HOMEWORK 3

Due: 07.05

1 Modular determinant computation

The problem of computing the determinant $\det A \in \mathbb{Z}$ of an $n \times n$ non-singular matrix $A = (a_{i,j}) \in \mathbb{Z}^{n\times n}$ can be solved in $\mathcal{O}(n^3)$ operations in $\mathbb{Q}$ by means of Gaussian elimination. However, one should take this complexity statement with care: if we start counting the number of *word operations* performed during the elimination, the absolute values on numerators and denominators involved grow quite rapidly. There exist a rather involved proof that the length of these values remain polynomial in the number of words, but in this exercise we develop an alternative approach that uses modular computation.

1. First, we would like to have an a priori bound on the output of the computation $|\det A|$. Show that $|\det A| \leq n^{n/2}B^n$, where $B = \max_{1\leq i,j\leq n} |a_{i,j}|$. (This inequality is known as Hadamard's inequality).

2. Denote $C = n^{n/2}B^n$ and let p be prime between $2C$ and $4C$ (there exist probabilistic polynomial time algorithm that finds such p). Now the goal is to compute $\det(A \bmod p)$. For that, modify Gaussian elimination as follows: except of dividing by a pivot element $a_{i,j}$ over $\mathbb{Q}$, we calculate its inverse modulo p. The rest of the computations are performed $\text{mod} p$, where the set of representatives $\text{mod} p$ is $[-(p-1)/2, (p-1)/2]$. Show that the algorithm returns $\det(A)$ (over $\mathbb{Z}$) and argue that the number of *word* operations needed is $\mathcal{O}(n^5(\log n + \log B)^2)$ (without using fast integer arithmetic).

3. Improve the above choosing several small primes. Give an explicit algorithm, argue that it is correct and show its running time (you may assume that getting these primes is for free). *Hint*: you may use the fact that the first r primes are of bit-size $\mathcal{O}(\log r)$.

2 n-th roots of integers

We consider the problem of computing a n-th root of an integer a, where n is odd, or certifies that a does not have such a root.

1. Assume a is odd. Formulate the Newton iteration for the problem. Find a modulus for which 1 is a good starting value (and explain why).

2. Assume that a is the n-th power of some integer b. Explain how Newton's iteration can give exactly b, not only modulo a prime, if it is done for enough rounds. (Hint: you may use the uniqueness property of TD10.)

3. Deduce an algorithm that solves the problem, and estimate its complexity. Recall that computations in $\mathbb{Z}/N\mathbb{Z}$ can be done in $O(\mathsf{M}(N))$.

3 Quasi-Cauchy matrices

Let $\mathbf{x}, \mathbf{y} \in K^n$. We assume that $x_i \neq y_j$ for all i, j and that $x_i \neq x_j, y_i \neq y_j$ for $i \neq j$. The *Cauchy matrix* associated to these vectors is the matrix $C(\mathbf{x}, \mathbf{y}) := (1/(x_i - y_j))_{i,j}$. For any $\mathbf{w} := (w_0, \ldots, w_j)$, we define $D(\mathbf{w})$ as the diagonal matrix whose entries are the w_j's.

For any $(\mathbf{x}, \mathbf{y}) \in K^n \times K^n$, we define the $(\mathbf{x}, \mathbf{y})$-*displacement rank* of a matrix A as the rank of the matrix $\varphi_{\mathbf{x},\mathbf{y}}(A) := D(\mathbf{x}) \cdot A - A \cdot D(\mathbf{y})$. For the sake of simplicity, we consider $(\mathbf{x}, \mathbf{y})$'s such that $\varphi_{\mathbf{x},\mathbf{y}}$ is an invertible map.

1. What is the $(\mathbf{x}, \mathbf{y})$-displacement rank of $C(\mathbf{x}, \mathbf{y})$?

2. Let $\mathbf{u}, \mathbf{v} \in K^n$ be two (column) vectors. Show that $\varphi^{-1}_{\mathbf{x},\mathbf{y}}(\mathbf{u} \cdot \mathbf{v}^T) = D(\mathbf{u})C(\mathbf{x}, \mathbf{y})D(\mathbf{v})$.

3. Deduce that if M has $(\mathbf{x}, \mathbf{y})$-displacement rank r, then there exist vectors $\mathbf{g}_1, \ldots, \mathbf{g}_r, \mathbf{h}_1, \ldots, \mathbf{h}_r$ such that

$$M = \sum_{j=1}^{r} D(\mathbf{g}_j)C(\mathbf{x}, \mathbf{y})D(\mathbf{h}_j). \tag{1}$$

   (Hint: if N has rank r, then N can be written as a sum of rank 1 matrices).

4. Conversely, prove that if M is of the form (1), then it has $(\mathbf{x}, \mathbf{y})$-displacement rank at most r.

   For the rest of this exercise, we say that a matrix M with $(\mathbf{x}, \mathbf{y})$-displacement rank r is represented by $(\mathbf{x}, \mathbf{y})$-generators of size r if M is given as a sequence of vectors $(\mathbf{g}_1, \ldots, \mathbf{g}_r, \mathbf{h}_1, \ldots, \mathbf{h}_r) \in (K^n)^{2r}$ such that (1) holds.

   This means that M can be represented by using $O(rn)$ elements, which is compact when the $(\mathbf{x}, \mathbf{y})$-displacement rank is low. In the following, we will study whether basic arithmetic of matrices can be done using this representation.

   Recall that if M is a Cauchy matrix and $\mathbf{v}$ is a vector, the product $M\mathbf{v}$ can be computed in $O(M(n) \log n)$ operations in K.

5. If M is represented by $(\mathbf{x}, \mathbf{y})$-generators of size r, prove that the product $M\mathbf{v}$ can be computed in $O(rM(n) \log n)$ operations.

6. If M, M' are represented by $(\mathbf{x}, \mathbf{y})$-generators of size r, resp. r', give $(\mathbf{x}, \mathbf{y})$-generators of size $r + r'$ for $M + M'$, and show that they can be computed in time $O((r + r')n)$.

7. If M, resp. M' are represented by $(\mathbf{x}, \mathbf{y})$-generators of size r, resp. by $(\mathbf{y}, \mathbf{z})$-generators of size r', give $(\mathbf{x}, \mathbf{z})$-generators of size $r + r'$ for $M \cdot M'$, and show that they can be computed in $O(rr'M(n) \log n)$.