Exercises Complexity Theory

Lecture 6

May 18, 2026

Only the exercises where points are given can be handed in.
(The maximum number of points per exercise is written in the margin.)

To be handed in on **June 2, 2026**, in Brightspace under Assignment 6, **deadline: 11:00.**

Exercise 1. Recall the problem QBF from the lecture: decide whether a closed quantified Boolean formula is true. We write altprenex-QBF for the problem QBF restricted to alternating prenex CNF-formulas.

(a) Give two alternating prenex CNF-formulas with at least 3 different variables occurring at least once, one which is a *yes*-instance of altprenex-QBF, and one which is a *no*-instance.

(b) Describe a naive recursive algorithm for solving altprenex-QBF that simply tries all possible substitutions for the bound variables. Prove that the runtime complexity of your algorithm is $\Theta\left(n \cdot 2^k\right)$, where k is the number of quantifiers in the formula, and n is the size of the matrix (the part after the quantifiers).

Exercise 2. For a finite set $S \subseteq \mathbb{N}$, we say that subsets $S_1, S_2 \subseteq S$ *partition* S if they are both non-empty, $S_1 \cap S_2 = \emptyset$ and $S_1 \cup S_2 = S$. Now consider the following decision problem Part:
Given a finite set $S \subseteq \mathbb{N}$, does there exist a partition S_1, S_2 of S such that

$$\sum_{s_1 \in S_1} s_1 = \sum_{s_2 \in S_2} s_2?$$

(3) (a) Give two sets (each with at least 3 elements), one which is a *yes*-instance of Part and one which is a *no*-instance.

(12) (b) Show that Part is **NP**-complete. (*<u>Hint:</u>* Consider the problem SubsSum.)

Exercise 3. Say for each of the following problems whether they are in **P** or **NP**-hard. Give a proof sketch of your answer.

(10) (a) By an edge contraction on $G = (V, E)$ we mean the operation which yields a graph $G' = (V', E')$ where $V' = (V \setminus \{u, v\}) \cup \{w\}$ for some $\{u, v\} \in E$ with $u \neq v$ and $w \notin V$, and $E' = (E \setminus \{\{u, v\}\}) \cup \{\{x, w\} \mid \{x, u\} \in E \text{ or } \{x, v\} \in E\}$. That is, we replace an edge and its two endpoints with a single vertex, maintaining edges adjacent to one of the original vertices. The decision problem of interest: given a graph $G = (V, E)$, can we obtain the complete graph on two nodes K_2 from G by a sequence of edge contractions?

(10) (b) Given a set of elements $X = \{x_1, \ldots, x_n\}$, each with a weight w_i and a value v_i and a knapsack with a carry capacity of C, can we find a subset $X' \subseteq X$ such that

$$\sum_{x_i \in X'} w_i \leq C \text{ and } \sum_{x_i \in X'} v_i \geq V$$

for some target value V?

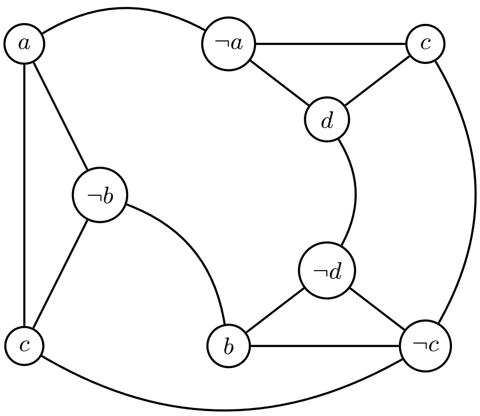
(10) (c) Given a graph $G = (V, E)$, does the graph consist of k connected components?

(10) (d) Given a graph $G = (V, E)$ and a natural number $n \geq 3$, is G the union of n cliques?

Exercise 4. Given an undirected graph $G = (V, E)$, a set of vertices $V' \subseteq V$ is called **independent** if no two vertices of V' are connected by an edge. $\mathsf{Indep}(G, k)$ is the problem of deciding whether for a graph G and a number k, the graph G contains an independent set of k vertices. We show that Indep is **NP**-complete in two ways: by a reduction from 3CNF-SAT and from Clique.

(2) (a) Show that Indep is in **NP**.

(6) (b) In the reduction from 3CNF-SAT to Indep, we define, for a 3CNF formula $\varphi = \bigwedge_{i=1}^{k} C_i$, a graph with $3k$ vertices. The construction is illustrated by an example: If $\varphi = (a \vee \neg b \vee c) \wedge (\neg a \vee c \vee d) \wedge (b \vee \neg c \vee \neg d)$, then the associated graph is:



Give three satisfying assignments $v : \{a, b, c, d\} \to \{0, 1\}$ for φ and for each of these a "corresponding" independent set of size 3.

(5) (c) Based on the construction in part (a), define *precisely* a reducing map f from CNFs to graphs.

(12) (d) Use the map f from part (b) to prove **NP**-hardness of Indep using the **NP**-hardness of 3CNF-SAT.

(5) (e) Prove that Indep is **NP**-hard using a reduction from Clique.

Exercise 5. A *spanning tree* for a graph $G = (V, E)$ is a graph $G' = (V, E')$ with $E' \subseteq E$ such that for each pair of nodes $u, v \in V$, there is exactly one path in G' connecting u and v. The *degree* of a node u i a graph $G = (V, E)$ is the number of edges in E with u as an endpoint.

The decision problem $\mathsf{BSP}(k)$ is defined to be the problem of deciding, for a given graph G, whether there is spanning tree G' for G such that the degree of each node in G' is at most k.

(3) (a) Give a graph G which is a *yes*-instance of $\mathsf{BSP}(2)$, and give a spanning tree $G' \neq G$ witnessing this.

(12) (b) Show that $\mathsf{BSP}(k)$ is **NP**-complete for all $k \geq 2$.