École Polytechnique Fédérale de Lausanne
Prof. Eisenbrand

March 1, 2018

Discrete Optimization (Spring 2018)

Assignment 2

Problem 5 can be **submitted** until March 9 12:00 noon into the box in front of MA C1 563. You are allowed to submit your solutions in groups of at most three students.

Problem 1

Describe an algorithm that multiplies two n-bit integers in time $O(n^2)$. You may assume to have a subroutine $Sum(d, e)$ which returns the sum of two n-bit natural numbers d and e in time $O(n)$.

Problem 2

Suppose $a, b \in \mathbb{N}$ are two n-bit integers, where n is a power of 2. Consider the first and the last $n/2$ bits of a, and denote their corresponding decimal numbers with a' and a'', respectively. Likewise decimal numbers b' and b'' correspond to the first and the second half of the bit-representation of b.

i) Show that $a = a' + a'' \cdot 2^{n/2}$ and $b = b' + b'' \cdot 2^{n/2}$.

ii) Show that $a \cdot b = a' \cdot b' + (a' \cdot b'' + a'' \cdot b') \cdot 2^{n/2} + a'' \cdot b'' \cdot 2^n$.

iii) Show that $(a'b'' + a''b') = (a' + a'')(b' + b'') - a' \cdot b' - a'' \cdot b''$.

iv) Design a recursive algorithm for n-bit integer multiplication whose running time $T(n)$ satisfies the recursion

$$T(n) \leq 3 \cdot T(n/2) + c \cdot n,$$

where $c > 1$ is some constant.

Hint: You can assume that there is a constant c' such that two n-bit numbers can be added and subtracted using at most $c'\, n$ basic operations.

v) *Unroll* the recursion above three times.

vi) Conclude that two n-bit numbers can be computed in $O(n^{\log_2(3)})$ elementary bit operations.

Problem 3

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

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

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.

i) 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?

ii) 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 4

In this exercise, you will see that matrix multiplication is in some sense not harder than matrix inversion.
Suppose that $I(n)$ with $I(n) = \Omega(n^2)$ is a function that satisfies $I(3\,n) = O(I(n))$ and that a non-singular $n \times n$ matrix can be inverted using $I(n)$ arithmetic operations. Show that two $n \times n$ matrices A and B can be multiplied using $O(I(n))$ arithmetic operations.
Hint: Construct an upper triangular $3n \times 3n$-matrix that contains A and B.

Problem 5 ($\star$)

Let M_{2^k} be a matrix of order $n := 2^k$, where $k \in \mathbb{N}_{>0}$ such that it is recursively defined as follows:

$$M_{2^k} = \begin{pmatrix} M_{2^{k-1}} & M_{2^{k-1}} \\ M_{2^{k-1}} & -M_{2^{k-1}} \end{pmatrix} \tag{1}$$

and $M_1 = [1]$. Prove that $|\det(M_n)| = n^{n/2}$, i.e. that the Hadamard bound is tight.