MATH 895, Assignment 2, Summer 2017

Instructor: Michael Monagan

Please hand in the assignment by 9:30am Friday June 2nd.
Late Penalty -20% off for up to 72 hours late, zero after than.
For Maple problems, please submit a printout of a Maple worksheet containing your Maple code and Maple output. Use any tools from the Maple library, e.g. `content(...)`, `Content(...) mod p`, `divide(...)`, `Divide(...) mod p`, `Eval(...) mod p`, `Interp(...) mod p`, `chrem(...)`, `Linsolve(A,b) mod p`, `Roots(f) mod p`, etc.

Question 1: Brown's dense modular GCD algorithm

REFERENCE: Section 7.4 of the Geddes text and Brown's original paper: On Euclid's algorithm and the computation of polynomial greatest common divisors. W. S. Brown, *Journal of the ACM* **18**(4), pp. 478–504, 1971. (see course webpage)

(a) (5 marks)

Let $a, b \in \mathbb{Z}[x]$, $g = \gcd(a, b)$, $\bar{a} = a/g$ and $\bar{b} = b/g$. For the modular GCD algorithm in $\mathbb{Z}[x]$ we said a prime p is *unlucky* if $\deg(\gcd(\bar{a} \bmod p, \bar{b} \bmod p))) > 0$ and a prime p is *bad* if $p|\mathrm{lc}(g)$. We apply Lemma 7.3 to identify the unlucky primes.

For $a, b, \in \mathbb{Z}[x_1, x_2, ..., x_n]$ we need to generalize these definitions for bad prime and unlucky prime and also define bad evaluation points and unlucky evaluation points for evaluating x_n. We do this using the lexicographical order monomial ordering. Let $g = \gcd(a, b)$, $a = \bar{a}g$ and $b = \bar{b}g$. Let's use an example in $\mathbb{Z}[x, y, z]$.

$$g = 5xz + yz - 1, \quad \bar{a} = 3x + 7(z^2 - 1)y + 1, \quad \bar{b} = 3x + 7(z^3 - 1)y + 1.$$

Let LC, LT, LM denote the leading coefficient, leading term and leading monomial respectively in lexicographical order with $x > y > z$. So in our example, $LT(a) = (5xz)(3x) = 15x^2z$, hence $LC(a) = 15$ and $LM(a) = x^2z$.

Let p be a prime. We say p is *bad prime* if p divides $LC(g)$ and p is an *unlucky prime* if $\deg(\gcd(\phi_p(\bar{a}), \phi_p(\bar{b})) > 0$ where deg here means total degree. Identify all bad primes and all unlucky primes for the example.

Suppose we have picked $p = 11$ and we evaluate at $z = \alpha \in \mathbb{Z}_{11}$. We think of a, b as elements of $\mathbb{Z}_p[z][x, y]$ with coefficients in $\mathbb{Z}_p[z]$. Define bad and unlucky evaluation points appropriately and identify them for example.

(b) (5 marks)

Prove the following modified Lemma 7.3 for $\mathbb{Z}[x_1, ..., x_n]$.

> Let $a, b \in \mathbb{Z}[x_1, ..., x_n]$ with $a \neq 0, b \neq 0$ and $g = \gcd(a, b)$. Let $LC(a)$ and $LM(a)$ denote the leading coefficient and leading monomial of a in lexicographical order with $x_1 > x_2 > ... > x_n$. Let p be a prime let $h = \gcd(\phi_p(a), \phi_p(b)) \in \mathbb{Z}_p[x_1, ..., x_n]$. If p does not divide $LC(a)$ then
>
> (i) $LM(h) \geq LM(g)$ and
> (ii) if $LM(h) = LM(g)$ then $\phi_p(g)|h$ and $h|\phi_p(g)$.

(c) (30 marks)

Implement the modular GCD algorithm of section 7.4 in Maple. Implement two subroutines, subroutine MGCD that computes the GCD modulo a sequence of primes (use 4 digit primes), and subroutine PGCD that computes the GCD at a sequence of evaluation points (use $0, 1, 2, ...$ for the evaluation points). Note, subroutine PGCD is recursive. Test your algorithm on the following example polynomials in $\mathbb{Z}[x, y, z]$. Use x as the main variable. First evaluate out z then y. Also test your algorithm on the problem from part (d).

```
> c := x^3+y^3+z^3+1; d := x^3-y^3-z^3+1;
> g := x^4-123454321*y*z^2*x^2+1;
> MGCD(c,d,[x,y,z]);
> MGCD(expand(g*c),expand(g*d),[x,y,z]);
> MGCD(expand(g^2*c),expand(g^2*d),[x,y,z]);

> g := z*y*x^3+1; c := (z-1)*x+y+1; d := (z^2-1)*x+y+1;
> MGCD(expand(g*c),expand(g*d),[x,y,z]);
> g := x^4+z^2*y^2*x^2+1; c := x^4+z*y*x^2+1; d := x^4+1;
> MGCD(expand(g*c),expand(g*d),[x,y,z]);
> g := x^4+z^2*y^2*x^2+1; c := z*x^4+z*x^2+y; d := z*x^4+z^2*x^2+y;
> MGCD(expand(g*c),expand(g*d),[x,y,z]);
```

Please make your MGCD procedure print out the sequence of primes it uses using `printf(" p=%d\n",p);` .

Please make your PGCD procedure print out the sequence of evaluation points α that it uses for each variable u using `printf("    %a=%d\n",u,alpha);`

You may use the `Content(...) mod p`, `Primpart(...) mod p`, `Interp(...) mod p` and `Divide(...) mod p` commands and `Gcd(...) mod p` for computing univariate gcds over $\mathbb{Z}_p$.

Note, procedures MGCD and PGCD on pages 307 and 309 in Chapter 7 of the Geddes text do not identify unlucky primes and unlucky evaluation points correctly.

(d) (5 marks) Below is Maple code for constructing two polynomials $A, B \in \mathbb{Z}[w, x, y, z]$ with a gcd G of total degree 2. The code runs Maple's gcd algorithm then runs two variations of the Euclidean algorithm which both use pseudo-division to avoid fractions. The first is Collin's reduced PRS (polynomial remainder sequence) and the second the Primitive PRS. Time them both.

To investigate the intermediate expression swell, for both algorithms, insert `printf` statements after the computation of the pseudo-remainder `pr` and also after division of `pr` by μ (the content) that print the following information:

(i) the degree of pr in each of w, x, y, z. The degree in x is decreasing (by what?) but the degree in w, y, z increases (by what?).

(ii) the length (in digits) of the largest integer coefficient (use `ilog10(maxnorm(pr))`).

(iii) the number of terms of pr (see `numterms`)

```
rp := proc(x,dx,X,D) local i;
   add( randpoly(X,dense,degree=D)*x^i, i=0..dx );
end:

G := randpoly([x,w,y,z],degree=2,dense):
A := rp(x,10,[w,y,z],1): A := expand(A*G):
B := rp(x, 9,[w,y,z],1): B := expand(B*G):

# Maple is using Zippel’s sparse modular gcd algorithm
st := time(): g := gcd(A,B); ZippelTime = time()-st;

# Collins’ reduced PRS
c := gcd( content(A,x,’a’), content(B,x,’b’) );
mu := 1:
while b <> 0 do
    pr := prem(a,b,x,’m’);
    divide(pr,mu,’pr’);
    a,b,mu := b,pr,m;
od:
g := c*primpart(a,x);

# Primitive PRS
c := gcd( content(A,x,’a’), content(B,x,’b’) );
while b <> 0 do
    pr := prem(a,b,x,’m’);
    if pr <> 0 then
       co := content(pr,x);
       divide(pr,co,’pr’); # make pseudo-remainder primitive
    fi;
    a,b := b,pr;
od:
g := c*primpart(a,x);
```

Question 2: Sparse Interpolation Algorithms

(a) (5 marks) Apply Ben-Or/Tiwari sparse interpolation to interpolate

$$f(w,x,y,u) = 101w^5x^3y^2u + 103w^3xy^3u^2 + 107w^2x^5y^2 + 109x^2y^3u^5$$

over $\mathbb{Q}$ using Maple. You will need to compute the integer roots of the $\lambda(z)$ polynomial and solve a linear system over $\mathbb{Q}$.

Now it is very inefficient to run the algorithm over $\mathbb{Q}$. Repeat the method modulo a prime p, i.e., interpolate f modulo p. Assume you know that $\deg f < 16$. Pick p suitably large so that you can recover all monomials of total degree $d \leq 15$. See the `Roots(...) mod p` and `Linsolve(...) mod p` commands.

(b) (5 marks) Suppose $f(w,x,y,z)$ is a polynomial in $\mathbb{Z}[w,x,y,z]$. Without first interpolating $f(w,x,y,z)$ is there a fast way to get a bound on the **total degree** of $f(w,x,y,z)$ using evaluation and univariate interpolation? State any assumptions you need.

(c) (5 marks)

REFERENCE (a copy is available on the course web page):
Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. *Proc. STOC '88*, ACM press, 301-309, 1988.

The Ben-Or/Tiwari sparse interpolation algorithm interpolates a polynomial $f(x_1, x_2, \ldots, x_n)$ in two main steps. First it determines the monomials then it solves a linear system for the unknown coefficients of the polynomial. Let

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{t} a_i M_i$$

where a_i are the coefficients and M_i are the monomials. Let $a = [a_1, a_2, \ldots, a_t]$ be the vector of unknown coefficients. Let $v = [v_0, v_1, \ldots, v_{t-1}]$ be the vector of values where $v_j = f(2^j, 3^j, 5^j, \ldots, p_n^j)$. Let $m_i = M_i(2, 3, 5, \ldots, p_n)$ be the value of the monomial M_i. The linear system to be solved is $V^T a = v$ where

$$V^T = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ m_1 & m_2 & m_3 & \ldots & m_t \\ m_1^2 & m_2^2 & m_3^2 & \ldots & m_t^2 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ m_1^{t-1} & m_1^{t-1} & m_1^{t-1} & \ldots & m_t^{t-1} \end{bmatrix}$$

is a transposed Vandermonde matrix. Solve this linear system for the problem in part (a) using the $O(t^2)$ method.