Exercises on Complexity Theory

Exercise. Show that if there is an efficient algorithm for the optimisation problem Π then there is an efficient algorithm for the decision problem of Π.

Answer. Solve the optimization problem and check whether the optimal solution cost is less than or equal to K.

Exercise. Suppose we have an efficient algorithm ALG for the decision problem of an optimisation problem Π. Show that we can use ALG to compute an optimal solution for Π. What are the conditions for this algorithm to be efficient?

Answer. Let I be an instance of Π of size n. We assume that the cost function is integer and non-negative. Suppose we knew that the cost of an optimal solution is in the interval $[0, L]$ for some integer L. We can then do binary search on the interval $[0, L]$ and invoke ALG for each query value K to check whether there exists a solution whose cost is at most K. This way we will find the optimal value after $O(\log L)$ steps. The resulting algorithm is efficient if $O(\log L)$ is polynomially bounded in the size of I, e.g., if $L = O(e^{n^k})$ for some constant k.

Exercise. Try to identify a problem for which you believe that it is not in NP.

Answer. Consider, for example, the decision problem of whether a given graph $G = (V, E)$ is *not* Hamiltonian. Every certificate for a yes-instance would have to prove that every cycle of G does not contain all vertices. Such a certificate can be given by listing all possible cycles and showing that each one of them does not contain all vertices of G. However, this certificate cannot be verified in polynomial time. In fact, no such certificate has been found until now.

Exercise. Show that 0/1-ILP is NP-complete. Reduce SAT to this problem.

Answer. Formulate each clause of the SAT-formula F as a constraint, e.g., $(x_1 \vee \neg x_2 \vee x_3)$ is encoded as $x_1 + (1 - x_2) + x_3 \geq 1$. This transformation can be done in polynomial time. A 0/1-assignment x satisfies F iff x is a feasible solution for the corresponding 0/1-ILP.

Exercise. Show that ILP is NP-complete. *Hint: This is trivial.*

Answer. 0/1-ILP is a special case of ILP-DECISION.

Exercise. Show that the following problem is NP-complete:
LONGEST PATH:
Instance: An undirected graph $G = (V, E)$ and a parameter K.
Goal: Determine whether there exists a (simple) path in G of length at least K.

Answer. We first show that the so-called HAMILTONIAN PATH problem is NP-complete: Given an undirected graph $G = (V, E)$, determine whether G contains a Hamiltonian path, i.e., a path that visits every vertex exactly once. It is easy to verify that HAMILTONIAN PATH is in NP. We show that $HamiltonianCycle \propto HamiltonianPath$. Given an instance $G = (V, E)$ of HAMILTONIAN CYCLE, we proceed as follows: For every edge $e = \{u, v\} \in E$ of G, construct an augmented graph G_e that consists of G and two additional vertices u', v' that are connected to u and v, respectively, i.e., $G_e = (V \cup \{u', v'\}, E \cup \{u', u\} \cup \{v', v\})$. Note that the following holds for every edge $e \in E$: There is a Hamiltonian cycle in G using edge e if and only if there is a Hamiltonian path in G_e. Thus, there is a Hamiltonian cycle in G if and only if there is a Hamiltonian path in G_e for some edge $e \in E$. Note that this reduction takes polynomial time.

Now it is easy to conclude that LONGEST PATH is NP-complete because it is in NP and $HamiltonianPath \propto LongestPath$ simply by observing that there is a Hamiltonian path in G if and only if there is a path of length $n - 1$.

[PS] 15.12. We first need to argue that FEEDBACK VERTEX SET is in NP. A certificate of a yes-instances consists of a subset $V' \subseteq V$ of vertices such that by removing all vertices in V' the induced subgraph $G[V \setminus V']$ is acyclic. This can be verified in polynomial time by constructing $G[V \setminus V']$ (which takes at most $O(n + m)$ time) and then checking whether this graph is acyclic (which can be done in time $O(n + m)$).

Next we reduce $VertexCover$ to $FeedbackVertexSet$. Given instance consisting of an undirected graph $G = (V, E)$ and a parameter K for VERTEX COVER we construct an instance to FEEDBACK VERTEX SET as follows: Create a directed graph $G' = (V, E')$ on the same vertex set by adding two directed edges (u, v) and (v, u) to G' for every edge $\{u, v\} \in E$ of G. We claim that V' is a vertex cover of G if and only if V' is a feedback vertex set of G'. Given a vertex cover V' of G, every edge $\{u, v\} \in E$ must be covered by a vertex in V'. That is, every directed edge $(u, v) \in E'$ has at least one of its endpoints in V'. Thus, G' does not contain a cycle. Suppose we are given a feedback vertex set V' of G'. Then every cycle in G' must contain at least one vertex from V'. In particular, every cycle of length 2 in G' must contain at least one vertex from V'. Every cycle of length 2 corresponds to an undirected edge in G. That is, every undirected edge in G has at least one of its endpoints in V' and thus V' is a vertex cover for G.