École Polytechnique Fédérale de Lausanne
Prof. Eisenbrand

April 8, 2025

**Discrete Optimization** (Spring 2025)

# Assignment 8

1) Let $a, e, k \in \mathbb{N}$ be three given natural numbers.

   (a) Argue that $a^{2^k}$ can be computed using $\Theta(k)$ multiplications.

   (b) How many bits ($\Theta$-notation) does $a^{2^k}$ have?

   (c) Let $(e_0, \ldots, e_l)$ be the bit representation of $e$, that is, $e = \sum_{i=0}^{l} e_i 2^i$ with $e_i \in \{0, 1\}$ for $i = 0, \ldots, l$. Complete the following algorithm by replacing each occurrence of three question marks (???) so that it computes $a^e$ using $O(l)$ many arithmetic operations.

$$
\begin{aligned}
&E = 1 \\
&S = a \\
&\text{For } (i = 0 \text{ to } l) \\
&\quad \text{if } (e_i == 1) \\
&\qquad E = E \cdot ??? \\
&\quad S = ??? \\
&\text{return } ???
\end{aligned}
$$

   (d) Show that for given $a, e, N \in \mathbb{N}$ one can compute $a^e \pmod N$ in time polynomial in the binary encoding length of $a, e$ and $N$.

   (e) Let $a, b, c, N \in \mathbb{N}$ be given and suppose that $N$ is a prime number. Show that $a^{b^c} \pmod N$ can be computed in polynomial time in the binary encoding length of $a, b, c$ and $N$. You may use Fermat's little theorem: $a^N \equiv a \pmod N$.

2) There are n types of animals, and you want to assign them to two stables. Unfortunately, some animals would eat other animals when left unattended. Therefore you need to assign the animals carefully. There are m relations of the form "u eats v", where u and v are animals. Find an $O(n + m)$ algorithm that decides whether there is an assignment of animals to the two stables such that no animal eats another one of the same stable, and outputs a feasible assignment.
   *Hint: Breadth-first-search might be useful.*

3) 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:

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

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

4) 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 $(n-1) \times (n-1)$ matrix that is obtained from $A$ by deleting its first row and j-th column. This yields the following recursive algorithm.
Input: $A \in \mathbb{R}^{n \times n}$
Output: $\det(A)$

```
if (n = 1)
return a_11
else
   d := 0
   for j = 1, ..., 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.

(a) 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?

(b) 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.