Computer Algebra

Lino Demasi

April 6, 2009

1 Introduction

- How do we represent long integers on a computer?

  Write an integer $a = \sum_{i=0}^{n-1} a_i B^i$ where B is the base of the representation.

  Maple versions < 9 used $B = 10^4$ on a 32 bit computer. $a = 123456789$ would be written as $1 \cdot 10^8 + 2345 \cdot 10^4 + 6789$ and stored in an array as

$POS4$	6789	2345	1

  Maple versions ≥ 9 use GMP package

$n = 4$	*pointer*

$32bits$	$32bits$	$32bits$	$32bits$

  GMP is coded in assembly, whereas older versions are coded in C.

- How do we multiply long integers?

  Let $a = \sum_{i=0}^{m-1} a_i B^i$, $b = \sum_{i=0}^{n-1} b_i B^i$

  Classical algorithm:

  $$\begin{array}{r} 345 \\ 73 \\ \hline 1035 \\ 24150 \\ \hline 25185 \end{array}$$

  Runs in $O(nm)$

  Maple algorithm:

  $t < B^2$. Running time is $< k_1 mn + k_2 n + k_3(m+n) + k_4 = O(mn)$.

- How fast can we multiply two n digit base B integers?

- Classical $O(n^2)$
- Karatsuba (1962) $O(n^{1.585})$
- Schonhage Strassen (1978) FFT $O(n \log n \log\log n)$

Karatsuba's Algorithm

$a = a_i B^{n/2} + a_2$

$b = b_i B^{n/2} + b_2$

$a \cdot b = a_1 b_1 B^n + (a_1 b_2 + b_1 a_2) B^{n/2} + a_2 b_2$

Cleverly calculate $a_1 b_2 + b_1 a_2$ as $(a_1 + a_2)(b_1 + b_2) - a_1 b_1 - a_2 b_2$

Cleverly calculate $a_1 b_2 + b_1 a_2$ as $(a_1 - a_2)(b_2 - b_1) + a_1 b_1 + a_2 b_2$

- How fast can we compute $\gcd(a, b)$?
  - Euclid's Algorithm (˜300 BC) $O(n^2)$
  - Stein's Binary GCD Algorithm $O(n^2)$
  - Lehmer's Algorithm $O(n^2)$
  - Schonhage Strassen $O(\log(n) M(n))$ where $M(n)$ is the cost of multiplication.
- How fast can we divide $c/b = a$?

  Classical long division is $O(mn)$ V.P. Newton iteration is $O(mn)$
- Can we compute gcd(a,b) in $O(M(n))$?

  Binary GCD Algorithm

  Suppose $a = \sum_{i=0}^{m-1} a_i B^i$ and $B = 2^k$

  We can divide a by powers of 2 easily by shifting bits.

  Knuth showed that if we take large a, random $0 < b < a$ then the probability that a quotient of x occurs at a given step of the Euclidean Algorithm is$\log_2(1 + \frac{1}{x(x+2)})$

  Lehmer's Algorithm

  Run Euclid's Algorithm on the leading 64 bits of a, b. On average we will get ˜20 quotients, $q_1, q_2, \cdots q_{20}$ correct.

  Observation

  $\begin{vmatrix} 0 & 1 \\ 1 & -q_{i+1} \end{vmatrix} \begin{vmatrix} r_{i-1} \\ r_i \end{vmatrix} \begin{matrix} = \\ = \end{matrix} \begin{vmatrix} r_i \\ r_{i+1} \end{vmatrix}$

  $A_{20} \cdot A_{19} \cdots A_1 \frac{r_0}{r_1} = \frac{r_2 0}{r_2 1}$ (fractions are vectors)

2 Review

2.1 Rings, Fields, and Integral Domains

A non-zero element a in a ring R is a zero-divisor if there exists a non-zero element $b \in R$ s.t. $a \cdot b = 0$. Consider $Z_6 = \{0, 1, 2, 3, 4, 5\}$ the ring of integers mod 6.

x	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Lemma. *In a ring R a unit cannot be a zero divisor.*

Proof. Suppose $u \in R$ is a unit and a zero divisor.
Then $\exists v \neq 0 : uv = 0, \exists w : uw = 1$. $v = 1v = wuv = w0 = 0$. □

Corollary. *A field has no zero-divisors*

Definition. A commutative ring D is an integral domain if D has no zero divisors.

Examples Z, fields, $D[x]$

Proof. Let a, b be non-zero polynomials in $D[x]$. $a \cdot b = a_n \cdot b_m \cdot x^{m+n} + \ldots \neq 0$ □

Definition. Cancellation Law: Let a, b, c be non-zero elements of a ring R. If $ab = ac \Rightarrow b = c$ then we say the cancellation law holds in R.

Lemma. *In an integral domain D, the cancellation law holds.*

Proof. Let a, b, c be non-zero elements in D s.t.

$$\begin{aligned} & ab = ac \\ \Rightarrow\ & ab - ac = 0 \\ \Rightarrow\ & a(b - c) = 0 \\ \Rightarrow\ & (b - c) = 0 \\ \Rightarrow\ & b = c \end{aligned}$$

□

2.2 Divisibility and Factorization in Integral Domains

Let D be an integral domain. A non-zero element $b \in D$ is a divisor of $a \in D$ if $\exists q \in D$ st. $a = bq = qb$.

If b is a divisor of a, we write $b|a$. How do we simplify $\frac{a}{b}$ where $a, b \in D$?

Definition. Let $a, b, c, d \in D$. c is the greatest common divisor of a and b if

- $c|a$ and $c|b$
- $d|a$ and $d|b \Rightarrow d|c$

Definition. Two elements $a, b \in D$ are associates if $a|b$ and $b|a$. We write $a \sim b$.

Lemma. *Let $a, b \in D$. Then $a \sim b \Leftrightarrow a = ub$ for some unit $u \in D$.*

Proof. ($\Rightarrow$) $a \sim b \Rightarrow b = ac, a = bd$ for some $c, d \in D$. Combining we get $cd = 1$, which means c, d are units.

($\Leftarrow$) Since $a = ub$, we have $b = au^{-1}$. These tell us that $a|b$ and $b|a$, so we are done. □

Theorem. *In an integral domain D the relation $a \sim b$ is an equivalence relation on the non-zero elements of D.*

- $a \sim a$
- $a \sim b \Rightarrow b \sim a$
- $a \sim b$ *and* $b \sim c \Rightarrow a \sim c$

Definition. Let $n : D \backslash \{0\} \to D$ be a function st. $n(a)$ returns a "canonical" or "standard" representative from the associate class with a in it.

We impose uniqueness on $g = \gcd(a, b)$ by returning $n(g)$.

2.3 Unique Factorization Domains

Let R be a commutative ring and let p be a non-zero, non-unit of R.

Definition. p is a prime if $p|ab \Rightarrow p|a$ or $p|b$

Definition. p is irreducible if $p = ab \Rightarrow a$ or b is a unit.

Lemma. *In an integral domain D, primes are irreducible*

Proof. Suppose $p = ab$ then $p|ab$ so $p|a$ or $p|b$. Assume $p|a$, so $p = pcb \Rightarrow cb = 1$. Thus b is a unit. $\square$

Definition. An integral domain D is a unique factorization domain (UFD) if $\forall a \in D$ s.t. $a \neq 0$ and a is not a unit

- $a = c_1 \cdot c_2 \cdots c_n$ for some irreducibles $c_i \in D$
- if $a = c_1 \cdot c_2 \cdots c_n = d_1 \cdot d_2 \cdots d_m$ where c_i, d_j are irreducibles, then $m = n$ and $c_i \sim d_j$ for some j.

Examples of integral domains which are not UFDs.

- $D = Q[s, c]/(s^2 + c^2 - 1)$

  $(1 - c)(1 + c) = 1 - c^2 = s \cdot s$

  These are both products of irreducibles and not associates.

  Consider $a = s + sc, b = s + sc + s^2$. $s, 1 + c$ are common divisors of a, b but neither is a greatest common divisor.

2.4 Euclidean Domain

Definition. An integral domain E is called a Euclidean Domain if there $\exists v : E \backslash \{0\} \rightarrow N \cup \{0\}$ such that

1. $\forall a, b \in E \backslash \{0\} v(a, b) \geq v(a)$
2. $\forall a, b \in E, b \neq 0, \exists q, r \in E$ s.t. $a = bq + r$ and $r = 0$ or $v(r) < v(b)$

Definition. The function v is called the valuation or the Euclidean norm.

Examples of Euclidean Domains

- Z is a Euclidean Domain with $v(a) = |a|$
- $F[x]$ for F a field with $v(a) = deg(a)$.
- Gaussian Integers $Z[i]$

Theorem. *Euclidean domains are UFDs.*

Theorem. *Let $a, b \in E \backslash \{0\}$, where E is a Euclidean Domain, then $\exists s, t \in E$ s.t. $sa + tb = g$ where $g = \gcd(a, b)$.*

```
$r_0, r_1 \leftarrow a, b$
$s_0, s_1 \leftarrow 1, 0$
$t_o, t_1 \leftarrow 0, 1$
$k \leftarrow 1$
while $r_k \neq 0$ do
   $q_{k+1} \leftarrow$ quotient of $r_{k-1}/r_k$
   $r_{k+1} \leftarrow r_{k-1} - q_{k+1} r_k$
   $s_{k+1} \leftarrow s_{k-1} - q_{k+1} s_k$
   $t_{k+1} \leftarrow t_{k-1} - q_{k+1} t_k$
   $k \leftarrow k + 1$
end while
$n \leftarrow k - 1$
output $r_n, s_n, t_n$
```

Claim: $s_k a + t_k b = r_k$ for all k

Proof. By double induction on k.

$$\begin{array}{lll} k = 0 & s_0 a + t_0 b = r_0? & 1a + 0b = a \\ k = 1 & s_1 a + t_1 b = r_1? & 0a + 1b = b \end{array}$$

For $k \geq 1$, assume holds for $k, k-1$. We will show true for $k+1$.

By algorithm,

$$\begin{aligned} s_{k+1} a + t_{k+1} b &= (s_{k-1} - q_{k+1} s_k) a + (t_{k-1} - q_{k-+} t_k) b \\ &= (s_{k-1} a + t_{k-1} b) - q_{k+1}(s_k a + t_k b) \\ &= r_{k-1} - q_{k+1} r_k \\ &= r_{k+1} \end{aligned}$$

□

2.5 Computing inverses in Z_m

Let $a \in Z_m$ with $m > a \geq 0$. Applying the EEA we obtain $s_n t_n \in Z$s.t.$s_n m + t_n a = r_n$.

If $g > 1$ then output a is not invertible. Otherwise $t_n a = r_n = 1($ mod $m)$.

But t_n can be negative. How big can t_n be? $|t_n| < m/g$.

2.6 Rational Number Reconstruction

Suppose $\frac{n}{d} \in Q$ with $\gcd(n, d) = 1$ and $d > 0$. Suppose $m > 0$ with $\gcd(m, d) = 1$. Suppose we have computer $u = \frac{n}{d}$ mod m and we want to recover $\frac{n}{d}$ given u, m.

Apply $EEA(m,u)$ to get $s_k m + y_k u = r_k$ for $k = 0, 1, \ldots, n+1$.
$t_k u = r_k (\mod m) \Rightarrow u = \frac{r_k}{t_k}$

Lemma. [*Wang, Guy, Davenport (1982)*] *If $m > 2|n|d$ then $\frac{n}{d} = \frac{r_k}{t_k}$ for some k*

Lemma. [*Monagan (2004)*] *If $m > (2|n|d)^2$ then if i satisfies $|r_i t_i| \leq |r_k t_k| \forall k$ then $\frac{n}{d} = r_1 t_i$*

In a Euclidean Domain E we have a function $v : E \backslash \{0\} \rightarrow N \cup \{0\}$ such that:

1. $\forall a, b \in E \backslash \{0\} v(a,b) \geq v(a)$
2. $\forall a, b \in E, b \neq 0, \exists q, r \in E$ s.t. $a = bq + r$ and $r = 0$ or $v(r) < v(b)$

Let u be a unit in E and let $c, d \in E$ which are not zero and not units.

1. $v(u) = v(1) \forall$ units u
2. $v(c) > v(u) \forall$ non-units c
3. $v(uc) = v(c)$
4. $v(cd) > v(c)$
5. $c|d$ and $d|c \Rightarrow v(c) = v(d)$
6. $v(c) < v(d) \Rightarrow d \not| c$

Proof. (1)

$$\begin{aligned}
&v(u) = v(1u) \geq v(1) \\
&v(1) = v(uu^{-1}) \geq v(u) \\
&\Rightarrow v(u) = v(1)
\end{aligned}$$

□

Proof. (2) We can write $1 = cq + r$. If $r = 0$ then c would be a unit. Thus $v(c) > v(r) = v(1r) \geq v(1)$. □

Definition. Let R be a ring. $R[x]$ dentoes the set of univariate polynomials in the variable x whose coefficients are in the ring R.

Let $a = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ where $a_n \neq 0$.

Definition. The degree of a wrt x is n. $\deg(0) = -\infty$

Definition. The leading coefficient of a wrt x is a_n. Denoted $lt(a)$.

Definition. The leading term of a wrt x is $a_n x^n$. Denoted $lc(a)$.

Definition. The leading monomial of a wrt x is x_n. Denoted $lm(a)$.

In maple, $degree(a), degree(a, x), lcoeff(a), lcoeff(a, x)$ do the expected things.

```
lterm := proc(a,x)
   lcoeff(a,x) * x^degree(a,x)
end;

lterm := proc(a,x) local c,m;
   c:= lcoeff(a,x,'m');
   c*m;
end
```

Theorem. *If R is an integral domain a, b non-zero polynomials in $R[x]$ then*

- $deg(ab) = deg(a) + deg(b)$
- $lc(ab) = lc(a) \cdot lc(b)$
- $lt(ab) = lt(a) \cdot lt(b)$
- $lm(ab) = lm(a) \cdot lm(b)$

2.7 Divison in $F[x]$, F a field

Theorem. *Let F be a field. Let $a, b \in F[x], b \neq 0$. There exist unique $q, r \in F[x]$ s.t. $a = bq + r$ with $r = 0$ or $deg(r) < def(b)$.*

Proof. (of uniqueness) Suppose we have that $a = bq + r_1$ and $a' = bq + r_2$ where $q_1, q_2, r_1, r_2 \in F[x]$ and ($r_1 = 0$ or $deg(r_1) < deg(b)$) and ($r_2 = 0$ or $deg(r_2) < deg(b)$)

$$\begin{aligned}
0 = bq_1 + r_1 - (bq_2 + r_2) &\Rightarrow r_2 - r_1 = b(q_1 - q_2) \\
&\Rightarrow b|(r_2 - r_1) \\
&\Rightarrow r_2 = r_1 \\
0 = b(q_1 - q_2) &\Rightarrow q_1 - q_2 = 0 \\
&\Rightarrow q_1 = q_2
\end{aligned}$$

□

```
r:= 0;
q:= 0;
while deg(r) \geq deg(b) do
   t := lt(r)/lt(b);
   q:= q + t;
   r := r - t*b;
end;
return q,r;

q:= quo(a,b,x);
r:= rem(a,b,x);
g:= gcd(a,b,x);
g:= gcd(a,b,x,'s','t');

q:= Quo(a,b,x) mod p;
q:= Rem(a,b,x) mod p;
q:= Gcdex(a,b,x,'s','t') mod p;
```

Theorem. *Let $a, b, c \in F[x], a, b \neq 0$. Let $g = gcd(a, b)$. If $g|c$ then there exist unique polynomials $\sigma, \tau \in F[x]$ s.t.*

- $\sigma a + \tau b = c$
- $deg(\sigma) < deg(\frac{b}{g}) = deg(b) - deg(g)$
- *If* $deg(c) < deg(\frac{ab}{g})$ *then* $deg(\tau) < deg(\frac{a}{g})$

Proof. (of existence) From the EEA, $\exists s, t \in F[x]$ s.t. $sa + tb = g$.
$g|c \Rightarrow c = gd \Rightarrow dsa + dtb = c$.
$\Rightarrow (ds)\frac{a}{g} + (dt)\frac{b}{g} = \frac{c}{g}$

$$\begin{aligned} ds = \tfrac{b}{g}q + \sigma \quad &\Rightarrow \quad (\tfrac{b}{g}q + \tau)\tfrac{a}{g} + (dt)\tfrac{b}{g} = \tfrac{c}{g} \\ &\Rightarrow \quad \sigma\tfrac{a}{g} + (dt + q\tfrac{a}{g})\tfrac{b}{g} = \tfrac{c}{g} \\ &\Rightarrow \quad \sigma a + [dt + q\tfrac{a}{g}]b = c \end{aligned}$$

(of uniqueness) Suppose that $\sigma_1 a + \tau_1 b = c$ and $\sigma_2 a + \tau_2 b = c$. Proof follows as the previous uniqueness proof.

□

2.8 Multivariate Polynomials

Let R be a ring. $R[x_1, x_2, \ldots, x_n] = \{\sum_{i=1}^{t} a_i x_1^{e_{i1}} x_2^{e_{12}} \cdots x_n^{e_{in}} : a_i \in R, A_i \neq 0, e_{ij} \in N \cup 0\}$ with addition and multiplication defined in the usual way is a multivariate polynomial ring.

The vectors $[e_{i1}, e_{i2}, \ldots e_{in}]$ are called exponent vectors.

Definition. The degree of a is $\max_{i=1}^{t} \sum_{j=1}^{n} e_{ij}$.

Two different ways of organizing terms:

- Pure lexocographical ordering - order wrt x_1, then x_2, etc.
- Grades lexocographical ordering - order by total degree, among each degree order by pure.

Theorem. *If R is an integral domain and a, b are non-zero polynomials in $R[x_1, x_2, \ldots x_n]$ then for either of the above orderings the properties of degrees, leading coefficients, leading terms and leading monomials are true.*

The ring $R[x_1, x_2, \ldots, x_n$ is isomorphic to the ring $R[x_2, x_3, \ldots x_n][x_1]$. So we can write $R[x_1, \ldots, x_n] \equiv R[x_n] \ldots [x_1]$.

2.9 The primitive Euclidean Algorithm

In $Z[x]$

$$\begin{aligned}
&a = 4x^2 + 12x + 8 = 4(x^2 + 3x + 2)\\
&b = 12x^2 + 6x + 6 = 6(2x^2 + x + 1)\\
&\gcd(a, b) = \gcd(4, 6) \cdot \gcd(x^2 + 3x + 2, 2x^2 + x + 1)
\end{aligned}$$

Definition. 4 is the content of a, $(x^2 + 3x + 2)$ is the primitive part.

In $Z[x, y] \equiv Z[y][x]$

$$\begin{aligned}
&a = 6x^2y - 6y^3 = (6y)x^2 - 6y^3 = 6y(x^2 - y^2)\\
&b = 9xy + 9y^2 = (9y)x + 9y^2 = 9y(x + y)\\
&\gcd(a, b) = \gcd(9y, 6y) \cdot \gcd(x^2 - y^2, x + y) = 3y(x + y)
\end{aligned}$$

Let R be a UFD (gcds exist). Let $a = a_n x^n + \cdots + a_1 x + a_0$.

Definition. The content of a, denoted $cont(a)$ is $\gcd(a_n, a_{n-1}, \ldots, a_0)$

Definition. The primitve part of a, denoted $pp(a) = \frac{a}{cont(a)}$.

Definition. We say a is primitive if $cont(a) =$ a unit.

Lemma. *$gcd(a, b) = gcd(cont(a), cont(b)) \cdot gcd(pp(a), pp(b))$*

$content(a, x), content(a, x,' p'), primpart(a, x)$, and $primpart(a, x,' c')$ are the maple commands to get the desired part(s).

Pseudo-Division in $D[x]$, D an integral domain. Let $a, b \in D[x], b \neq 0$. In general, he quotient and reaminder $\notin D[x]$. However, the fractions we obtain are all powers of the leading coefficient of the divisor. So we can multiply the expression by that power.

Theorem. *Let $a = \sum^l a_i x_i, b = \sum^n b_j x^j$. There exist unique polynomials $\bar{q}, \bar{r}$ (called the pseudo quotient and pseudo remainder in $D[x]$) s.t. $ma = b\bar{q} + \bar{r}$ where $deg(\bar{r}) < deg(b)$ and $m = k(b)^{l-n+1}$*

In maple $prem(a, b, x), prem(a, b, x,' m',' q')$

Lemma. *Let a, b be non-zero primitive in $D[x]$. Let $\bar{r}, \bar{q} \in D[x]$ s.t. $ma = b\bar{q} + \bar{r}$ with $\deg(\bar{r}) < \deg(b)$. Then $\gcd(a, b) \equiv \gcd(pp(\bar{r}), b)$*

Proof. Let $g = gcd(a, b), h = gcd(pp(\bar{r}), b)$. We must show $g|h$ and $h|g$. $g|a, g|b$ so $g|\bar{r}$. Since a, b are primitive, g is also primitive and thus $g|pp(\bar{r})$. $h|b, h|\bar{r}$ so $h|ma$. Again h is primitive, so $h|a$. □

2.10 Algorithm - Primitive Euclidean Algorithm

Let $a, b \in R[x_1, x_2, \ldots, x_n]$, R a UFD, $a, b \neq 0$. Output $gcd(a, b)$. Needs no factorization.

```
1)  If $n = 0$ then output $gcd_R(a, b)$.
2)  Write $a, b \in R[x_1, x_2, \ldots x_{n-1}][x_n]$.
3)  $c := \gcd(cont(a), cont(b))$
4)  primitive polynomial pseudo-remainder sequence
    $r_0 = a/cont(a)$
    $r_1 = b/cont(b)$
    $k = 1$
    while($r_k \neq 0$) do
      $r_{k+1} = pp(prem(r_{k-1}/r_k))$
      $k \leftarrow k + 1$
    end
    output $c * r_{k-1}$
```

3 Normal Forms and Algebraic Representations

3.1 LEvels of Abstraction

- The object (mathematical) level.

$$\begin{array}{l} D[x] \text{ an int. dom } \Rightarrow +, -, *, a|b, =, \\ \text{arithmetic in } D \text{ and with monomials} \\ \text{other polynomial operations } \deg, lc, lt, prem \end{array}$$

- The form (simplification) level

$$\begin{array}{rl} Z[x,y] & \text{distributed form } 12x^2y - 4xy - 9x + 3 \\ & \text{recursive form } (12y)x6x^2 + (-4y - 9)x + 3 \\ & \text{factored form } (3x-1)(4xy-3) \\ Q(x) & \text{expanded form } \frac{-x^4}{x^3-1} \\ & \text{factored form } -1x^4(x-1)^{-1}(x^2+x+1)^{-1} \\ & \text{factored pf form } -x + \frac{\frac{1}{3}}{x^2+x+1} + \frac{-\frac{1}{3}}{x-1} \end{array}$$

- Data structure level

3.2 Normal and Cannonical Forms

LEt E be a set with the relation $a \equiv b \iff a - b = 0$. Then $\equiv$ is an equivalence relation on E. So it partitions E into equivalence classes.

For simplification we want a function $f : E \to E$ s.t.

1. $f(a) \equiv a$
2. $f(a)$ is simpler than a.

Definition. A normal function $f : E \to E$ satisfies

- $f(a) \equiv a$
- if $a \equiv 0$ then $f(a) = 0$.
- $f(f(a)) = f(a)$

An element is in normal form if $f(a) = a$.

Definition. $f : E \to E$ is in cannonical form if $a \equiv b \Rightarrow f(a) = f(b)$.

Maple's simplify function gives a normal form for $Q(x)$. Factor gives a cannonical form for $Q(x)$. Expans does not give a normal form.

If $a \equiv b$ to test if $a \equiv b$ using a normal function, $f : E \rightarrow E$ we use $f(a - b) = 0$.

How do we reprsent a formula on the computer?

Maple uses a sum of products representation.

Programming using the op/nops form model.

$nops(a) = 3$ because there are 3 operands.

$op(0, a) =$ ‘ + ‘ if op(0,a) = ‘+‘ then ...

$op(1, a) =$ first term

If $type(a,$ ‘ + ‘$)$ then

$b := op(1, a) 3xy^3$

$op(0, b) =$ ‘ $*$ ‘

$c := op(3, b)$

$op(0, c) =$ ‘‘

integer	rational	numeric	algebraic
fraction	rational	numeric	algebraic
float		numeric	algebraic
symbol	name	algebraic	
indexed	name	algebraic	
‘+‘			algebraic
‘*‘			algebraic
‘ˆ‘			algebraic
function			algebraic
set			
list			
‘=‘			

Forward quotes represent unevaluated.

3.3 Data Structures for Polynomials

Consider the polynomial $3x^3y + 5x^2 - 7y^4$.

- Maple

SUM 7	·	3	·	5	·	-7

PROD 5	·	3	·	1

NAME 3	x	ϕ

In total this takes 18 words to store. This structure is

  - distributed

  - no monomial ordering
  - sparse
  - slow arithmetic

- Pari - designed for computational number theory

  $(3y+5)x^3 + (-7)y^4$

POLY	·	·

2	x	y

3	·	ϕ	ϕ	·

4	0	0	0	0	-7

1	5	3

  - dense
  - recursive
  - variables out
  - monomials implicit
  - sorted

- Axiom

  Distributed multivariate system.

- Spares Distributed Multivariate Polynomials Monagan and Pearce.

4 Chinese Remainder Theorem

4.1 Homomorphisms

Definition. Let R, S be two rings wth identity. A function $\phi : R \to S$ is called a homomorphism if

- $\phi(a+b) = \phi(a) + \phi(b)$
- $\phi(ab) = \phi(a)\phi(b)$
- $\phi(1) = 1$

The homomorphisms for evaluating a polynomial and eonsidering the elements mod n commute.

We can apply this to determining whether a matrix is singular. We select randomly some values for the polynomial at random, and having a random prime. We evaluate a few times and if we don't get zero for one then it is non-singular. If it is always zero then the matrix is likely singular.

4.2 The integer chinese remainder theorem

Given pairwise relatively prime integers $m_1, m_2, \ldots m_n$ and integers $u_1, u_2, \ldots u_n$, find $u \in Z$ such that $u \equiv u_i \mod m_i$.

Theorem. *There exists a unique solution to the congruences satisfying* $u < m_1 m_2 \cdots m_n$.

Proof. Uniqueness: Easy.

Existence: algorithmic.

We can find v_is such that $u = \sum_i v_i \frac{M}{m_i} \mod M$. This generates $u = v_i \frac{M}{m_i} \mod m_i$ which is computationally intensive to solve.

Mixed Radix Represntation: We can find w_is such that $u = w_0 + w_1 m_1 + w_2 m_1 m_2 + \cdots w_{n-1} m_1 m_2 \cdots m_{n-1}$. We solve this reursively. $\square$

We prefer mixed-radix as it is easier to solve computationally.

In maple, *chrem*$([u_1, \ldots, u_n], [m_1, \ldots, m_n])$ will solve. Works on polynomials as well.

Let $a, b \in Z[x]$ and let $c = ab$. How fast can we compute c? Suppose $a = \sum_{i=0}^{n-1} a_i x^i$, $b = \sum_{i=0}^{n-1} b_i x^i$. Assume each coefficient is $< B^m$.

We need n^2 multiplications which take $O(m^2)$ time each. We could instead do Karatsuba and get $O(m^1.585)$ for the multiplications. We could also do Karatsuba on the polynomials and only use $O(n^1.585)$ multiplications.

Using the chinese remainder theorem we can do this in $O(n^2 m + m^2 n)$.

Let $\phi_p : Z - Z_p$ be $\phi(a) = a \mod p$. We can calculate $\phi_p(c)$ for many primes and then combine using the chinese remainder theorem. Should we use one large prime or many small primes? IF we use machine primes, then arithmetic will be easy and use $O(n^2)$ machine instructions.

Definition. The height of a polynomial is the largest coefficient of the polynomial.

$a = 3x - 4, b = 6x + 5$. height of c is at most 48, so we need a product of primes that is at least 96. We choose 3,5,7.

	$\phi_p(a)$	$\phi_p(b)^{\phi}_{p}(c)$	
5	$3x+1$	x	$3x^2+x$
7	$3x+3$	$6x+5$	$4x^2+5x+1$
3	2	2	4

$c = 18x^2 + 96x + 85 \mod 105$. So $c = 18x^2 - 9x - 30$.

Analysis of the modular multiplication algorithm.

If we use machcine primes, how many primes do we need? If we assume that the primes are at least B, then we need at most $2m$ primes.

The cost of calculating our polynomials $\mod p$ is $2nO(m)O(m) = O(nm^2)$. The cost of the multiplications in Z_{p_i} is $O(m)O(n^2)$. The cost of the chinese remainder theorem is $(2n)O(m^2) = O(nm^2)$.

Let F be a field. Given n distinct points $\alpha_1, \alpha_n \in F$ an values $y_1, y_2, \dots y_n \in F$ find a polynomial $f(x) \in F[x]$ s.t. $f(\alpha_i) = y_i$.

Theorem. *There exists a unique polynomial with degree $< n$.*

We can solve this by considering a general polynomial and using gaussian elimination. This takes $O(n^3)$ time. To show uniqueness, we consider the polynomial $f - g$ which has n roots but is of degree at most $n-1$.

Lagrange Interpolation. $L(x) = (x-\alpha_1)(x-\alpha_2)\cdots(x-\alpha_n)$. $f(x) = \sum_{i=1}^{n} a_i \frac{L(x)}{x-\alpha_i}$. $a_i = \frac{y_i}{L_i(\alpha_i)}$.

Newton Interpolation. Let $f(x) = b_0 + b_1(x-\alpha_1) + b_2(x-\alpha_1)(x-\alpha_2)\cdots + b_{n-1}(x-\alpha_1)\cdots(x-\alpha_{n-1})$. $b_0 = y_1$. $b_1 = \frac{y_2-b_0}{\alpha_2-\alpha_1}$.

Both algorithms require $O(n^2)$ arithmetic operations in the field. Newton is a constant time faster than Lagrange and is easily extended to higher degree polynomials.

In Maple, over Q, we use $interp([\alpha],[y],x)$. Interp will work over Z_p.

When considering multivariate polynomials, we have the polynomial $a(x) + yb(x) + y^2c(x) + \cdots$ and we interpolate each polynomial seperately.

Evaluation of $f(x)$ of degree $\leq n-1$ at $x = \alpha$. $f = a_0 + a_1x + \cdots = a_0 + x(a_1 + x(\cdots))$. Which takes $O(n)$ mults and $O(n)$ adds.

4.3 Fast Fourier Transform

Let $a(x) = \sum a_i x^i$ with $a_i \in F$. How fast can we evaluate $a(x)$ and interpolate $a(x)$?

We can write the polynomial as the vector $[a_0, a_1, \dots, a_n]$. We can also store the polynomial as $[a(1), a(2), \dots, a(n)]$ without loss of information. To interpolate the points takes $O(n^2)$ time and to do the evaluations each take $O(n)$ time and we need n of them, so the time is also $O(n^2)$.

$$\begin{aligned} a(x) = &\ [a_0 + x^2 a_2 + \cdots] + x[a_1 + a_3 x^2 + \cdots] \\ a(x) = &\ b(x^2) + xc(x^2) \end{aligned}$$

We evaluate the polynomial at $\pm 1, \pm 2, \ldots, \pm \frac{n}{2}$. We can further break this down.

$$a(x) = \ d(x^4) + x^2 e(x^4) + x[f(x^4) + x^2 g(x^4)$$

We can evalute this at $\pm 1, \pm i, \pm 2, \pm 2i, \ldots, \pm \frac{n}{4}, \pm \frac{n}{4} i$.

In general, we want $n = 2^k$ points such that $\omega_i^n = 1$.

Definition. An element is an nth root of unity if $\omega^n = 1$. ω is a primitive nth root of unity if $\omega^n = 1$ and $\omega^k \neq 1, \forall k < n$. If ω is primitive, $\{\omega, \omega^2, \ldots, \omega^n\}$ are called the fourier points.

Definition. If $a(x) \in F[x]$ of degree $< n$, then $[a(1), a(\omega), \ldots, a(\omega^{n-1})] \in F^n$ is the Fourier Transform of $a(x)$.

In Z_{13}, 8 is a primitive 4th root of unity, so $[1, 8, 12, 5]$ are fourier points.

Input $n = 2^k$, $a = [a_0, a_1, \ldots, a_{n-1}] \in F^n$, $\omega \in F$ a primitive nth root of unity.

```
Output $[a(\omega^0), a(\omega^1), \ldots] \in F^n$.
If $n = 1$ then output $[a_0]$.
Write $a(x) = b(x^2) + xc(x^2)$
$b \rightarrow [1_0, 1_2, \ldots a_{n-2}]$
$c \rightarrow [a_1, a_3, \ldots a_{n-1}]$
$B \rightarrow DFFT(\frac{n}{2}, b, \omega^2)$
$C \rightarrow DFFT(\frac{n}{2}, c, \omega^2)$
for $i$ from 0 to $\frac{n}{2} - 1$ do
$A_i \rightarrow B_i + \omega^i C_i$
$A_{i+\frac{n}{2}} \rightarrow B_i - \omega^i C_i$
```

4.4 The inverse Fourier Transform

Consider

$$V(\omega) = \begin{array}{c} 111 \cdots 1 \\ 1 \omega \omega^2 \ldots \omega^{n-1} \\ 1 \omega^2 \omega^4 \ldots \omega^{2(n-1)} \end{array}$$

$V(\omega)A = W = [a(1), a(\omega)$. We get n^2 multiplications in F to solve for W.

Lemma. $V_\omega \cdot V_{\omega^{-1}} = nI$

$A = V_\omega^{-1} W = \frac{1}{n} V_{\omega^{-1}} W = \frac{1}{n} DFFT(n, W, \omega^{-1})$.

So we can compute the inverse in $O(n \log(n))$ time as well.

4.5 FFT Multiplication

Given 2 polynomials a, b, we can multiply them by evaluating, multiplying the results, and interpolating that. This takes $O(n \log(n))$ time.

4.6 Computing primitive nth roots of finity

For reals, they don't always exist. So we have to look over C. For Z_p we can do it when $n|p-1$.

To compute over $Z[x]$, we find a large prime p so that Z_p has primitive nth roots and do the computation over that field.

We don't even need a field, we just need that inverses of the primitive roots exist.

5 The Modular GCD algorithm

Let $a = \sum^{n-1} a_i x^i, b = \sum^{n-1} b_i x^i$. Assume all coefficients ahave length $\leq m$.

We can multiply $a \cdot b$ in order $O(n^2 m^2)$ using the classical algorithm. By using the GCD and the CRT, we can reduce this to $O(mn^2 + nm^2)$.

How fast can we compute $gcd(a, b) \in Z[x]$? Primitve Euclidean algorithm does $O(n^2)$ integer operations on integers of size $m, cm, 2cm, 3cm, \ldots (n-1)cm$. This actually takes $O(n^6)$ time to run.

5.1 Modular GCD Algorithm

Definition. Let $g = \gcd(a, b)$. $\bar{a} = a/g$.

Definition. A prime p is unlucky if $gcd(\phi_p(\bar{a}), \phi_p(\bar{b})) \neq 1$.

Definition. A prime is bad if $p|lc(g)$.

Lemma. *If $\phi_p(lc(a)) \neq 0$ then $deg(g_p \geq deg(g)$. Moreover, if $def(g_p) = deg(g)$ hen $\phi_p(g) = ug_p$ for some $u \in Z_p$.*

Proof. Lame. ☐

5.2 The leading coefficient problem

We always get monic polynomials, so we are stuck with geting a monic gcd, so we don't get the correct answer.

If we knew the leading coefficient, we could multiply through by it. Since we don't, we can multiply through by the smaller of the leading coefficients, since the leading coefficient must divide this. This means we have to have a larger bound for our product of primes to be larger than.

```
$\gamma = gcd(lc(a), lc(b))$
$G = 0$
$M = 1$
Loop
Pick a new prime $p$. st. $p \not| lc(a)$. $g_p = gcd(\phi_p(a), \phi_p(b))$.
If $deg(gp) = 0$ output 1.
$g_p : g_p \cdot \gamma \mod p$.
If $G = 0$ then $G = g_p, M = p$.
elif $deg(g_p) > def(G)$ then do nothing
elif $deg(g_p) < deg(G)$ then $G = g_p, M = p$.
else
Use CRT to find gcd $\mod M \cdot p$. If we get the same answer, do trial division, if not, find another prime.
```

5.3 Running time of Modular GCD

Let $a, b, c \in Z[x] - \{0\}, cont(a), cont(b) = 1, g = \gcd(a, b)$. Let $deg(a) = n - 1, deg(b) = n - 1$ $|a|, |b| < B^m$.

How many primes do we need? Assuming we get no unlucky primes. $|g| < 2^{n-2} \min(|a|, |b|) < 2^{n-2} B^m$. $\gamma = \gcd(lc(a), lc(b)) < B^m$

If we assume that all primes are between B and $2B$ then we need $< \log_B 2^{n-1} B^{2m} = (n-1)\log_B(2) + 2m = O(m)$.

- Cost of calculating $\phi_p(a), \phi(b)$

  $O(m)2nO(m) = O(m^2 n)$

- Cost of computing $gcd\phi_p(a), \phi_p(b)$ using Euclid's Algorithm

  $O(m)O(n^2) = O(m^2 n)$

- Cost of applying CRT

  $nO(m^2) = O(nm^2)$

- Cost of pp(g)

  $n(O(m^2) + O(m^2)) = O(nm^2)$

- Trial Divisions

  $O(n^2m^2)$ but we can do it in $O(n^2m+m^2n)$ using the modular division algorithm.

So the total cost is $O(m^2n + n^2m)$

Running time of Euclids Algorithm.

To do division, we need at most $n - m + 1$ steps. This requires $O((n - m + 1)n)$ multiplications.

Euclidean algorithm steps

$O(n - m + 1)m + m^2)$. When $m = n$, we get n^2.

6 The resultant

Let R be an integral domain, $A, B \in R[x]$ of degree $m, n > 0$.

The resultant $res(A, B) = a_m^n \cdot b_m^n \prod \prod (\alpha_i - \beta_j)$.

Sylvester's Matrix $S(A, B)$ is the following:

$$\begin{array}{ccccc}
a_0 & a_1 & \dots & a_n & 0 \\
0 & 1_0 & \dots & & \\
0 & 0 & a_0 & & \\
\vdots b_0 & b_1 & \dots & b_m & 0 \\
0 & b_0 & & & \\
\vdots 0 & \dots & b_0 & \dots & b_m
\end{array}$$

Theorem. $det(S(A, B)) = res(A, B)$

6.1 Project Info

???

7 Newton's Iteration and the Hensel Costruction

7.1 p-adic representation

Theorem. *Let $u \in Z, p > 1$. For $0 \le u \le p^n$, there exist unique integers $u_0, u_1, \dots, u_{n-1}$ s.t. $0 \le u_i \le p$ and $u = u_0 + u_1 p + \dots + u_{n-1}p^{n-1}$*

Proof. obvious. □

What about negative numbers?

Theorem. *Let $u \in Z, p > 2$, p odd. For $\frac{-p^n}{2} < u < \frac{p^n}{2}$ there exist unique integers u_i s.t. $\frac{-p}{2} \leq u_i \frac{p}{2}$ and $u = u_o + u_1 p + \ldots + u_{n-1}p^{n-1}$.*

Application:
Multiply $(3x^2 + 2x + 1)(2x^2 + 2x + 3) = 6x^4 + 8x^3 + 13x^2 + 5x + 3$.
Let $x = 1000$
$3002001 \cdot 2002003 = 6008013005003$
So we can reduce polynomial multiplication to integer multiplication.
$(3x - 3)(2x + 1) = -3 - 3x + 6x^2$
$2997 \cdot 2001 = 5996997$
In maple, $genpoly(5996997, 1000, x)$ will spit out the polynomial.
Let $a \in Z, a \geq 0$. Let $u = \sqrt{a}$. Is $u \in Z$? If $u \in Z$, how can we compute it? For a to have a square root, it must have a square root modulo p for any prime p.
Example: is $\sqrt{12312} \in Z$? examining modulo 5 we get 2, which is a non-residue.

7.2 Linear p-adic Newton Iteration

LEt $a \in Z, a > 0, u = \sqrt{a}$. Suppose $u \in Z$. Let $f(x) = a - x^2$. To compute u, we want to solve $f(u) = 0$. Let p be an odd prime and let $u = u_0 + u_1 p + \ldots + u_{n-1}p^{n-1}$.
Let $u^{(k)} = u_0 + u_1 p + \ldots + u_{k-1}p^{k-1}$. So $u \equiv u^{(k)} \mod p^k$.
Algorithm:

- Solve $a - x^2 \equiv 0 \mod p$ for $x = u_0$ (somehow).
- Given $U^{(k)}$ find $u^{(k+1)}$. For $f(x) = a - x^2$.

$$\begin{aligned}
f(uk + 1 &= f(u^k + u_k p^k) \\
&= a - [u^{(k)} + u_k p^k]2 \\
&= a - [u^{(k)}]^2 - 2u^{(k)} u_k p^k - u_k^2 p^{2k} \\
0 &= f(u^{(k)}) + f'(u^{(k)}) u_k p^k \mod p^{k+1} \\
0 &= \frac{f(u^{(k)})}{p^k} + f'(u^{(k)}) u_k \mod p
\end{aligned}$$

But modulo p, $f'(u^{(k)}) = f'(u_0)$. So $u_k = -\frac{f(u^k)}{p^k} / f'(u_0) = \frac{f(u^k)}{p^k} / 2u_0$.
Stop when $p^k > 2\sqrt{a}$
Example
Find $\sqrt{49}$.
Let $p = 5$. $49 = 4 \mod 5 . e_1 = f(u^1) = 49 - 4 = 45$.
$u_1 = \frac{45}{5} / 4 = 9/4 \mod 5 = 1$

7.3 p-adic representations

Let $a(x) \in Z[x]$ and let $u(x) = \pm\sqrt{a(x)}$. Is $u(x) \in Z[x]$? If so, how do we compute it?

$a = x^3 + 5$ is not a perfect square because the degree is odd and the trailing coefficient is not a perfect square.

We could instead plug in some values and interpolate. But we get two possible square roots for each value and don't know which go together.

So instead, we choose a big integer and plug it in. Then take the square root of it and use that polynomial. But that might give a false positive. So we need to be sure we choose a large enough evaluation point so that the coefficients of the resulting polynomial can be properly captured by the value.

Also, we could reduce the polynomial modulo a prime.

Let $u \in Z$, p an odd prime. If $\frac{-p^n}{2} < u < \frac{p^n}{2}$ then $\exists$ unique $u_i \in Z$ satisfying $u = u_0 + u_1 P + \ldots + u_{n-1}p^{n-1}$ and $\frac{-p}{2} < u\frac{p}{2}$.

Suppose $u(x) = \sum_{i=0}^{m} a_i x^i$ with coefficients less than $|\frac{p^n}{2}$, then there exists unique $u_{ij} \in Z$ such that $u(x) = \sum_{i}^{m} \sum_{j=0}^{n-1} u_{ij} p^j x^i = \sum_{j=0}^{n-1} \sum_{i=0}^{m} u_{ij} x^i p^j$.

Thus, $u(x) = u_0(x) + u_1(x)p + \ldots u_{n-1}(x)p^{n-1}$. If $u(x) = 11x^2 - 7x + 4$ and $p = 3$

$u_0 = -x^2 - x + 1, u_1 = x^2 + x + 1, u_2 = x^2 - x$

7.4 Linear p-adic Newton Iteration

Given $f(u) \in Z[x][u]$, eg. $f(u) = a - u^2$ where $a \in Z[x]$ and given $u_o \in Z_p[x]$ s.t. $f(u_0) \equiv 0 \mod p$, find $\bar{u} \in Z[x]$ s.t. $f(\bar{u}) = 0 \in Z[x]$.

Definition. We let $u^{(k)} = u_0 + u_1 p + \ldots u_{k-1}p^{k-1}$ is called a k^{th} order approximation to $\bar{u}$ if $f(u^{(k)} \equiv 0 \mod p^k$.

Theorem. *If $f(u) \in Z[x][u]$ then $\exists g(u, v) \in Z[x][u, v]$ such that $f(u + v) = f(u) + f'(u)v + g(u, v)v^2$.*

Applying this to $f(u^{(k+1)})$ yields:

$$\begin{array}{rcl} f(u^{k+1}) & = & f(u^k + u_k p^k) \\ & = & f(u^k) + f'(u^k)u_k p^k \mod p^k \\ & \Rightarrow & p|\frac{f(u^k)}{p^k} + f'(u^k)u_k \\ u_k & = & -\frac{f(u^k}{p^k}\frac{1}{f'(u^k)} \mod p \\ & = & -\frac{f(u^k}{p^k}\frac{1}{f'(u_0)} \end{array}$$

If $f(u) = a(x) - u^2$, then $f'(u) = -2u$

$$\begin{aligned} u_k &= \tfrac{a-[u^{(k)}]^2}{p^k}/2u_0 \\ &= \tfrac{e_k}{p^k}/2u_0 \mod p \end{aligned}$$

Definition. $u_k = \frac{e_k}{p^k}/2u_0 \mod p$ is called the linear $p-$adic update.

Example: $u = \sqrt{a = 49x^2 - 56x + 16} = \pm(7x - 4)$
Use $p = 5$.

$$\begin{aligned} u_0 &= 2x + 1 \\ f'(u) &= -24 \\ 2u_0 &= -x + 2 \\ e_1 &= a - u_0^2 \\ &= 45x^2 - 60x + 15 \\ \tfrac{1_1}{p} &= 9x^2 - 12x + 3 \\ u_1 &= \tfrac{-x^2-2x-2}{-x+2} \\ &= x - 1 \mod 5 \\ u^{(2)} &= 2x + 1 + (x-1)\cdot 5 \\ &= 7x - 4 \end{aligned}$$

Theorem. *Let D be an integral domain and let $f \in D[u]$. Then $\exists g \in D[u, y]$ s.t. $f(u + y) = f(u) + f'(u) \cdot y + g(u, y) \cdot y^2$.*

Proof.

$$\begin{aligned} f(u+y) &= a_0(u) + a_1(u)y + g(u,y)\cdot y^2 \\ & \quad y = 0 \Rightarrow \\ f(u) &= a_0(u) \\ &= \tfrac{d}{dy}, y = 0 \Rightarrow \\ \tfrac{df(u+y)}{dy} &= a_1(u) + y \cdot h(u,y) \\ f'(u+y) &= a_1(u) + y \cdot h(u,y) \\ f'(u) &= a_1(u) \end{aligned}$$

□

Theorem. *Let D be an integral domain and $f(u, v) \in D[u, v]$. Then $\exists g, h, i \in D[u, v, y, z]$ s.t. $f(u+y, v+z) = f(u, v)+f_u(u, v)\cdot y+f_v(u, v)\cdot z+g(u, v, y, z)\cdot y^2 + h(u, v, y, z) \cdot z^2 + i(u, v, y, z) \cdot yz$.*

Proof. As in the previous case. □

7.5 Hensel's Lemma

Given $f \in Z[x][u, v]$, solve $f(u, v) = 0$ for $u, w \in Z[x]$.

This can be used for

- GCD $a, b \in Z[x], g = \gcd(a, b), a = g\bar{a}, b = g\bar{b}$, then $a - g\bar{a} = 0$
- Factoring $a = p_1 p_2 \ldots p_n \Rightarrow a - p_1(p_2 \ldots p_n) = 0$.

1. Find $u_0, w_0 \in Z_p[x]$ s.t. $f(u_0, w_0) \equiv 0 \mod p$
2. Given $u^{(k)}, w^{(k)}$, a solution modulo p^k, we find a solution modulo p^{k+1}.

Theorem. *If $f(u, w) \in Z[x][u.w], \exists E, F, G \in Z[x][u, v]$ s.t. $f(u^{(k)}+u_k p^k, w^{(k)}+w_k p^k) = f(u^{(k)}, w^{(k)})+f_u(u^{(k)}, w^{(k)})u_k p^k+f_w(u^{(k)}, w^{(k)})w_k p^k+G(u^k, w^k)u_k^2 p^{2k}+E(u^k, w^k)w_k^2 p^{2k} + F(u^k, w^k)u_k w_k p^{2k}$.*

Considering moduло p^{k+1}, we find that $f_u(u_0, w_0 \cdot u_k + f_v(u_0, w_0)w_k \equiv \frac{f(u^k, w^k}{p^k}$

For $f(u, w) = a - u \cdot w, f_u = -w, f_w = -u$. So we have the equation $u_k w_0 + w_k u_0 \equiv -\frac{e_k}{p^k} \mod p$.

To solve this, we will require $\gcd(u_0, w_0) = 1$.

Lemma. *Let $a(x) \in Z[x]$, p a prime,$p \not| lc(a)$. If u_0, w_0 have gcd 1, then forall n, there exist u^n, w^n such that $u^n w^n = a \mod p^n$ and $u^n = u_0 \mod p, w^n = w_0 \mod p$.*

In our algorithm, we don't necessarily get the proper factors. We must multiply a by $lc(a)$ and then take the primitive part of the resulting factors.

8 Polynomial Factorization

8.1 Kronecker's Algorithm

To find quadratic factors, interpolate them with 3 points.

Consider $a(0), a(1), a(2)$ and look at their factors. Some factor of these must be for the desired polynomial. We try all combinations of them to see.

8.2 Squre free factorizarion

Definition. A polynomial a is square-free if a has no repeated factors.

Lemma. *a is square free iff* $\gcd(a, a') = 1$.

Proof. Easy □

Algroithm Squarefree: Staring with $a = f_1 f_2^2 \cdots f_n^n$, output this.

1. If degree of polynomial is 1 or 0, then done.
2. If $gcd(a, a') = 1$ then output a, otherwise $\bar{a} = \frac{a}{g}$
3. $f_1 = \bar{a} / \gcd(g, \bar{a})$
4. recursively find the other terms.

8.3 Factorization in $Z[x]$

Lemma. *Provided $p \not| lc(a)$ then the number of factors of a over $Z_p[x]$ is at least the number of factors of a over $Z[x]$.*

Belectamp-Hensel Procedure

We have two bad choices: Choose one large prime which has factorizing in $Z_p[x]$ expensive, or choose many small primes which is also expensive.

Instead, we use $\log(degree)$ primes and choose the one with the fewest factors $\mod p$ to reduce the number of combinations to consider.

8.4 Square-free polynomial factoring

To find linear factors modulo p we compute the gcd of a with $x^p - x$.

To find higher degree terms, we take the gcd with $x^{p^k} - x$. However, this gives very high degree polynomials and so doing the computations is not feasible.

Distinct Degree Factorization Algorithm

Input: $a \in Z_p[x], deg(a) > 0, gcd(a, a') = 1$.

Output: $g_1, g_2, \ldots, g_k$ and each g_i is a product of irreducible factors of degree i.

```
$k \leftarrow 1$
$w \leftarrow x$
$while k \leq \frac{deg(a)}{2} do$
   $w \leftarrow rem(w^p, a)$
   $g_k \leftarrow gcd(w - x, a)$
   $a \leftarrow a/g_k$
$end$
$output g_1, g_2, \ldots g_k$
```

How do we compute $rem(w^p, a)$ in $Z_p[x]$? For large p, $r \leftarrow 1$, for $i = 1, 2, \ldots p$ do $r \leftarrow rem(wr, a)$.

We can speed up this implementation by using square and multiply instead.

9 Finite Fields

Lemma. *In $GF(p^k), \alpha^{p^k} = \alpha$.*

Theorem. *x^{p^k} is the product of all monic irreducible polynomials of degree $d|k$.*

How do we split g_k in $Z_p[x]$? If m is irreducible of degree k, $m|v^{p^k} - v$ in $Z_p[x]$. If $p \neq z$ $x^{p^k} = x(x^{p-1}) = x(x^{\frac{p-1}{2}} - 1)(x^{\frac{p-1}{2}} + 1)$. We compute the gcd with each of these factors. The probabolity that we don't split the polynomial is at most $\frac{4}{9}$.

10 Polynomial Factorization

10.1 Computing square roots in Z_p

To try and find the square root of a mod p, we see if the polynomial $x^2 - a$ has linear factors. To do this we take the gcd with $x^p - x$. If we get $x^2 - a$ then we know it is a square. We can find the roots by finding the gcd with $(x + a)^{\frac{p-1}{2}} - 1)$.

10.2 Resultant Calculus

Let $A = A_n x^n + \ldots = a_n \prod(x - \alpha_i)$.

$$res(A, B) = a_n^m \cdot b_m^n \prod \prod (\alpha_i - \beta_j)$$

- $res(c, B) = c^m$
- $res(a, b) = (-1)^n res(B, A)$
- $res(A, B) = \pm b_m^n \prod A(\beta_j)$
- $res(AC, B) = res(A, B) \cdot res(A, C)$
- $res(cA, B) = c^m \cdot res(A, B)$

- $res(BQ+R,B) = b_m^{n-l} res(R,B)$

We can combine these into an algorithm for calculating the resultant of two polynomials.

But we get fractions. So we should use the primitive Euclidean Algorithm.

11 Rational Function Integration

Given $f(x) \in Q(x)$, calculate the antiderivative of $f(x)$.

Theorem. *$\int \frac{1}{x} \notin Q(x)$*

Proof. Assume $\int \frac{1}{x} \in Q(x) = \frac{p(x)}{q(x)}$. Applying the quotient rule, we find that $x^n | q$ for some n. But this would tell us that $x|p$. So we have a contradiction. □

It would be nice if we could compute the integral without using partial fractions.

Theorem. *$\int \frac{P}{Q} = \frac{A}{B} + \int \frac{C}{D}$ where $B = gcd(Q,Q')$, $D = Q/B$. If $deg(A) < deg(B)$ and $deg(C) < deg(D)$ then A, C are unique.*

11.1 Method 1 - Newton's

$\frac{P}{Q} = \frac{A'}{B} - \frac{B'A}{B^2} + CB$

$B|B'D$, so let $H = \frac{B'D}{B}$. So $P = A'D - AH + CB$.

We equate coefficients in x^i and solve for A, C.

11.2 Method 2 - Hermite's

Let $Q = g_k^k T$. Solve $\sigma T q_k' + \tau q_g = P$ for σ, τ with $deg(\sigma) < deg(g_k)$.

$$\frac{P}{Q} = \sigma \frac{Tq_k'}{Q} + \tau \frac{q_k}{Q} = \sigma \frac{q_k'}{q_k^k} + \frac{tau}{Tg_k^{k-1}}$$

integrating, we get

$\int \frac{P}{Q} = \frac{\sigma/(1-k)}{+} \int \frac{\tau - T\sigma'(1-k)}{Tq_k^{k-1}}$.

Theorem. *Let $C, D \in k[x], deg(C) < deg(D), \gcd(D, D') = 1, lc(D) = 1$. Let $R(z) = res_x(C - ZD', D) \in K[z]$. Then $\int \frac{C}{D} = \alpha_1 \ln(v_1) + \cdots + \alpha_k \ln(v_k)$, where α_i are the roots of $R(z)$ and $v_i = gcd(C - \alpha_i D', D)$. Moreover, $L = K(\alpha_1, \ldots, \alpha_k)$ is the smallest field extension of K in which we can write $\int \frac{C}{D}$.*

Lemma. $\alpha_i = c(beta_i)/D'(\beta_i)$

Consider $R(z) = res_x(C(x) - zD'(x), D(x))$.

Lemma. $x - \beta_j | gcd(x(C) - \alpha_i D'(x), D(x)) \iff \alpha_j = \alpha_i$

12 Risch Integration Algorithm

12.1 The Risch Structure Theorem

Let K be a field of constants, $F_0 = K(x)$ and $F_n = F_0(\theta_1, \theta_2, \ldots, \theta_n)$ where $\theta_i' \neq 0$ and θ_i is either logarithmic, exponential, or algebraic over F_{i-1}. Let $f(x) \in F_n$. To compute $\int f(x)dx$ we want a representation for F_n s.t.

- $\theta_i \notin F_{i-1}$
- n is as small as possible

Theorem. *Let F be a differential field with algebraically closed constant field K. Suppose $f(x) \in F$ and $\int f(x)dx \in G$ where G is an elementary extension of F. Then $\exists v_0, v_1, \ldots, v_m \in F$ and $c_1, \ldots, c_m \in K$ such that $\int f(x)dx = v_0 + c_1 \log(v_1) + \ldots + c_m \log(v_m)$*

12.2 Special case of one logarithmic extension

Suppose $\int f(x) \in F(\theta)$ where $\theta = \log(u), u \in F, u' \neq 0, \theta \notin F$. Then $\int f(x) = \frac{a(\theta)}{b(\theta)}$ for some $a, b \in F[\theta]\ \{0\}$ with $gcd(a, b) = 1$ in $F[\theta]$ and $lc_\theta(b) = 1$.

12.3 The Risch Algorithm

Let $F_n = K(x, \theta_1, \ldots, \theta_n)$ where K is a constant field, θ_i is an elementary extesnion of F_{i-1} and $\theta_i' \neq 0$. To integrate $f(x) \in F_n$ the algorithm will need to integrate functions in F_{n-1} recursively. So the base of the recursion is $K(x)$.

Consider the integral

$$\int \frac{1 + x\log(x)}{(1+x)^2 \log(X)} dx$$

We have $F = C(x)(\theta_1 = \log(x))$ and we must have

$$\begin{array}{rl} & \int \frac{1+x\log(x)}{(1+x)^2\log(X)}dx \\ = & v_0 + \sum c_i \log(v_i) \\ = & \frac{a}{b} + c_1 \log(1+x) + c_2 \log(\log(x)) \end{array}$$

where $v_i \in F$ and $c_i \in C$.

12.3.1 The Transcendental Case

$\theta_i = \log(u_i)$ or $\theta_i = e^{w_i}$

12.4 The Logarithmic Extension subcase

Let $f(x) = F(\theta), \theta = \log(u), u \in F, \theta \notin F, \theta' \neq 0$ and $F = K(x, \ldots)$

$$\int f(x) = \int \frac{a(\theta)}{b(\theta)} = \int P + \frac{R}{b}$$
$$\int \frac{R}{b} = \frac{A}{B} + \int \frac{C}{D}$$

$\int \frac{\log^3(x)+1}{\log^2(x)} = \int \log(x) + \frac{1}{\log^2(x)}$

12.4.1 Hermite Reduction

$\int \frac{P}{Q}$ where $P, Q \in F[\theta], \theta = \log(u)$ and degree $P <$ degree Q.

Let Q be a square free factorization of Q in $F[\theta]$ $(Q = q_1 q_2^2 \cdots)$

Let $T = \frac{Q}{q_k^k}$.

Solve $\sigma q_k' T + \tau q_k = P$ for $\sigma, \tau \in F[\theta]$ with degree $\sigma <$ degree q_k.

$$\int \frac{P}{Q} = \int \frac{\sigma q_k'}{q_k^k} + \frac{\tau}{Q/q_k} = -\frac{\sigma/(k-1)}{q_k^{k-1}} + \int \frac{\sigma'/(k-1)T + \tau}{Q/q_k}$$

$\int \frac{1}{\log^2(x)} = \int \frac{1}{\theta^2}$

Solve $\sigma \frac{1}{x} 1 + \tau\theta = 1$ to get $\sigma = x, \tau = 0$.

So we have our integral is $-\frac{x}{\log(x)} + \int \frac{1}{\log(x)}$.

But we kind of cheated, since we need q_k', q_k relatively prime when the derivative is taken wrt x, but we only know that is is true wrt θ.

Theorem. *If $q_k, \frac{d}{d\theta} q_k$ are relatively prime then $q_k, \frac{d}{dx} q_k$ are relatively prime.*

Theorem. *Let $R(z) = res_\theta(C - zD', D) \in F(z)$ then*

$\int \frac{C}{D}$ is elementary iff all roots of $R(z)$ are constants.

$\int \frac{C}{D} dx = \sum c_i \log(v_i)$ where c_i are the roots of the resultant and $v_i =$ monic $gcd(C - c_i D', D) \in F[x]$.

12.4.2 The Polynomial Part

$\int P(\theta)dx, P \in F[\theta], \theta = \log(u), u \in F, \theta' \neq 0$

Let $P = \sum p_i\theta^i$. By Louisville's Thm, if the integral is elementary (ie computable) then $\int P(\theta) = v_0(\theta) + \sum c_i v_i \theta$ where $v_i \in F, c_i \in K, v_0(\theta) \in F[\theta]$.

We equate coefficients of θ^i.

$$\begin{aligned}
&p_l = q_{l+1}(l+1)\theta' + q_l' \\
&p_{l-1} = lq_l\theta' + q_{l-1}' \\
&\vdots \\
&p_1 = q_1\theta' + q_0' \\
&p_0 = q_1\theta' + q_0' + \sum L
\end{aligned}$$

We recursively compute $\int p_l$. If $\int p_l$ is not elementary then $\int P$ is not elementary. If $\int p_l$ is elementary but contains $\log(v) \in F[\theta]$ then $\int P$ is not elementary.

12.5 Expopnential Subcase

See text.

13 Exam

9:30 am on April 21st. 24 hour take home final.