Exercises marked with a ⋆ 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** \(2 | r \text{ and } 2 | r'\):

- **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 \(|\det(A)| \leq \prod_{j=1}^n |a_j|_2\), where \(\cdot |_2\) is the Euclidean norm, using the Gram-Schmidt orthogonalization process. Derive from this that \(|\det(A)| \leq n^{n/2} B^n\).

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

**Exercise 3 (⋆)**
In class we stated (without proving it) that
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 $\Delta$, 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 (★)
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.