MATH 895, Assignment 2, Summer 2013

Instructor: Michael Monagan

Please hand in the assignment by Thursday June 6th before class starts.
Late Penalty -20% off for up to 24 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 the original paper.

(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 $\mathbb{Z}_p[z]$. Identify all bad evaluation points and unlucky evaluation points for z in the 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.

```
> 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);`

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.

Question 2: Sparse Interpolation Algorithms

(a) (10 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.

Is there any way to get a bound on the total degree of $f(w,x,y,z)$ using evaluation and univariate interpolation?

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.

(b) (optional, bonus, 10 marks)

Prove the Schwartz-Zippel Lemma by induction on n the number of variables.

> Let K be a field and f be a non-zero polynomial in $K[x_1, x_2, ..., x_n]$ of total degree $d \geq 0$ and let S be any non-empty finite subset of K. If $\alpha_1, \alpha_2, ..., \alpha_n$ are chosen at random from S then
>
> $$\text{Prob}(f(\alpha_1, \alpha_2, ..., \alpha_n) = 0) \;\leq\; \frac{d}{|S|}.$$

(c) (optional, bonus, 10 marks)

Modify subroutine PGCD to use Zippel's sparse interpolation.
REFERENCE: Section 7.5 of the Geddes text.
For simplicity, assume that the gcd g is monic in x_1. Run both your sparse algorithm and dense algorithm on the following input. Count the number of univariate gcd computations in $\mathbb{Z}_p[z]$ that each algorithm does.

```
> g := 2*x^8 + (u^8*v - 3*v^8*y + y^8*u)*x^4 + (w^8*z - 3*z^8*w + 1);
> c := 4*x^8 + 5*w^4*x^4 + 2*y^4*z^4 + 3*u^4*v^4 + 1;
> d := 6*x^8 - 5*y^4*x^4 - 4*u^4*v^4 - 3*w^4*z^4 - 2;
> a := expand(g*c):
> b := expand(g*d):
> PGCD(a,b,[x,u,v,w,y,z],p);
```

Note, to get random numbers from $\mathbb{Z}_p$ first create a random number generator for $[0,p)$ using `r := rand(p);` then use `alpha := r();` to get a random number.