q \) : \( \{u, v\} \in E \).

- **Input**: Encoding \( \langle G, k \rangle \) of an undirected, unweighted, simple graph \( G = (V, E) \) and \( k \in \mathbb{N} \).

- **Question**: Is there a clique of size at least \( k \)?

\( \mathcal{P} \) is the set of languages which can be decided by an algorithm whose runtime can be bounded by \( p(n) \), where \( p \) is a polynomial and \( n \) the size of the respective input (problem instance). Show that the following languages (\( \cong \) problems) are in the class \( \mathcal{P} \). Since it is typically easy (i.e. feasible in polynomial time) to decide whether an input is well-formed, your algorithm only needs to consider well-formed inputs. Use the \( O \)-notation to bound the run-time of your algorithm.

(a) 1-DominatingSet

(b) 2-VertexColoring

(c) 3-Clique

(d) Any context-free language \( L \). \hspace{1cm} \textit{Hint: You can use results from previous exercise sheets.}