TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Department of Computer Science
AG Algorithms and Complexity

Issue Date: 20.04.2015
Version: 2015-04-10 16:28

1st Exercise sheet for Advanced Algorithmics, SS 15

Hand In: *Until Monday, 27.04.2015, 12:00am, in lecture, exercise sessions, hand-in box in stairwell 48-6 or via email.*

Exercise Policy

- The exercise problems take up the material from lecture. Working on them deepens understanding and increases proficiency. As such, we highly recommend you invest some time on the problems.
- Participation – that is, hand-ins and attendance – is optional. However, we will assume that you are familiar with the exercise problems (and, ideally, their solutions) during exams.
- In order to give you feedback on your progress, hand-ins will be graded.
- Exercise sessions will be dynamic in nature, that is we will work on the attendees' problems. Do not expect to get full solutions.

Problem 1

$2 + 3 + 5 + 5$ points

Consider the following problems. Prove that they are $\mathcal{NP}$-complete, respectively.

You may use that the standard problems 3SAT, Hamilton Path, Clique, Knapsack, Subset Sum, Vertex Cover and Traveling Salesperson are $\mathcal{NP}$ -complete.

a) **Subgraph Isomorphism:**

Input: Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

Question: Does G_1 have a subgraph which is isomorphic to G_2?

b) **Longest Path:**

Input: A graph $G = (V, E)$ with edge weights $w : E \to \mathbb{N}$, nodes $s, t \in V$ and $k \in \mathbb{N}$.

Question: Is there a simple path from s to t in G that has weight – w. r. t. w – at least k?

c) **Optimal Composition:**

Input: A finite set A, a set $C \subseteq 2^A$ of subsets of A and $J \in \mathbb{N}$.

Question: Is there a sequence of $j \leq J$ unions

$$\langle x_1 \cup y_1, x_2 \cup y_2, \ldots, x_j \cup y_j \rangle \ ,$$

so that

i) $x_i \cap y_i = \emptyset$ for all $i \in [1..j]$,

ii) $x_i, y_i \in \{\{a\} \mid a \in A\} \cup \{x_k \cup y_k \mid k < i\}$ for all $i \in [1..j]$ and

iii) there is an i with $x_i \cup y_i = c$ for every $c \in C$?

d) **Scheduling with Missed Deadlines:**

Input: A set $T = \{t_1, \ldots, t_n\}$ of jobs, each with duration 1, deadlines $d : T \to \mathbb{N}$, a partial order $\lessdot$ on T and a natural bound $k \leq n$.

Question: Is there a mapping $\sigma : T \to \{1, 2, \ldots, n\}$ which orders the jobs such that

i) $i \neq j \implies \sigma(t_i) \neq \sigma(t_j)$ for all $i, j \in [1..n]$,

ii) $t_i \lessdot t_j \implies \sigma(t_i) < \sigma(t_j)$ for all $i, j \in [1..n]$ and

iii) $|\{t \in T \mid \sigma(t) > d(t)\}| \leq k$?

Advice: Remember what you know from basic courses! What are all the necessary steps? Try to write down at least one of these proofs in formal detail!