MATH 895, Assignment 2, Summer 2019

Instructor: Michael Monagan

Please hand in the assignment by 5:00pm May 31st.
Late Penalty -20% off for up to 72 hours late, zero after that.

Question 1: The Bareiss/Edmonds Algorithm

Reference: Ch. 9 of *Algorithms for Computer Algebra* by Geddes, Czapor and Labahn.

Part (a) (10 marks)

For an n by n matrix A with integer entries, implement ordinary Gaussian elimination and the Bareiss/Edmonds algorithms as the Maple procedures `GaussElim(A,n,'B');` and `Bareiss(A,n,'B');` to compute $\det(A)$. The algorithms should initially assign `B` a copy of the matrix A so that after the algorithm finishes and returns $\det(A)$ the value of B will be $A^{(n-1)}$. Note, you will need to take care of pivoting: if at any step k, the matrix entry $B_{k,k} = 0$ and $B_{i,k} \neq 0$ for some $k < i \leq n$, interchange row k with row i before proceeding. Remember, interchanging two rows of a matrix changes the sign of the determinant.

Use `iquo(a,b)` to compute the quotient of a divided by b. When you are debugging, print out the matrices $A^{(1)}$, $A^{(2)}$, ... after each step of the elimination.

Execute both algorithms on the following matrices for $n = 2, 3, 4, ..., 10$.

```
> n := 4;
```

```
n := 4
```

```
> m := 4:
> c := rand(10^m):
> A := Matrix(n,n,c);
```

$$A := \begin{bmatrix} 7926 & 8057 & 5 & 3002 \\ 2347 & 9765 & 3354 & 5860 \\ 6906 & 5281 & 5393 & 1203 \\ 311 & 9386 & 9810 & 5144 \end{bmatrix}$$

For $n = 4$ print out final triangular matrix for both algorithms.
Finally, in class we showed that $|\det(A)| < \sqrt{n}^n B^{mn}$ where $B = 10$ and $m = 4$ here.
Check this for $n = 4$.

Part (b) (10 marks)

Let F be a field, $D = F[x]$ and A be an n by n matrix over D. If we assume $\deg A_{i,j} \leq d$ and classical quadratic algorithms are used for polynomial multiplication and exact division in $F[x]$, how many multiplications in F does the Bareiss/Edmonds algorithm do?

Try to get an exact formula in terms of n and d assuming $\deg A_{i,j} = d$. I suggest you do this for a 3x3 matrix first. Recall that to divide a polynomial in $F[x]$ of degree d by a polynomial of degree $m \leq d$, the classical division algorithm does at most $(d - m + 1)m$ multiplications in F.

Part (c) (5 marks)

Let A be an n by n matrix with $n > 2$ and let B be the $n-2 \times n-2$ principal submatrix of A (rows 1 to $n-2$, columns 1 to $n-2$). At the last step (step $n-1$) of the Bareiss algorithm, assuming no pivoting occurred, we compute

$$A_{n,n}^{n-1} = \frac{A_{n-1,n-1}^{n-2} A_{n,n}^{n-2} - A_{n,n-1}^{n-2} A_{n-1,n}^{n-2}}{A_{n-2,n-2}^{n-3}}$$

Here $A_{n-2,n-2}^{n-3}$ is the determinant of B and $A_{n,n}^{n-1}$ is the determinant of A. Thus

$$A_{n-1,n-1}^{n-2} A_{n,n}^{n-2} - A_{n,n-1}^{n-2} A_{n-1,n}^{n-2} = \det A \det B.$$

The Bareiss method has an expression swell when applied to matrices of multivariate polynomials in that the number of terms in this polynomial when expanded can be much bigger than in $\det A$ the final answer. To investigate how big the numerator can be consider the symmetric Töplitz matrices T_n. Let B_n be the $n-2 \times n-2$ principle submatrix of T_n. Here is T_4 and B_4.

```
> with(LinearAlgebra):
> T4 := ToeplitzMatrix([x1,x2,x3,x4],symmetric);
```

$$T4 := \begin{bmatrix} x1 & x2 & x3 & x4 \\ x2 & x1 & x2 & x3 \\ x3 & x2 & x1 & x2 \\ x4 & x3 & x2 & x1 \end{bmatrix}$$

```
> B4 := T4[1..2,1..2];
```

$$\begin{bmatrix} x1 & x2 \\ x2 & x1 \end{bmatrix}$$

For $4 \leq n \leq 11$ compute the number of terms in $\det T_n, \det B_n$ and `expand`$(\det T_n \det B_n)$. How much bigger is $\det T_{11} \det B_{11}$ than $\det T_{11}$?

Question 2: Solving $Ax = b$ using p-adic lifting and rational reconstruction.

Part (a) (10 marks)

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)`. Use the prime $p = 2^{31} - 1$. Your 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 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];
> y := map( op, [y$5] );
> x := Vector(y);
> b := A.x;
> A,b := 12*A,12*b; # clear fractions
> x := padicLinearSolve(A,b);
> convert( A.x-b, set ); # should be {0}
```

To compute A^{-1} mod p use the Maple command `Inverse(A) mod p`.
To multiply A times a vector x use `A.x` in Maple.

For rational number reconstruction use the Maple command `iratrecon`. Note, if u is a vector of integers modulo m, `iratrecon(u,m)` will apply rational reconstructon to each entry in u separately.

Part (b) (5 marks)

Reference: Maximal Quotient Rational Reconstruction: An Almost Optimal Algorithm for Rational Reconstruction by M. Monagan. Available on course website.

In class I presented an Theorem of Guy, Davenport and Wang for rational number reconstruction. One good way to understand what a Theorem is saying is to first check that it's true on some examples. Checking a Theorem will often reveal the conditions under which the Theorem is true. For example, should it be $r_i \leq N$ or $r_i < N$.

Implement Wang/Guy/Davenport's rational number reconstruction algorithm as presented in class as the Maple procedure `RATRECON`(m, u, N, D). For a modulus $m > 0$ and input $0 \le u < m$ and integers N, D satisfying $N \ge 0$, $D > 0$ and $2ND < m$ run the extended Euclidean algorithm for input $r_0 = m, r_1 = u$ and output the first r_i/t_i satisfying $r_i \le N$ provided $\gcd(t_i, m)$=1 and $|t_i| \le D$, otherwise output FAIL. Do this by modifying my extended Euclidean algorithm code from the handout. I've put it on the course webpage so you don't have to type it in.

Run your algorithm on the following inputs

$$m = 13,\ u = i,\ N = 3,\ D = 2 \ \text{ for } \ 0 \le i < 13.$$

Now verify that all rationals n/d satisfying $|n| \le N$ and $d \le D$ are recovered and only those rationals are recovered (all other outputs are FAIL).

Part (c) (10 marks)

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 $O(f(m, n, L))$.

For two integers of m digits base B use the fact that the Euclidean algorithm is $O(m^2)$ and the extended Euclidean algorithm is also $O(m^2)$.

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 $L \in O(mn)$? Recall that we showed in class that the modular algorithm cost $O(mn^4 + m^2n^3)$ to solve $Ax = b$.