Prof. Friedrich Eisenbrand
Alfonso Cevallos and Yuri Faenza

Discussions from: April 01, 2014

Computer Algebra

Spring 2014

Assignment Sheet 4

Exercises marked with a $\star$ can be handed in for bonus points. Due date is April 15.

Exercise 1
Show that the following alternative algorithm for computing the gcd of $a, b \in \mathbb{N}$ is correct and give an upper bound on its running time.

INPUT: $a, b \in \mathbb{N}$, OUTPUT: gcd(a,b).
SET $r = a$, $r' = b$, $e = 0$.
WHILE $2|r$ and $2|r'$

  SET $r = r/2$, $r' = r'/2$, $e = e + 1$.

WHILE $r' \neq 0$

  WHILE $2|r$ SET $r = r/2$.

  WHILE $2|r'$ SET $r' = r'/2$.

  IF $r' < r$ SET $(r, r') = (r', r)$.

  SET $r' = r' - r$.

RETURN $r \cdot 2^e$.

Exercise 2
Let $A \in \mathbb{Q}^{n \times n}$. Denote the columns of A by $a_1, \ldots, a_n$. Let B be an upper bound on the absolute values of entries in A.

1. Give a formal proof (only sketched in class) of the Hadamard bound $|\det(A)| \le \prod_{j=1}^{n} |a_j|_2$, where $|\cdot|_2$ is the Euclidean norm, using the Gram-Schmidt orthogonalization process. Derive from this that $|\det(A)| \le n^{n/2} B^n$.

2. Prove Leibniz's bound $|\det(A)| \le B^n n!$. How does it compare to Hadamard bound?

Exercise 3 ($\star$)
In class we stated (without proving it) that

- $\diamond$ The number of bit operations for performing Gaussian elimination is polynomial in the bit size of the input.

First prove

- For a matrix $A \in \mathbb{Z}^{n \times n}$ with all entries bounded in absolute value by Δ, one has $\log(|\det(A)|) = O(n\log(n) + n\log(\Delta)$.

Then prove $\diamond$.

Exercise 4
Let $M \in \mathbb{R}^{m \times n}$, and M' be the matrix obtained after performing an elementary row operation on M.

a) Show that there exists an invertible matrix X such that $M' = XM$.

b) Let B be the output of Gaussian elimination when applied to A. From *a*), one immediately checks that $B = XA$ for some invertible matrix X. Modify the Gaussian elimination algorithm as to compute X.

Exercise 5
Let $T(G)$ be the Tutte matrix of a graph G, and $\nu(G)$ the cardinality of the maximum matching of G.

a) Given a graph G, show that there exists a subgraph H of G with a perfect matching such that $2\nu(G) = 2\nu(H) = \text{rank}(T(H)) = \text{rank}(T(G))$.

b) Give an efficient randomized algorithm for computing the cardinality of a maximum matching of a graph that outputs the correct answer with probability at least $1/2$.

Exercise 6 ($\star$)
Implement an algorithm that takes as input the adjacency matrix of a graph, and then uses the Tutte matrix and the Schwartz-Zippel lemma to check if the graph has a perfect matching.