MTAT.03.286: Advanced Methods in Algorithms **University of Tartu**

Final exam

Course staff: Vitaly Skachek, Pille Pullonen May 25th, 2016

Student name: ______________________________________

Student ID: ____

1. This exam contains 10 pages. Check that no pages are missing.
2. It is possible to collect up to 110 points. Try to collect as many points as possible.
3. Justify and prove all your answers (where applicable).
4. All facts and results that were proved or stated in the class can be used in your solution without a proof. Such results need to be rigorously formulated.
5. Any printed and written material is allowed in the class. No electronic devices are allowed.
6. Exam duration is 3 hours.
7. Good luck!

Question 1	
Question 2	
Question 3	
Question 4	
Total	

Question 1 (25 points). You are given the following linear-programming (LP) problem:

$$\begin{array}{rrcrcrcl}
\textbf{max} & x_1 & - & 3x_2 & + & x_3 & & \\
\textbf{s.t.} & x_1 & + & x_2 & & & \leq & 1 \\
 & x_1 & + & & & x_3 & \leq & 2 \\
 & & & x_2 & + & 2x_3 & \leq & 2 \\
 & \multicolumn{7}{c}{x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0}
\end{array}$$

(a) Solve this LP problem by using the simplex method.

(b) Formulate the dual LP problem.

(c) Verify that your solution of (a) is optimal by the substitution of the point $(0, 1, 0)$ into the dual problem. Explain your answer.

Question 2 (25 points). Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a finite undirected graph with the real-valued weight function w defined on the edges, such that $w(e) > 0$ for all edges $e \in \mathcal{E}$. It is known that the weights of edges in $\mathcal{G}$ are all different. Let e_1 be some edge, and $\mathcal{C}$ be a simple circuit that contains e_1. Prove or disprove the following statements.

(a) If e_1 has the lowest weight in $\mathcal{C}$, then there exists a minimum spanning tree in $\mathcal{G}$ that contains e_1.

(b) If e_1 has the highest weight in $\mathcal{C}$, then there exists no minimum spanning tree in $\mathcal{G}$ that contains e_1.

Question 3 (25 points). You are given an integer number k and a flow network $\mathcal{N}(\mathcal{G}(\mathcal{V}, \mathcal{E}), s, t, c)$, such that all edge capacities are $c(e) = 1$. Propose an algorithm, which finds a subset of k edges of $\mathcal{E}$, such that by deleting these edges from $\mathcal{G}$, the maximum flow from s to t in the resulting network is minimal. The algorithm should have time complexity of $O(|\mathcal{V}|^2|\mathcal{E}|)$. Prove correctness of that algorithm and analyze its time complexity.

Question 4 (35 points).

A student is moving to a new apartment. He needs to take n objects a_1, a_2, ..., a_n with him. The object a_i has weight $w_i > 0$ kilograms, $i = 1, 2, \cdots, n$. The student's car can carry at most k kilograms at a time (assume that there are no other restrictions on the objects placed in the car). The student wants to minimize m, which is the number of times he drives from the old apartment to the new apartment.

Consider the following greedy algorithm. The objects are ordered in an arbitrary order. The student loads the objects into the car in that order as long as the total weight of the loaded objects does not exceed k. If the next object a_i does not fit into the car (the load exceeds k kilograms), then the student leaves a_i in the old apartment, and drives with the items that are already loaded. The student then returns and continues the same algorithm starting with the object a_i.

Denote by ℓ_j, $j = 1, 2, \cdots, m$, the load of the car when the student drives j-th time from the old apartment to the new apartment.

(a) Prove that for any $j = 1, 2, \cdots, m - 1$ it holds that $\ell_j + \ell_{j+1} > k$.

(b) Show that the above greedy algorithm achieves approximation factor 2.

(c) Show that for every set of objects there exists an initial ordering such that m is optimal.

(d) Show an example of k, of n, of weights $w_1, w_2, \cdots, w_n$, and of the object ordering, such that by using the above algorithm, $m \geq 1.9 \cdot \mathrm{OPT}$, where OPT is the optimum value.