TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Department of Computer Science
Sebastian Wild

Issue Date: 10.05.2017
Version: 2017-05-10 14:22

# 3rd Exercise Sheet for Advanced Algorithmics, Summer 17

***Hand In:*** *Until Wednesday, 17.05.2017, 12:00 am, hand-in box in 48-4 or via email.*

## Problem 5

20 points

Design an algorithm that computes a minimal vertex cover for a given *tree* in time $\mathcal{O}(n)$, $n$ the number of tree nodes.

## Problem 6

20 + 30 + 20 points

We consider again the improved depth-bounded search algorithm `BetterFptVertexCover` discussed in class.

a) Show that the search space induced by `BetterFptVertexCover` is $\mathcal{O}(1.4656^k)$ and its worst-case running is in $\mathcal{O}(1.4656^k(n+m))$ for any graph with $n$ nodes and $m$ vertices given in the adjacency-lists representation (where we identify the nodes with $[n]$).

   Be sufficiently precise on how to implement the abstract pseudocode to show that the stated running time is possible.

   **Hint:** You can use `wolframalpha.com` for computations.

b) `BetterFptVertexCover` directly computes an optimal vertex cover if one with at most $k$ nodes exists, but assume for this exercise it would *not*. Rather assume it is given to you as a black-box $\mathcal{O}(1.4656^k(n+m))$-time *decider* `hasVertexCover` for the decision problem $p$-Vertex-Cover, where $k$ is the threshold of the decision problem.

   Design an algorithm `minVertexCoverSize` using galloping search and black-box calls to `hasVertexCover` for the *evaluation* version of the vertex cover problem with an fpt-style running time $\mathcal{O}(f(k)p(n+m))$ where now $k$ is the size of the *optimal* vertex cover in the graph, i.e., the output of your algorithm.

   Analyze the running time of your algorithm and compare it to one for the original `BetterFptVertexCover`. Can you improve the exponential part?

c) Now finally, design an algorithm `minVertexCover` using only black-box calls to `hasVertexCover` and `minVertexCoverSize` for the *optimization* version of the vertex cover problem with an fpt-style running time $\mathcal{O}(f(k)p(n+m))$ where again $k$ is the size of the *optimal* vertex cover in the graph.

Analyze the running time of your algorithm and compare it to one for the original `BetterFptVertexCover`.

**Hint:** Recall Problem 2.