MATH 895, Assignment 5, Summer 2009

Instructor: Michael Monagan

Please hand in the assignment by 11am on August 10th.
Late Penalty -20% off for up to 24 hours late, zero after that.

Question 1: Fraction-Free Gaussian Elimination

Reference: Ch. 9 of *Algorithms for Computer Algebra* by Geddes, Czapor and Labahn.
Let D be an integral domain and let A be an n by n matrix with entries $A_{i,j} \in D$. To compute $det(A)$ we can use ordinary Gaussian elimination over the fraction field D/D or Bareiss' fraction-free Gaussian elimination over D.

Implement both algorithms as Maple procedures GaussElim(A,n) and FracFree(A,n) for the integral domain $D = \mathbb{Z}[x_1, x_2, ..., x_n]$. The algorithms should return $det(A)$ and the eliminated matrix $A^{(n)}$.

To do the rational function arithmetic in D/D in Gaussian elimination, use the `normal(...);` command in Maple. For the exact polynomial divisions in the Bareiss algorithm, use `divide(...,...,'Q');` to compute the quotient Q. Also, you'll need to take care of pivoting – if at any step k, the pivot $A_{k,k} = 0$ and $A_{i,k} \neq 0$ for some $k < i \leq n$, interchange row k with row i before proceeding.

Test both algorithms on the Vandermonde matrices

$$V_n = \begin{bmatrix} {x_1}^{n-1} & {x_1}^{n-2} & \dots & x_1 & 1 \\ {x_2}^{n-1} & {x_2}^{n-2} & \dots & x_2 & 1 \\ \dots & \dots & \dots & \dots & \\ {x_n}^{n-1} & {x_n}^{n-2} & \dots & x_n & 1 \end{bmatrix}$$

for $n = 1, 2, 3, 4, 5, 6, 7$ and verify that the determinants are computed correctly. For $n = 4$ also print out the determinant and the final matrix $A^(n)$ for both algorithms.

Question 2: Rational Number Reconstruction.

Reference: Maximal Quotient Rational Reconstruction: An Almost Optimal Algorithm for Rational Reconstruction by M. Monagan.
Study Wang's rational number reconstruction algorithm and Monagan's maximal quotient rational reconstruction algorithm (MQRR) in the paper by Monagan. Implement both algorithms in Maple as procedures `Wang` and `MQRR` respectively. For Wang's algorithm, use $N = D = \lfloor\sqrt{m/2}\rfloor$. For Monagan's algorithm, use $T = 1000\lfloor\log_2 m\rfloor$. Execute Wang's algorithm on the following input

```
> m := 23;
> M := floor(sqrt(m/2));
> r := [ seq( Wang(u,m,M), u=0..m-1 ) ];
```

Observe that all rationals n/d satisfying $|n| \leq 3$ and $0 < d \leq 3$ appear once in r. Execute Monagan's and Wang's algorithm on the following inputs

```
> p1 := 2^31-1; p2 := prevprime(p1); m := p1*p2;
> U := [ 2/12345678901, 12345678901/2, 123456/78901 ] mod m;
> Digits := 20; M := floor(sqrt(m/2));
> [ seq( Wang(u,m,M), u=U ) ];
> T := 1000*ilog2(m);
> [ seq( MQRR(u,p,T), u=U ) ];
```

The Maple command `ilog2(m)` computes $\lfloor\log_2 m\rfloor$.

Question 3: Solving $Ax = b$ using p-adic lifting.

Part (a)

Let $A \in \mathbb{Z}^{n \times n}$ and $b \in \mathbb{Z}^n$. In class I presented an algorithm for solving $Ax = b$ for $x \in \mathbb{Q}^n$ using linear $p-$adic lifting and rational number reconstruction. Implement the algorithm in Maple as the procedure `PadicLinearSolve(A,b)`. You procedure should return the solution vector x and also print out the number of lifting steps k that are required. Test your implementation on the following examples. The first has large rationals in the solution vector. The second is constructed so that the solution vector x has very small rationals.

```
> with(LinearAlgebra):
> B := 2^16; m := 3; U := rand(B^m);
> n := 50;
> A := RandomMatrix(n,n,generator=U);
> b := RandomVector(n,generator=U);
> x := padicLinearSolve(A,b);
> convert( A.x-b, set ); # should be {0}
> y := [1,0,-1/2,2/3,4,3/4,-2,-3,0,-1];
> x := Vector( [seq( op(y), i=1..5 )] );
> b := A.x;
> b := 12*b; A := 12*A; # clear fractions
> x := padicLinearSolve(A,b);
> convert( A.x-b, set ); # should be {0}
```

To compute $A^{-1} \bmod p$ use `Inverse(A) mod p`.
To multiply A times a vector x use `A.x`.
For rational number reconstruction use the Maple command `iratrecon`.

Part (b)

Suppose $\dim A = n$, $\dim b = n$ and $|A_{i,j}| < B^m$ and $|b_i| < B^m$, i.e., the coefficients in the linear system are m base B digits (or less). Suppose the $p - adic$ lifting algorithm does L lifting steps, i.e. solves $Ax = b \bmod p^L$ and then successfully reconstructs $x \in \mathbb{Q}^n$ using rational reconstruction.

What is the running time of the algorithm assuming classical algorithms are used for integer arithmetic, rational reconstruction and matrix inverse. Express your answer in the form $O(f(m, n, L))$.

Since the integers in the solution vector x may be as large as mn base B digits, as illustrated by the first example, $L \in O(mn)$ in general. What is the running time for $L \in O(mn)$? You should find that $p-$adic method is faster than the Chinese remaindering method.