École Polytechnique Fédérale de Lausanne
Prof. Eisenbrand

February 24, 2017

Discrete Optimization (Spring 2017)

Assignment 1

Problem 8 can be **submitted** until March 3 12:00 noon into the right box in front of MA C1 563.
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

Find the binary representation of 134.

Problem 4

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

Problem 5

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 6

The *determinant* of a matrix $A \in \mathbb{R}^{n\times n}$ can be computed by the recursive formula

$$\det(A) = \sum_{i=1}^{n} (-1)^{1+j} a_{1j} \det(A_{1j}),$$

where A_{1j} is the $(n-1)(n-1)$ matrix that is obtained from A by deleting its first row and j-th column. This yields the following recursive algorithm (see the lecture notes, Example 1.4).

Input: $A \in \mathbb{R}^{n\times n}$
Output: $\det(A)$
if $(n = 1)$
  return a_{11}
else
  $d := 0$
  for $j = 1, \ldots, n$
    $d := (-1)^{1+j} \det(A_{1j}) + d$
  return d

Let $A \in \mathbb{R}^{n\times n}$ and suppose that the n^2 components of A are pairwise different.

1. Suppose that B is a matrix that can be obtained from A by deleting the first k rows and k of the columns of A. How many (recursive) calls of the form $\det(B)$ does the algorithm create?
2. How many different submatrices can be obtained from A by deleting the first k rows and some set of k columns? Conclude that the algorithm remains exponential, even if it does not expand repeated subcalls.

Problem 7

Complete the algorithm below such that it adds two natural numbers in binary representation $a_0, \ldots, a_{l-1},\ b_0, \ldots, b_{l-1}$. What is the asymptotic running time (number of basic operations) of your algorithm? Can there be an asymptotically faster algorithm?

Input: Two natural numbers a and b in their binary representation $a_0, \ldots, a_{l-1},\ b_0, \ldots, b_{l-1}$.
Output: The binary representation $c_0, \ldots, c_l$ of $a + b$

carry := 0
for $i = 0, \ldots, l-1$
    $c_i = \text{carry} + a_i + b_i \pmod 2$
    carry :=
c_l :=
return $c_0, \ldots, c_l$

Problem 8 ($\star$)

Recall the Gaussian elimination algorithm.

Input: $A \in \mathbb{Q}^{m\times n}$
Output: A' in row echelon form such that there exists an invertible $Q \in \mathbb{Q}^{m\times m}$ such that $Q \cdot A = A'$.

$A' := A$
$i := 1$
while $(i \leq m)$
    find *minimal* $1 \leq j \leq n$ such that there exists $k \geq i$ such that $a'_{kj} \neq 0$
    If no such element exists, then **stop**
    swap rows i and k in A'
    for $k = i+1, \ldots, m$
        subtract (a'_{kj}/a'_{ij}) times row i from row k in A'
    $i := i + 1$

It can be shown (see Lecture notes, section 1.1) that the algorithm runs in polynomial time, in particular that during any iteration of the while loop the entries of A' have polynomial size in the binary encoding length of the input matrix A. Using this, show that the matrix $Q \in \mathbb{Q}^{m\times m}$ that transforms A into A' in the Gaussian algorithm via $QA = A'$ has entries that are of polynomial size in the binary encoding length of A.