Stockholm University
Department of Mathematics
Lecturer: Marc Hellmuth
Tutor: Anna Lindeberg and Liam Cuclair

2. Exercise "Algorithm and Complexity (DA 4005)"

This exercise counts as a part of Individual Assignment 1

Exercise 1: $\mathcal{P}$ *vs* $\mathcal{NP}$ 2+2=4p
Let B be a decision problem in the class $\mathcal{P}$ and A a decision problem with $A \leq_p B$.

(a) Explain in a few words whether $A \notin \mathcal{NP}$ or $A \in \mathcal{NP}$.

(b) What can you conclude about the "subset relation" of the classes $\mathcal{P}$ and $\mathcal{NP}$ if Problem A is $\mathcal{NP}$-hard. Explain your results.

Exercise 2: NP-completeness 4p
Given is the following problem, called 2-ANSWER-SAT:

> Given a Boolean expression φ as in the usual SAT problem.
> Are there *two* distinct assignments of values "true" and "false" to the Boolean variables such that the expression φ becomes true?

Show that 2-ANSWER-SAT is NP-complete.

Exercise 3: NP-completeness 4p
Show that the following decision problem is NP-complete:

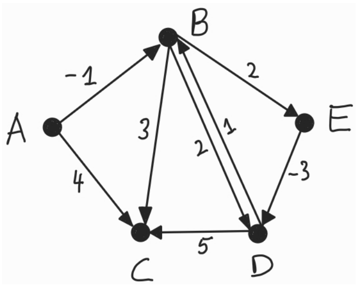
> Given two undirected graphs H and G.
> Is there an isomorphism φ from H to some subgraph $G' \subseteq G$?

Exercise 4: Shortest SIMPLE Paths 2p
To recall Lemma 3.1:

Let $G = (V, E)$ be a digraph with weighting function $w\colon E \to \mathbb{R}$. Let $< v_0, \dots, v_k >$ be a shortest $v_0 - v_k$ path in G. Then, for all $1 \leq i, j \leq k$ it holds that $< v_i, \dots, v_j >$ is a shortest $v_i - v_j$ path in G.

Provide an example that shows that Lemma 3.1 is, in general, not satifsfied for **simple** paths. That is, show that if $< v_0, \dots, v_k >$ is shortest simple $v_0 - v_k$ path in G, then there might be a subpath $< v_i, \dots, v_j >$ that is not a shortest simple $v_i - v_j$ path in G.

Exercise 5: Bellmann-Ford 2.5+2.5+2=7p
Given is the following weighted di-graph $G = (V, E)$:



(a) Verify that the BELLMANN-FORD algorithm applied on G provides a correct solution, that is, the distance from s to any other vertex for all $s \in V$.

(b) Apply BELLMANN-FORD on G by chosing $s = A$ and where all edges are processed in the following order: $(B, E), (D, B), (B, D), (A, B), (A, C), (D, C), (B, C), (E, D)$.

Provide for the initial step and each step i the values $A.d, \ldots, E.d$ whenever one of these values changed. Specify which particular edge in step i caused this change.

(c) Modify the weights in G such that BELLMANN-FORD would fail to compute the distance from A to some other vertex. Explain, in which step and why the algorithm fails.

Deadline: Wedneday - September 24, 2025