Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, P. Bamberger, H. Ghodselahi, Y. Maus, P. Schneider

Theoretical Computer Science - Bridging Course
Summer Term 2017
Exercise Sheet 7

Hand in (electronically or hard copy) by 12:15 pm, July 3rd, 2017

Exercise 1: $\mathcal{O}$-Notation Formal Proofs (1+2+3 points)

The set $\mathcal{O}(f)$ contains all functions that are asymptotically not growing faster than the function f (when additive or multiplicative constants are neglected). That is:

$$g \in \mathcal{O}(f) \Longleftrightarrow \exists c \geq 0, \exists M \in \mathbb{N}, \forall n \geq M : g(n) \leq c \cdot f(n)$$

For the following pairs of functions, check whether $f \in \mathcal{O}(g)$ or $g \in \mathcal{O}(f)$ or both. Proof your claims (you do not have to proof a negative result $\notin$, though).

(a) $f(n) = 100n,\ g(n) = 0.1 \cdot n^2$

(b) $f(n) = \sqrt[3]{n^2},\ g(n) = \sqrt{n}$

(c) $f(n) = \log_2(2^n \cdot n^3),\ g(n) = 3n$

***Hint**: You may use that $\log_2 n \leq n$ for all $n \in \mathbb{N}$.*

Exercise 2: Sort Functions by Asymptotic Growth (3 points)

Sort the following functions by asymptotic growth using the $\mathcal{O}$-notation. Write $g <_{\mathcal{O}} f$ if $g \in \mathcal{O}(f)$ *and* $g \notin \mathcal{O}(f)$. Write $g =_{\mathcal{O}} f$ if $f \in \mathcal{O}(g)$ *and* $g \in \mathcal{O}(f)$. *Note: For each error you will lose $\frac{1}{2}$ points.*

n^2	$\sqrt{n}$	2^n	$\log(n^2)$
3^n	n^{100}	$\log(\sqrt{n})$	$(\log n)^2$
$\log n$	$10^{100}n$	$n!$	$n \log n$
$n \cdot 2^n$	n^n	$\sqrt{\log n}$	n

Exercise 3: The class $\mathcal{P}$ (1+2+1+2 points)

Show that the following languages ($\cong$ problems) are contained in the class $\mathcal{P}$. Do this by giving an algorithm whose run-time can be bound by a polynomial and decides whether a given input string is in the respective language. Use the $\mathcal{O}$-notation to bound the run-time of your algorithm.[1]

(a) Palindrome:= $\{w \in \{0,1\}^* \mid w \text{ is a palindrome}\}$

(b) List := $\{\langle A, c\rangle \mid A \text{ is a finite list of integers which contains } x, y \in A \text{ such that } x + y = c\}$.

(c) 3-Clique := $\{\langle G\rangle \mid \text{graph } G \text{ has a } \textit{clique} \text{ of size at least 3}\}$

[1] Since it is typically easy (i.e. feasible in polynomial time) to decide whether an input is *well-formed* (cf. lecture), your algorithm only needs to consider well-formed inputs.

(d) 17-DominatingSet := $\{\langle G\rangle \mid$ graph G has a *dominating set* of size at most 17$\}$

- A *clique* of a graph $G = (V, E)$ is a subset $Q \subseteq V$ such that for all $u, v \in Q : \{u, v\} \in E$.
- A *dominating set* of a graph $G = (V, E)$ is a subset $D \subseteq V$ such that for every vertex $v \in V$: $v \in D$ or v adjacent to a node $u \in D$.

Exercise 4: Traveling Salesman Problem (2+3 points)

A *Hamiltonian cycle* in a complete (a graph is *complete* if it is a clique), weighted (with weights in $\mathbb{N}$) graph is a path in that graph, which contains every node exactly once and where the first node of the path equals the last node. Its length is the sum of the weights of its edges. The *decision variant* of the Hamiltonian cycle is given by

D-TSP := $\{\langle G, k\rangle \mid$ undirected, weighted graph G has a hamiltonian cycle of length at most $k\}$.

The objective of the *optimization variant* of the Hamiltonian cycle is to find, for a given complete, weighted graph G, the minimal k such that there is a Hamiltonian cycle in G that has length k. Assume that you already have an algorithm that solves D-TSP, i.e., it tells you for $\langle G, k\rangle$ whether $\langle G, k\rangle \in$ D-TSP or not.

(a) Show how to use a polynomial number of repetitions of the *D-TSP* algorithm to determine the length k of a shortest Hamiltonian cycle of a given graph G.

(b) Use the previous result to show that if the decision variant of TSP is in $\mathcal{P}$ then we can find the edges of a shortest Hamiltonian cycle in $\mathcal{P}$ as well.[2]

Hint: Try to find out whether an edge is needed for a shortest Hamiltonian cycle by manipulating its weight.

[2] This is called a polynomial reduction.