Assignment 4

CS 487/687 – CM 730
due Wednesday April 4, 5pm
March 15, 2018

Submission by email to cs487@student.cs.uwaterloo.ca

1. (20 marks.) Implement, in the language of your choice, the Wiedemann algorithm (as we saw in class) for solving a sparse linear system. As input, the algorithm will take:
   - an integer \( n \) (size of the matrix),
   - a prime integer \( p \) (we work in \( \mathbb{F}_p \)),
   - \( n \) integers in \( \{0, \ldots, p-1\} \) (this is the right-hand vector \( b \))
   - a sequence of triples \( i, j, a_{i,j} \), with \( i, j \) in \( \{1, \ldots, n\} \) and \( a_{i,j} \) in \( \{0, \ldots, p-1\} \) (these are the non-zero entries in the matrix \( A \)).

The output is the \( n \) coordinates of the vector \( x \) such that \( Ax = b \). The algorithm is randomized, but if something goes wrong, you can detect it; in this case, return “error”, or throw an exception, or something along these lines. We give input matrices on the assignment page, with the same format as above. Your code should pass the inputs of size up to 10 and 100 with \( p = 9001 \) (size 1000 should also be not too hard); the input matrix with coefficients modulo \( p = 2 \) is more challenging (there is a good chance the algorithm will fail).

You have to implement the algorithm for discovering a recurrence for a given sequence, but you do not have to reimplement polynomial multiplication or division. Remember, all computations must be done modulo \( p \). Maple, Julia, Sage, … all allow you to do this; I don’t know about Matlab, though.

2. (15 marks.) In this problem, we work with polynomials that have coefficients in \( \mathbb{Q} \) (this would not work over \( \mathbb{F}_2 = \{0, 1\} \)).

Let \( P \) be monic in \( \mathbb{Q}[T] \) and consider a factorization of it over the complex numbers:

\[
P(T) = \prod_{1 \leq i \leq s} (T - e_i);
\]

1
the \(e_i\)'s are in \(\mathbb{C}\), and do not have to be all distinct. For \(j \geq 0\), the \(j\)th power sum of \(P\) is
\[
\pi_j = \sum_{1 \leq i \leq s} e_i^j.
\]

It seems as if the \(\pi_j\)'s are in \(\mathbb{C}\), but we will see that they are actually in \(\mathbb{Q}\).

(a) For \(P = T^2 - 2\), what are the \(\pi_j\)'s?

(b) Let \(Q\) be the reciprocal of \(P\): if \(P = p_0 + p_1 T + \cdots + p_{s-1} T^{s-1} + T^s\), \(Q = p_0 T^s + p_1 T^{s-1} + \cdots + p_{s-1} T + 1\). Using the factorization of \(P\), write the factorization of \(Q\). Hint: you can rewrite \(Q\) as \(Q = T^s P(1/T)\).

(c) Give the power series expansion of \(Q'/Q\) in terms of the \(\pi_j\)'s. Hint: for any polynomials \(A, B\),
\[
\frac{(AB)'}{AB} = \frac{A'}{A} + \frac{B'}{B}.
\]

(d) Show that as a result, the \(\pi_j\)'s are in \(\mathbb{Q}\)

(e) Given \(\pi_1, \ldots, \pi_s\), give a \(O(s^2)\) algorithm to recover \(Q\), then \(P\).

(f) (bonus 5 marks). Over \(\mathbb{F}_2\), show that it may not be possible to recover \(P\) from \(\pi_1, \ldots, \pi_s\).

3. (15 marks.) In this problem, we work with matrices with coefficients in \(\mathbb{Q}\); the algorithm would not work over, for instance, \(\mathbb{F}_2 = \{0, 1\}\).

We consider a matrix \(A\) of size \(n\), and we want to compute its characteristic polynomial \(C \in \mathbb{Q}[T]\). There are two equivalent definitions for it:

- \(C\) is the determinant of \(T I_n - A\), where \(I_n\) is the identity matrix
- \(C\) is the unique monic polynomial whose power sums \(\pi_j, j \geq 0\), are given by \(\pi_j = \text{trace}(A^j)\).

Using the second definition,

(a) Give an algorithm that computes \(C\) in \(O(n^{\omega + 1})\) operations.

(b) Give an algorithm that computes \(1, A, A^2, \ldots, B = A^{\sqrt{n}}\) and \(1, B, B^2, \ldots, B^{\sqrt{n}}\) in \(O(n^{\omega + 1/2})\) operations.

(c) Given two matrices \(U, V\) of size \(n\), prove that you can compute \(\text{trace}(UV)\) in \(O(n^2)\) operations.

(d) Deduce that you can actually compute \(C\) in \(O(\max(n^{\omega + 1/2}, n^3))\) operations. Hint: any number in \(\{1, \ldots, n\}\) can be written \(i + j \sqrt{n}\), for some \(0 \leq i, j \leq \sqrt{n}\).

(e) Given two matrices \(R, S\), with \(R\) of size \((\sqrt{n}, n)\) and \(S\) of size \((n, \sqrt{n})\), prove that you can compute \(RS\) using \(O(n^{(\omega + 1)/2})\) operations. Hint: cut these matrices into square blocks.
(f) (bonus, 8 marks) Deduce that you can actually compute $C$ in $O(n^{\omega+1/2})$ operations.

4. (2 marks.) How much time did you spend on the assignment?