École Polytechnique Fédérale de Lausanne
Prof. Eisenbrand

February 22, 2018

Discrete Optimization (Spring 2018)

Assignment 1

Problem 8 can be **submitted** until March 2 12:00 noon, please send the source code in C++ to `igor.malinovic@epfl.ch` . You are allowed to submit your solutions in groups of at most three students.

Problem 1

Provide a certificate (as in Theorem 0.1 in the lecture notes) of the unsolvability of the linear equation

$$\begin{pmatrix} 2 & 1 & 0 \\ 5 & 4 & 1 \\ 7 & 5 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$$

Problem 2

Show the "if" direction of the Farkas' lemma: given $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, if there exist a $\lambda \in \mathbb{R}^m_{\geqslant 0}$ such that $\lambda^\top A = 0$ and $\lambda^\top b = -1$, then the system $Ax \leq b$ is unfeasible.

Problem 3

Consider the following linear program:

$$\begin{array}{lrcrcl} \max & x & + & y & & \\ \text{s.t.} & 3x & + & 2y & \leq & 6 \\ & x & + & 4y & \leq & 4. \end{array}$$

The solution $(x, y) = (8/5, 3/5)$ satisfies the both constraints and has the objective value $11/5$. Provide a certificate that this is an optimal solution.

Problem 4

Find the binary representation of 235.

Problem 5

Show that the binary representation with leading bit one of a positive natural number is unique.

Problem 6

Show that there are n-bit numbers $a, b \in \mathbb{N}$ such that the Euclidean algorithm on input a and b performs $\Omega(n)$ arithmetic operations. *Hint: Fibonacci numbers*

Problem 7

Suppose we are given three $n \times n$ matrices $A, B, C \in \mathbb{Z}^{n \times n}$ and we want to test whether $A \cdot B = C$ holds. We could multiply A and B and then compare the result with C. This would amount to running time (number of arithmetic operations) of $O(n^3)$ with the standard matrix-multiplication algorithm.
We now show how to perform an efficient *randomized test*. Suppose that you can draw a vector $v \in \{0, 1\}^n$ i.i.d. at random in time $O(n)$. The idea is then to compute the product $B \cdot v$ and then the product $A \cdot (B \cdot v)$ and afterwards $C \cdot v$, all in time $O(n^2)$. Show the following.

a) If $A \cdot B \neq C$, then $P(A \cdot (B \cdot v) = C \cdot v) \leq 1/2$.

b) Let $v_1, \dots, v_k \in \{0,1\}^n$ be i.i.d. at random and suppose that $A \cdot B \neq C$. The probability of the event: $A \cdot (B \cdot v_i) = C \cdot v_i$ for each $i = 1, \dots, k$ is bounded by $1/2^k$.

c) Conclude that there is an algorithm that runs in time $O(k \cdot n^2)$ which tests whether $A \cdot B = C$ holds. The probability that the algorithm gives the wrong result is bounded by $1/2^k$.

Problem 8 ($\star$)

Let a and b be two natural numbers with binary representations $a_0, \dots, a_{l-1}$ and $b_0, \dots, b_{l-1}$, respectively. Given that $a > b$ design an algorithm which outputs $c = a - b$ in its binary representation with leading bit one. Additionally, we require this algorithm to have the running time of $O(l)$ basic operations. The algorithm shall be implemented in C++.