Intro to Symbolic Computation

CS 687 (Winter 2020) with Professor Arne Storjohann

University of Waterloo

Scribe : Saiyue Lyu

Contents

List of Definitions

Chapter 1

Basic Algebraic Operations

1.1 Overview

Example 1.1.1 (Simplifying Rational Expressions)

$$f = \frac{x+1}{x-1} - \frac{x^3+2x+x^2+2}{x^3+2x-x^2-2} + \frac{x^2+3}{x-1}$$
$$g = \frac{(x-1)^2 - x^2 - 1 + 2x}{(x+y+2)^{100}}$$

One can define a 'normal' function by:

if the expression is zero, then it should return 0

if the expression is nonzero, then it returns $\frac{poly}{poly}$ in lowest terms.

$$normal(f) = \frac{x^2+3}{x-1}$$
$$normal(g) = 0$$

Example 1.1.2

Solving Recurrences :

$$T(n) = \begin{cases} 2T(n/2) + n/2 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Give it to Maple, we get $T(n) = n(1 + log_2 n)$

Example 1.1.3 (Symbolic Summation)

$\sum_{i=0}^{n-1} i^4 = n(n-1)(2n-1)(3n^2-3n-1)/30$

Example 1.1.4

What is gcd of -15 and 6?

either -3 or 3.

For this course, -3 = lu(-3) $\cdot$ $normal(3) = (-1) \cdot 3$, lu stands for leading unit.

Remark
Solution $A^{-1}b$ contains rational numbers even though input is only integers. Why? A Crammer's Rule

$$A^{-1}b = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

where $v_i = \frac{det \text{ of A with column } i \text{ replaced by } b}{det\ A}$

Remark
Let $A \in \mathbb{Z}^{n \times n}$, how "large" is $det(A)$?

Let $||A|| = \max_{i,j} |A_{ij}|$, Hadamard's bound gives $|det(A)| \leq n^{n/2}||A||^n$

N=magnitude	$log_{10}N$
10	2
100	3
1000	4

For example, $\log|det(A)| \in \mathcal{O}\big(n(logn + log||a||)\big)$

1.2 Representation of Integer

Current computer based on architecture with 64 bits, i.e. word-size=64

For example, "unsigned long" in $\mathbb{C}$, can represent integer exactly in the range $[0, 2^{64} - 1]$

How to represent a larger number?

Use an array of word-size numbers. Any integer a can be represented as

$$a = (-1)^s \sum_{i=0}^{n} a_i 2^{64}$$

where $s \in \{0, 1\}$ and $0 \leq a_i \leq 2^{64} - 1$

For example, $532 = (-1)^0 \cdot (2 + 3 \cdot 10 + 5 \cdot 10^2)$

If we assume $0 \leq n+1 < 2^{63}$, then we can encode a as an array $[s \cdot 2^{63} + n + 1, a_0, a_1, \cdots, a_n]$

Which is sufficient for all practical purposes.

Note : The length of a is given by $\lfloor log_{2^{64}}|a| \rfloor + 1 \in \mathcal{O}(log|a|)$ words.

1.3 Addition of Integer

Input : $a_0 + a_1\beta + \cdots + a_m\beta^m + \cdots + a_n\beta^n$ and $b_0 + b_1\beta + \cdots + b_m\beta^m$

Output : $c_0 + c_1\beta + \cdots +$

How large can $* + * \cdot \beta + \cdots$ be?

If $\beta = 2$, then $1 + 1 \cdot 2 + \cdots + 1 \cdot 2^m = 2^{m+1} - 1$

So $\sum_{i=0}^{n}(\beta-1)\beta^i = \beta^{m+1} - 1$

How large is $a_0 + a_1\beta + \cdots + a_m\beta^m + (b_0 + b_1\beta + \cdots + b_m\beta^m) \le 2\beta^{m+1} - 2 = \beta^{m+1} - 2 + \beta^{m+1}$

For example, 111111+111111=1111110 in modulo 2

1.4 Representation and Addition of Polynomials

Example 1.4.1

$a = 3x^2 + 12x + 2 \in \mathbb{Z}[x]$ with coefficients 3,12,2 come from $\mathbb{Z}$

Aside: a ring R has the operations $\{+, -, \times\}$ with an identity element 1 and usual rules for arithmetic (Commutativity, Distributivity, Associativity).

Addition of two polynomials with degree bounded by n costs at most (n+1) additions of ring elements from R

1.5 Complexity of Arithmetic Operations

Basic operation $\{+, -, \times, " \div "\}$ over a ring, where $" \div "$ is not always possible. Note R is commutative with identity.

1.6 Concrete Ring

1) $\mathbb{Z}$

2) $\mathbb{Q}$

3) $\mathbb{Z}/(7) = \{0, 1, 2, 3, 4, 5, 6\}$, which is a field, i.e. all nonzero elements are invertible.

   For example, 2+3=5, 3+5=1, 2 $\cdot$ 3=6, 2 $\cdot$ 4=1, then $2^{-1} = 4$.

4) $R[x]$, where R is any commutative ring (e.g. as above). For example, $\mathbb{Z}[x], \mathbb{Q}[x], \mathbb{Z}/(p)[x]$.

5) $\mathbb{Q}(x)$, the field of rational functions

1.7 Naive Upper Bound on Cost (up to multiplicative const) Operation

operations	$a, b \in R[x] \setminus \{0\}$, $n = deg\, a, m = deg\, b$	$a, b \in \mathbb{Z}$ bit operations
$a+b$	$n+m+1$	$\lg a + \lg b$
$a-b$	$n+m+1$	$\lg a + \lg b$
$a*b$	$(n+1)(m+1)$	$(\lg a)(\lg b)$
$a = qb + r$	$(n-m+1)(m+1)$	$\lg a/b)(\lg b)$

For $a \in \mathbb{Z}, \lg a = \begin{cases} 1 & \text{if } a = 0 \\ 1 + \lfloor log_2 |a| \rfloor & \text{otw} \end{cases}$

Addition

Over $R[x]$, $(a_0 + \cdots + a_m x^m + \cdots + a_n x^n) + (b_0 + \cdots + b_m x^m) = c_0 + \cdots + c_m x^m + \cdots + c_n x^n$. If $m \leq n$, exactly how many basic operations from R? $m+1$! But realistically, what is the cost of a function $C = add(a, b)$.

Over $\mathbb{Z}$, same idea in radix B expansion, but with carries. E.g, $a = 66599989, b = 911$,

$$\begin{array}{r} 66599989 \\ 911 \\ 66600900 \end{array}$$

Multiplication

First consider polynomials in $R[x]$, $a = \sum_{i=0}^{n} a_i x^i$, $b = \sum_{i=0}^{m} b_i x^i$, then

$$c = a * b = \sum_{k=0}^{n+m} c_k x^k \ , \ c_k = \sum_{0 \leq i \leq n, 0 \leq j \leq m, i+j=k} a_i b_j$$

Classical school method:

$$\begin{aligned} &(a_2x^2 + a_1x + a_0) \cdot (b_3x^3 + b_2x^2 + b_1x + b_0) \\ =&(a_0b_3x^3 + a_0b_2x^2 + a_0b_1x + a_0b_0)+ \\ &(a_1b_3x^4 + a_1b_2x^3 + a_1b_2x^2 + a_1b_0x)+ \\ &(a_2b_3x^5 + a_2b_2x^4 + a_2b_1x^3 + a_2b_0x^2) \end{aligned}$$

cost is $(n+1)(m+1)$ multiplications and nm additions exactly.

$\Rightarrow$ $\Theta(nm)$ arithmetic operations from R, if $deg\, a, deg\, b > 0$

Algorithm 1: Asido

Result: Too slow, cost is $\Theta(a * (\lg ab))$
//compute $c = ab$ where $a, b > 0$;
$aa = a$;
$c = 0$;
while $aa > 0$ **do**
  $c = c + b$;
  $aa = aa - 1$;
end

Division With Remainder $a = qb + r$

Given $a, b \in R[x]$ or $a, b \in \mathbb{Z}$, express $a = qb + r$, where $size(r) < size(b)$

Example 1.7.1
$a = 32125, b = 123,$

$$
\begin{aligned}
32125 &= 200 \cdot b + 7515 \\
&= 200b + 60b + 135 \\
&= 200b + 60b + b + 12 \\
&= 261b + 12
\end{aligned}
$$

Polynomial Division

Given $a, b \in R[x]$ with $b \neq 0$, find $q, r \in R[x]$ such that $a = qb + r, deg\, r < deg\, b$

Assume $lc(b)$, the leading coefficient of b is a unit, i.e. has an inverse in R

Example 1.7.2
Over $\mathbb{Z}$, the units are ± 1. Over $\mathbb{Q}$, all nonzero elements are units.

Example 1.7.3
$b = x^2 + 2x + 1$, $a = 3x^5 + 2x^4 + x^3 + x^2 + 2x + 1$

$$
\begin{array}{r}
3X^3 - 4X^2 + 6X - 7 \\
X^2 + 2X + 1 \overline{\big) \, 3X^5 + 2X^4 + X^3 + X^2 + 2X + 1} \\
\underline{-3X^5 - 6X^4 - 3X^3} \qquad\qquad\qquad\quad \\
-4X^4 - 2X^3 + X^2 \qquad\qquad\quad \\
\underline{4X^4 + 8X^3 + 4X^2} \qquad\qquad\quad \\
6X^3 + 5X^2 + 2X \qquad\quad \\
\underline{-6X^3 - 12X^2 - 6X} \qquad\quad \\
-7X^2 - 4X + 1 \\
\underline{7X^2 + 14X + 7} \\
10X + 8
\end{array}
$$

In general $(b_m x^m + \cdots)(a_n b_m^{-1} x^{n-m}) + r = a_n x^n + \cdots$, think of a 45 degree - half space of the whole square :)

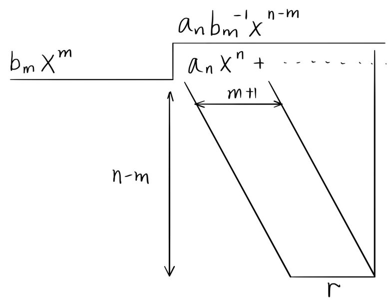

Figure 1.1: Division

If $deg\, a = n$ and $deg\, b = m$, then $deg\, q = n - m$ and the cost of division with remainder is $\mathcal{O}\big((1 + deg\, b)(1 + deg\, q)\big) = \mathcal{O}\big((m + 1)(n - m + 1)\big)$ arithmetic operations from R

1.8 Reduction Modulo Many Primes/Moduli

Suppose $a \in \mathbb{Z}$, $p_1, p_2, \cdots , p_k \in \mathbb{Z}_{>1}$ with $a < P := p_1 p_2 \cdots p_k$

Example 1.8.1

$a = 581869302, P = (30)(17017)(12673) = 2 * 3 * 5 * 7 * 11 * 13 * 19 * 23 * 29 = 6469693230$, what is the cost of computing $a\ rem\ p_1 , \cdots , a\ rem\ p_k$?

The cost is bounded by $\mathcal{O}\big(k(\lg P)^2\big)$

More detailed, the cost is bounded by

$$\begin{aligned}
\sum_{i=1}^{k} C(\lg a/p_i)(\lg p_i) &= C \sum_{i=1}^{k} (\lg a/p_i)(\lg p_i) \\
&\leq C \sum_{i=1}^{k} (\lg P)(\lg p_i) \quad \text{make simplification } \lg a/p_i \leq \lg a \leq \lg P \\
&= C(\lg P) \sum_{i=1}^{k} (\lg p_i) \\
&\leq C(1 + \log_2 P) \sum_{i=1}^{k} (1 + \log_2 p_i) \quad \text{by definition of } \lg \\
&\leq C(2 \log P) \sum_{i=1}^{k} (2 \log p_i) \quad \text{if } x > 1, \text{ then } 1 + \log \lambda \leq 2 \log \lambda \\
&\leq 4C(\log P)(\log p_1 p_2 \cdots p_k) \\
&= 4C(\log P)(\log P)
\end{aligned}$$

which is $\mathcal{O}\big((\log P)^2\big)$, independent of k

Runtime Analysis in Naive Cost Model over $\mathbb{Z}$

- Introduce constant C from big $\mathcal{O}$ bound
- Use sums
- $\lg a \leq 1 + \log_2 a$ if $a > 1$
- if $x > 1$, then $1 + \log \lambda \leq 2 \log \lambda$

1.9 Greatest Common Divisor

Unit : units in $\mathbb{Z}$ are ± 1, units in $F[x]$(a field) are all non-zero constant polynomials, for example, in $\mathbb{Q}[x]$, $3 \cdot \frac{1}{3} = 1$, but $3x + 1$ does not have an inverse.

Associates : elements $a, b \in R$ are associate if there exists a unit $u \in R$ such that $a = ub$, and thus $u^{-1}a = b$, for example, 3 and -3 over $\mathbb{Z}$, $2x^2 + 3$ and $x^2 + \frac{3}{2}$ over $\mathbb{Q}$.

Zero Divisor : A zero divisor in R is an element $a \in R$ such that $\exists b \in R \setminus \{0\}$ with $ab = 0$, for example, in $\mathbb{Z}/(6)$, $2 \cdot 3 = 0$, then 2 and 3 are zero divisors.

Integral Domain : An integral domain is a ring with no zero divisors. For example, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{Z}/(p)$ for p prime

Field : A field is an integral domain !!!!!!!

GCDs : Need not to always exists (depends on R), but GCDs do exists over $\mathbb{Z}$ and over $R[x]$ when R is a field.

LCM : Least Common Multiple is defined similar.

It is convenient to define $gcd(a, b)$ and $lcm(a, b)$ to be non-negative to make them unique.

Euclidean Domain : For exmaple, $\mathbb{Z}$, $F[x]$. Note $q = a$ quo b, $r = a$ rem b.

when $R = \mathbb{Z}$, $d(a) = |a|, d(0) = 0$, quo and rem are not unique over $\mathbb{Z}$, $7 = 5 \cdot 1 + 2 = 5 \cdot 2 - 3$.

when $R = F[x]$, F a field. $d(a) = deg(a), d(0) = -\infty$, quo and rem are unique over $F[x]$

1.10 Extended Euclidean Algorithm

Input $a, b \in R$, $b \neq 0$, R a Euclidean Domain

Output $s, t, g \in R$ such that $sa + tb = g$, where g is a gcd of a, b

Example 1.10.1

compute gcd of 91 and 63 : $\begin{bmatrix} & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 91 \\ 63 \end{bmatrix} = \begin{bmatrix} 63 \\ 28 \end{bmatrix}$, then 28 =rem (91,63), 1=quo(91,63)

$\begin{bmatrix} & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 63 \\ 28 \end{bmatrix} = \begin{bmatrix} 28 \\ 7 \end{bmatrix}$, then 7=rem (63,28), 2=quo(63,28)

$\begin{bmatrix} & 1 \\ 1 & -4 \end{bmatrix} \begin{bmatrix} 28 \\ 7 \end{bmatrix} = \begin{bmatrix} 7 \\ 0 \end{bmatrix}$

Let $Q = \begin{bmatrix} & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -4 \end{bmatrix} = \begin{bmatrix} -2 & 3 \\ 9 & -13 \end{bmatrix}$, then we have

$\begin{bmatrix} -2 & 3 \\ 9 & -13 \end{bmatrix} \begin{bmatrix} 63 \\ 28 \end{bmatrix} = \begin{bmatrix} 7 \\ 0 \end{bmatrix}$, which gives (-2)· 63 +3·28=7

Algorithm 2: Extended Euclidean Algorithm (EEA)

Input: $a, b \in R$, $b \neq 0$, R is a ED, $d(a) = d(b)$

$r_0 = a$;

$r_1 = b$;

$c = 0$;

for $i \geq 10$ **do**

Compute q_i and r_{i+1} such that $r_{i-1} = q_i r_i + r_{i+1}$;

$Q_i = \begin{bmatrix} & 1 \\ 1 & -q_i \end{bmatrix}$;

$Q_i \begin{bmatrix} r_{i-1} \\ r_i \end{bmatrix} = \begin{bmatrix} r_i \\ r_{i+1} \end{bmatrix}$;

Stop loop at $i = l$ such that $r_{i+1} = 0$

end

Claim 1.10.1

r_l is a gcd(r_0, r_1)

Proof

1) $r_l | r_0$ and $r_l | r_1$ 2) if $d | r_0$ and $r | r_1$, then $d | r_l$, for all $d \in R$.

Then $Q_l Q_{l-1} \cdots Q_1 \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} r_l \\ 0 \end{bmatrix}$

Let $R_i = Q_i Q_{i-1} \cdots Q_1 = \begin{bmatrix} s_i & t_i \\ s_{i+1} & t_{i+1} \end{bmatrix}$

Then $\begin{bmatrix} s_l & t_l \\ s_{l+1} & t_{l+1} \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} r_l \\ 0 \end{bmatrix}$, so $s_l r_0 + t_l r_1 = r_l$

Each Q_i is invertible over R where $Q_i^{-1} = \begin{bmatrix} q_i & 1 \\ 1 & \end{bmatrix}$

Then each R_i is invertible over R, in particular, $\begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = R_l^{-1} \begin{bmatrix} r_l \\ \emptyset \end{bmatrix}$

1.11 Cost Analysis of Extended Euclidean Algorithm

Consider $R = F[x]$, assume $deg\, r_0 \geq deg\, r_1$

Cost of computing (q_i, r_{i+1})

Question : How many divisor steps l?

Answer : $1 \leq deg\, r_1$ since $-\infty = deg\, r_{l+1} < deg\, r_l < \cdots < deg\, r_1$

Note Dividing r_{i-1} by r_i with remainder costs $C(deg\, r_i + 1)(deg\, q_i + 1)$ operations from F

Key Observation :

$$\begin{aligned}\sum_{i=1}^{l} deg\, q_i &= \sum_{i=1}^{l}(deg\, r_{i-1} - deg\, r_i) \\ &= (r_0 - r_1) + (r_1 - r_2) + \cdots + (r_{l-1} - r_l) \\ &\leq deg\, r_0\end{aligned}$$

Total cost (operation from F) is thus :

$$\begin{aligned}&\leq \sum_{i=1}^{l} C(deg\, r_i + 1)(deg\, q_i + 1) \\ &\leq \sum_{i=1}^{l} C(deg\, r_1 + 1)(deg\, q_i + 1) \quad \text{since } deg\, r_i \leq deg\, r_1 \\ &\leq C(deg\, r_1 + 1)\sum_{i=1}^{l}(deg\, q_i + 1) \\ &\leq C(deg\, r_1 + 1)(deg\, r_0 + l) \\ &\in \mathcal{O}\big((1 + deg\, r_0)(1 + deg\, r_1)\big) \text{ operations from } F\end{aligned}$$

Extension : what is cost of computing ?

$Q_i Q_{i-1} \cdots Q_1 = \begin{bmatrix} & 1 \\ 1 & -q_i \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -q_{i-1} \end{bmatrix} \cdots \begin{bmatrix} & 1 \\ 1 & -q_1 \end{bmatrix} = \begin{bmatrix} s_l & t_l \\ s_{l+1} & t_{l+1} \end{bmatrix}$

Which is per multiplying R_i with Q_{i-1}, pretty similar as above, still $\mathcal{O}\big((1+deg\, r_0)(1+deg\, q_{i-1})\big)$

1.12 Applications of the EEA

- Computing over finite field $\mathbb{Z}/(p) = \{0, 1, \cdots, p-1\}$
- Operations are $\{+, -, \times, /$ by nonzero, $\}$
- Given nonzero $a \in \mathbb{Z}/(p)$, use EEA to find $s, t \in \mathbb{Z}$ such that $sa + tp = 1$, then $sa \equiv 1 \mod p$, thus $s = a^{-1}$ in $\mathbb{Z}/(p)$

1.13 Rational Number Reconstruction

For $-4/5 \equiv 40 \mod 51$ over rationals $\mathbb{R}$, 40 is called modular image.

Input : a modulos $m \in \mathbb{Z}_+$,

an image $u \in \mathbb{Z}_{\geq 0}$ such that $0 \leq u < m$,

bounds $N, D \in \mathbb{Z}_+$ such that $2ND < m$

Output : A signed and reduced number n/d such that $n/d \equiv u \mod m$, $|n| \leq N, d \leq D$

Fact : there is a unique n/d if it exists, that satisfy the bounds.

Example 1.13.1 (Algorithm use EEA on m and u)
$u = 40, m = 51, N = D = 5,$

then $Q_6 \cdots Q_1 \cdot v = \begin{bmatrix} & 1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 51 \\ 40 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

We look at $R_3 = Q_3 Q_2 Q_1 = \begin{bmatrix} -3 & 4 \\ 4 & -5 \end{bmatrix}$, $R_3 v = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$

Then $(-3)51 + (4)40 = 7$, therefore $40 \equiv 7/4 \mod 51$

We look at $R_4 = Q_4 Q_3 Q_2 Q_1 = \begin{bmatrix} 4 & -5 \\ -7 & 9 \end{bmatrix}$, $R_4 v = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$,

then $4 \cdot 51 + (-5)40 = 4$, therefore $40 \equiv -4/5 \mod 51$

Chapter 2

Evaluation and Multiplication of Polynomials

2.1 Motivation

$f(x) = 5x^{100} + 2x^{999} + \cdots + 3x + 2I_n \in F[x]$ with $F = \mathbb{Z}/(7)$

Suppose $\alpha = \begin{bmatrix} 2 & 3 & 1 & \cdots & 6 \\ 6 & 3 & 0 & \cdots & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 3 & 2 & \cdots & 5 \end{bmatrix} \in F^{300\times 200}$

Question : What is the cost to evaluate $F(\alpha)$?

- Expensive operation is matrix multiplication (How many?)
- $\alpha^2, \alpha^3, \cdots \alpha^{1000}$
- Need 999 matrix multiplications

Today : Method that needs only 63 matrix multiplication.

Problem : R, a ring with binary operations $\{+, -, \times\}$. Given $n \in \mathbb{N}$, find algorithm that does polynomial evaluation.

Input : $\alpha\,, a_0\,, \cdots\,, a_n \in R$, where α is the evaluation

Output : $f(\alpha) \in R\,,\ f(x) = a_n x^n + \cdots + a_1 x + a_0 \in R[x]$

Outline
- Obvious Algorithm
- Horner's Scheme
- Non-scalar Complexity
- Paterson and Stockmeyer
- Karatsuba

2.2 Obvious Algorithm

- Compute $\alpha^2, \alpha^3, \cdots, \alpha^n$. (cost $n-1$ multiplications)
- Compute $a_i\alpha^i$. (cost n multiplications)
- Add. (cost n additions)

2.3 Horner's Scheme

$$f(\alpha) = \Big(((\cdots(\underbrace{a_n\alpha + a_{n+1}}_{\text{1multiplication, 1 addition}})\alpha + \cdots)\alpha + a_2)\alpha + a_1\Big)\alpha + a_0$$

- Repeat n times
- n multiplications and n additions
- In 1954, Ostrowski asked if Horner's Scheme is optimal.

2.4 Non-scalar Complexity

Let $R = F[x, a_0, \cdots, a_n, \alpha]$, ring of polynomials in indeterminants $x, a_0, \cdots, a_n$

Scalar Operation : Addition of two elements of R.

Multiplication of elements of R by a fixed constant of F

Non-scalar Complexity : Multiplication of two input or non-scalar quantities.

Example 2.4.1 (Circuit for Horner's Scheme)
picture here!!!

Aside (Circuit Model is Useful)

Example 2.4.2
Compute dot product $\begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$

picture here!!!

What does the depth of the circuit correspond to ? $logn$?

Question : Is Horner's rule optimal w.r.t non-scalar cost?

No! Victor Pan 1959

2.5 Evaluation at a Known Polynomial

Now let $f \in F[x]$ of degree n be fixed.

Example 2.5.1
$f_1 = a_1x + a_0$, $f_2 = a_2x^2 + a_1x + a_0$

picture here!!

The only non-scalar quantity is α

Question : What is the non-scalar cost (if α is the only input)?

So non-scalar quantities are those that involved α

Only counting multiplications of two non-scalars

Answer : For deg 1, it is 0, for deg 2, it is 1, etc.

Theorem 2.5.1 (Patterson and Stockmeyer 1973)
Let $f \in F[x]$ of degree n, then $f(\alpha)$ can be evaluated at any $\alpha \in F$ with $2\lceil\sqrt{n}\rceil - 1$ non-scalar (both operands involve α) multiplications

Partition f into about $k \cong \sqrt{n}$ blocks of length $m \cong \sqrt{n}$

let $m = \lceil\sqrt{n}\rceil$, $k = 1 + \lceil\frac{n}{m}\rceil$

Example 2.5.2
if $n = 8$, then $m = 3$ (the length of each block) and $k = 4$ (upper bound on number of blocks)

$$\begin{aligned} f(x) &= \underbrace{2x^8 + x^7 + 5x^6}_{F_2x^6} + \underbrace{2x^5 + 8x^4 + 2x^3}_{F_1x^3} + \underbrace{x^2 + x + 4}_{F_0} \\ &= (2x^2 + x + 5)x^6 + (2x^2 + 8x + 2)x^3 + (x^2 + x + 4) \end{aligned}$$

Algorithm :
- Compute $\alpha, \alpha^2, \cdots, \alpha^n$

  cost is $m - 1$ non-scalar multiplication
- Compute $\beta_i = F_i(\alpha)$ for $0 \leq 1 \leq k - 1$

  cost is zero because all multiplications are by scalars.
- $f(\alpha) = \beta_{k-1}(\alpha^m)^{k-1} + \beta_{k-1}(\alpha^m)^{k-1} + \cdots + \beta_0$

  now use Horner's rule, cost k-1 non-scalar mults and some free adds

Total cost : $(m-1) + (k-1) \leq 2\lceil\sqrt{n}\rceil - 1$ non-scalar mults

2.6 Polynomial Multiplications

Input : $f, g \in R[x]$ of degree $n > 0$

Standard Algorithm :
- $f * g$ costs $\mathcal{O}(n^2)$ from R
- $(n+1)^2$ mults and $n^2 + 1$ adds

Example 2.6.1
$(ax+b)(cx+d) = acx^2 + (ad+bc)x + bd$

Let $n = 2^k, k \in \mathbb{N}, a, b \in R[x]$ with $deg\, a < n$, $deg\, b < n$ and $m = n/2$

Write : $a = (A_1x^m + A_0)$ and $b = (B_1x^m + B_0)$

Example 2.6.2
For $a = x^5 + 3x^4 + 2x^3 + x^2 + 3x + 5, n = 8, m = 4, a = (x+3)x^4 + (2x^3 + x^2 + 3x + 5)$

$ab = A_1B_1x^n + (A_0B_1 + A_1B_0)x^m + A_0B_0$ (recall $n = 2m$)

Cost? Let $T(n)$ be the cost of multiplying two polynomials of $deg\ < n = 2^k$, then

$$\begin{aligned} T(n) &\leq \begin{cases} 4T(\frac{n}{2}) + 4n & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases} \\ &= n(5n-4) \\ &\in \Theta(n^2) \end{aligned}$$

2.7 Karatsuba

$ab = A_1B_1(x^n - x^m) + (A_1 + A_0)(B_1 + B_0)x^m + A_0B_0(1 - x^m)$

$$\begin{aligned} T(n) &\leq \begin{cases} 3T(\frac{n}{2}) + cn & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases} \\ &\in \Theta(n^{\log_2 3}) \ \ \text{note } \log_2 3 \simeq 1.59 \end{aligned}$$

Theorem 2.7.1
$T(2^k) \leq 3T(2^{k-1}) + c2^k \ \Rightarrow \ T(2^k) \leq 3^k - 2c2^k$ for $k \geq 1$

Proof
By induction on k

Assume it is true for some $k - 1 \geq 1$, prove for k

$$\begin{aligned} T(2^k) &\leq 3T(2^{k-1}) + c2^k \\ &\leq 3(3^{k-1} - 2c2^{k-1}) + c2^k \\ &= 3^k - 2c2^k \end{aligned}$$

Since $3^k = 3^{\log_2 n} = (2^{log_2 3})^{\log_2 n} = 2^{(\log_2 3)(\log_2 n)} = n^{\log_2 3}$

About A1 Q2:

Consider $f, g \in R[x]$, $deg\, f = n > 0$, $deg\, g = m > 0$, naive cost $\mathcal{O}(nm)$ ring operations from R to compute fg, but what about Karatsuba?

Suppose $n > m$, then the cost is $\mathcal{O}(n^{1.59})$, but if $m \in \mathcal{O}(n^{1/2})$, then naive cost is only $\mathcal{O}(n^{1.5})$. For example, if $m \in \mathcal{O}(n^{1/2})$, can show how to use karatsuba with cost $\mathcal{O}(n^{1.295})$.

Algorithm 3: About A1Q2

---

ebinarygcd := proc(a,b);
local $\bar{s}\,,\bar{t}\,,\bar{g}$;
if $modp(a,2) = 0$ *and* $modp(b,2) = 0$ **then**
  $\bar{s}\,\bar{t}, \bar{g}$:=ebinarygcd($\frac{a}{2}, \frac{b}{2}$)
end
Want: $\bar{s}(a/2) + \bar{t}(b/2) = \bar{g}(= gcd(a/2, b/2))$ **Have:** $sa + tb - g$

Other cases : $\begin{bmatrix} s & t \\ -b/g & a/g \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} g \\ 0 \end{bmatrix}$

Non-scalar Complexity : only count multiplications where both operands depends on input quantities.

Polynomial evaluation : let $f = f_0 + \cdots + f_n x^n \in F[x]$ with degree n be fixed. Patterson&Stockmeyer, for fixed f, $f(\alpha)$ can be computed in $\mathcal{O}(\sqrt(n))$ non-scalar multiplications.

2.8 Polynomial Multiplications

Theorem 2.8.1

Given $a, b \in F[x]$, $deg\, a\,, deg\, b < n$, where n is the length bound for polynomials. Multiplying $a{\cdot}b$ has cost $2n-1$ non-scalar (coefficients of polynomials we want to multiply: $a_0, \cdots, a_{n-1}, b_0, \cdots, b_{n-1}$) multiplications if $\#F \geq 2n-1$.

Idea : Use polynomial evaluation/interpolation

Example 2.8.1

Let $a = 2+3x, b = 1+2x$, have $deg\, ab = 2$, then need 3 evaluation points, let $u_0 = 0, u_1 = 1, u_2 = 2$, we have

$$\begin{aligned} a_{x=0} &= 2 \quad b_{x=0} = 1 \quad (ab)_{x=0} = 2 \\ a_{x=1} &= 5 \quad b_{x=1} = 3 \quad (ab)_{x=1} = 15 \\ a_{x=2} &= 8 \quad b_{x=2} = 5 \quad (ab)_{x=2} = 40 \end{aligned}$$

Let $L_0 = \frac{(x-1)(x-2)}{(0-1)(0-2)}, L_1 = \frac{(x-0)(x-2)}{(1-0)(1-2)}, L_2 = \frac{(x-0)(x-1)}{2-0)(2-1)}$, then

$$\begin{matrix} L_0(0) = 1 & L_0(1) = 0 & L_0(2) = 0 \\ L_1(0) = 0 & L_1(1) = 1 & L_1(2) = 0 \\ L_2(0) = 0 & L_2(1) = 0 & L_2(2) = 1 \end{matrix}$$

Let $c = 2L_0 + 15L_1 + 40L_2 = 2 + 7x + 6x^2$

Proof
Choose $u_0, \cdots, u_{2n-2} \in F$

(1) Evaluate $\alpha_i = a(u_i)$ and $\beta_i = b(u_i)$ for $i = 0, 1, \cdots, 2n-2$

(2) Compute $\gamma_i = \alpha_i \cdot \beta_i$ ($2n-1$ non-scalar multiplications)

(3) Interpolate to get $c = ab$ using Lagrange Formula

$L_i = \prod_{i \neq j} \frac{x-u_j}{u_i-u_j} \in F[x]$ satisfies $L_i(u_k) = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{otherwise} \end{cases}$ and $c = \sum_{0 \leq i \leq 2n-2} \gamma_i L_i$

2.9 Evaluation and Interpolation Related to Matrix-vector Product

Given $a = a_0 + \cdots + a_{n-1}x^{n-1}$, define the **Vandermonde matrix**

$$VDM(u_1, \cdots, u_n) = \begin{bmatrix} u_1^0 & u_1^1 & \cdots & u_1^{n-1} \\ u_2^0 & u_2^1 & \cdots & u_2^{n-1} \\ \vdots & \vdots & & \vdots \\ u_n^0 & u_n^1 & \cdots & u_n^{n-1} \end{bmatrix}$$

Evaluation : compute $\alpha_1 = a(u_1), \cdots, \alpha_n = a(u_n)$, then

$$VDM(u_1, \cdots, u_n) \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} a(u_0) \\ a(u_1) \\ \vdots \\ a(u_{n-1}) \end{bmatrix}$$

Interpolation : recall n evaluation points define a unique polynomial of degree $< n$:

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = VDM(u_1, \cdots, u_n)^{-1} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{n-1} \end{bmatrix}$$

Example 2.9.1

Polynomial multiplication via evaluation/interpolation Working over $F = \mathbb{Z}/(7)$, $f = 2x^2 + 3x + 1$, $g = x^2 + 5x + 2$, choose evaluation points 0,1,2,3,4, then

$$VDM(0,1,2,3,4) \begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 2 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 6 & 1 \\ 1 & 2 \\ 0 & 5 \\ 3 & 3 \end{bmatrix}$$

pointwise multiplication : $(fg)(0) = 2$, $(fg)(1) = 6$, $(fg)(2) = 2$, $(fg)(3) = 0$, $(fg)(4) = 2$

interpolation : $VDM(0,1,2,3,4)^{-1} \begin{bmatrix} 2 \\ 6 \\ 2 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 6 \\ 2 \end{bmatrix}$, then $fg = 2 + 4x + 6x^2 + 6x^3 + 2x^4$

Definition 2.9.1 (primitive n-th root of unity (n-PRU))

Let $n \in \mathbb{N}$ and $w \in F$, F a field. then w is a **primitive n-th root of unity (n-PRU)** if (1) $w^n = 1$. (2) n is a unit in F. (3) $w^k \neq 1$ for $1 \leq k \leq n$. And n is an integer power, n is a unit in F.

Example 2.9.2

$x^{403} + 2x^3 + x + 1 \in \mathbb{Z}_3[x]$, but $1 + 1 + 1 \equiv 0 \mod 3$

Example 2.9.3

(1) $F = \mathbb{C}$, $w = e^{\frac{2\pi i}{8}}$ is an 8-PRU, -1 is a 2-PRU, i is a 4-PRU.

(2) **Fermat Prime**, $m = 2^4 + 1 = 17$, 3 is a 16-PRU in $\mathbb{Z}/(17)$, 13 is a 4-PRU in $\mathbb{Z}/(17)$

Remark

(1) if w is an $n-$PRU, then w^{-1} is also.

(2) if n is even, then w^2 is a $\frac{n}{2}$-PRU.

Example 2.9.4

13 is a 4-PRU in $\mathbb{Z}/(17)$, then $13^{-1} = 4$ is also a 4-PRU in $\mathbb{Z}/(17)$

13^0	13^1	13^2	13^3	13^4
1	13	16	4	1
4^0	4^1	4^2	4^3	4^4
1	4	16	13	1

Let w be a $n-$PRU in F, recall $\underbrace{w^0 = 1, w, w^2, \cdots, w^{n-1}}_{n \text{ distinct elements of } F}, w^n = 1$.

Define $$V(w) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & w & \cdots & w^{n-1} \\ 1 & w^2 & \cdots & w^{2(n-1)} \\ \vdots & \vdots & & \vdots \\ 1 & w^{n-1} & \cdots & w^{(n-1)(n-1)} \end{bmatrix} = VDM(w^0, w^1, \cdots, w^{n-1}).$$

And $$V(w^{-1}) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & w^{-1} & \cdots & w^{-(n-1)} \\ 1 & w^{-2} & \cdots & w^{-2(n-1)} \\ \vdots & \vdots & & \vdots \\ 1 & w^{-(n-1)} & \cdots & w^{-(n-1)(n-1)} \end{bmatrix}$$

Theorem 2.9.1

Let w be an n-PRU, then $V(w) \cdot V(w^{-1}) = nI_n$, where I_n is $n \times n$ identity.

Proof

$$
\begin{aligned}
u &= \left(V(w)\cdot V(w^{-1}\right)_{ij} = (i\text{-th row of } (V(w)) \times (j\text{-th column of } V(w^{-1}) \\
&= \sum_{0\le k<n} w^{ik}w^{-kj} \\
&= \sum_{0\le k<n} (w^{i-j})^k \\
&= \begin{cases} \sum_k 1 & \text{if } i=j \\ \frac{w^{(i-j)n}-1}{w^{i-j}-1} & \text{otw} \end{cases} \\
&= \begin{cases} n & \text{if } i-j \\ 0 & \text{otw} \end{cases}
\end{aligned}
$$

Definition 2.9.2 (Discrete Fourier Transform (DFT))

Let $w \in F$ be an $n-$PRU, then $DFT(w)$(Discrete Fourier Transform) is the linear map $F^n \to F^n$ defined by $\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} \mapsto \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix} = V(w) \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$, where $b_j = \sum_{0\le k<n} a_k w^{jk}$

Let $f = a_0 + \cdots a_k x^k$, consider evaluating $f(1)$ and $f(-1)$

Decompose $f(x) = (\underbrace{a_0 + a_2x^2 + a_4 + x^4 + \cdots}_{f_{even}(x^2)}) + x(\underbrace{a_1 + a_3x^2 + a_5x^4 + \cdots}_{x\cdot f_{odd}(x^2)})$

Then $f(1) = f_{even}(1) + f_{odd}(1)$ and $f(-1) = f_{even}(1) - f_{odd}(1)$

Then evaluate f at ± 1 reduced to evaluating two polynomials of $\frac{1}{2}$ degree at 1. $\frac{1}{2}2 = 1$, but number of evaluation points halved.

Consider starting 4 evaluation points : $1, i = w$ a 4-PRU , $-1, -i = (\pm 1, \pm i)$, evaluating f at $(\pm 1, \pm i)$ reduced to evaluating two polynomials of half the degree at $(1, i)$. So we can apply one step of the recipe whenever we have n evaluation points of the form $(u_1, -u_1), \cdots, (u_{n/2}, -u_{n/2})$.

Theorem 2.9.2

Let n be a power of 2. Let $w \in F$ be an n-PRU. Then $DFT(w)$ can be computed in $\mathcal{O}(n\log n)$ field operations of from F.

Lemma 2.9.1

$w^{n/2+i} = w^i$

Example 2.9.5

$w = 2$ is an 8-PRU in $\mathbb{Z}/(17) = \{-8, -7, \cdots, 0, \cdots, 7, 8\}$

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7
1	2	4	8	-1	-2	-4	-8
4^0		4^1		4^2		4^3	
-1^0				-1^4			

Proof
Our goal is to compute (for $0 \leq k < n$)

$$
\begin{aligned}
f(w^k) &= \sum_{0 \leq j < n} a_j w^{kj} \\
&= \Big(a_0 + a_2(w^{2k})^1 + a_4(w^{2k})^2 + \cdots\Big) + w^k\Big(a_1 + a_3(w^{2k})^1 + a_5(w^{2k})^2 + \cdots\Big) \\
&= f_{even}(w^{2k}) + w^k f_{odd}(w^{2k})
\end{aligned}
$$

where $f_{even} = \sum_{0 \leq j < n/2} a_{2j}x^j$ and $f_{odd} = \sum_{0 \leq j < n/2} a_{2j+1}x^j$

Ans use the fact that $f(x) = f_{even}(x^2) + xf_{odd}(x^2)$ and that $w^{n/2+k} = -w^k$ for $0 \leq k < n/2$.

Computing $DFT(w)(f)$ reduced to

(1) compute $w^2, w^3, \cdots, w^{n-1}$. cost less than n evaluations.

(2) compute $DFT(w^2)(f_{even})$ and $DFT(w^2)(f_{odd})$

(3) $f(w^k) = f_{even}\big((w^2)^k\big) + w^k f_{odd}\big((w^2)^k\big)$ for $k = 0, 1, \cdots, n-1$.

Cost : If $T(x)$ is cost for size n, then $T(n) \leq 2T(n/2) + 3n$, which gives $T(n) \in \mathcal{O}(n \log n)$

Example 2.9.6

$w = 2$ is an 8-PRU in $\mathbb{Z}/(17)$, let $f = a_0 + \cdots + a_7x^7$, try to compute $V(w)\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$

$$
V(w)\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 4 & 8 & -1 & -2 & -4 & -8 \\
1 & 4 & -1 & -4 & 1 & 4 & -1 & -4 \\
1 & 8 & -4 & 2 & -1 & -8 & 4 & 2 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -2 & 4 & -8 & -1 & 2 & -4 & 8 \\
1 & -4 & -1 & 4 & 1 & -4 & -1 & 4 \\
1 & -8 & -4 & -2 & -1 & 8 & 4 & 2
\end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix}
$$

$$
DFT(2)([a_0, \cdots, a_7]) \rightarrow \begin{cases}
DFT(4)([a_0, a_2, a_4, a_6]) \rightarrow \begin{cases} DFT(-1)([a_0, a_4]) \\ DFT(-1)([a_2, a_6]) \end{cases} \\
DFT(4)([a_1, a_3, a_5, a_7]) \rightarrow \begin{cases} DFT(-1)([a_1, a_5]) \\ DFT(-1)([a_3, a_7]) \end{cases}
\end{cases}
$$

Theorem 2.9.3

Let F be a field, $n = 2^k$, w in F an n-PRU, polynomials in $F[x]$ of degree $< \frac{n}{2}$ can be multiplicated using $\mathcal{O}(n \log n)$ field operations.

Proof
Let $a = a_0 + \cdots + a_{n/2-1}x^{n/2-1}$ and $b = b_0 + \cdots + b_{n/2-1}x^{n/2-1}$.

Let $\bar{a} = \begin{bmatrix} a_0 \\ \vdots \\ a_{n/2-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ and $\bar{b} = \begin{bmatrix} b_0 \\ \vdots \\ b_{n/2-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$, our goal is to compute $\bar{c}$ where $c = ab$

$$\bar{c} = \big(DFT(w)\big)^{-1}\big(DFT(w)(a)\cdots DFT(w)(b)\big) \text{ pointwise multiplication for cdot}$$

Use the fact that $DFT(w)^{-1} = \frac{1}{n}DFT(w^{-1})$.

Definition 2.9.3 (support FFT)
We say F supports the FFT is F has a 2^l-PRU for any $l \in \mathbb{N}$.

Chapter 3

From Polynomial Multiplication to Integer Multiplication

3.1 Overview

Theorem 3.1.1
If F supports the FFT, then polynomials of degree at most n can be multiplicated in $\mathcal{O}(n \log n)$ field operations.

Theorem 3.1.2 (Schonhage & Strason)
Integer multiplication can be done in time $\mathcal{O}(n \log n(\log \log n))$

Theorem 3.1.3 (Cantor & Kaltofen, 1991)
Over any ring polynomials of degree n can be multiplied in $\mathcal{O}(n \log n(\log \log n))$ ring operations

Definition 3.1.1 (M(n))
A function $M : \mathbb{N}_{>0} \to \mathbb{R}_{>0}$ is a multiplication time for $R[x]$, R a ring, if polynomials in $R[x]$ of degree $< n$ can be multiplicated using at most $M(n)$ ring operations in R.

- Standard : $M(n) \in \mathcal{O}(n^2)$
- Karatsuba(1960) : $M(n) \in \mathcal{O}(n^{\log_2 3})$
- Cantor & Kaltofen (1991) : $M(n) \in \mathcal{O}(n(\log n)(\log \log n))$
- Schonhage & Strassen (1971) : $M(n) \in \mathcal{O}(n(\log n)(\log \log n))$ word operations.
- Furer(2007) : $\mathcal{O}(n(\log n)k^{\log *n})$
- Harvey & VanderHoeven(2019) : $\mathcal{O}(n \log n)$

Analysis m terms of M adds information to cost estimates.

This is usually very different than naive method

3.2 Useful Assumption about M

Superlinearity :

$$
\begin{aligned}
M(n)/n &\geq M(m)/m \text{ , if } n \geq m \\
M(mn) &\geq mM(n) \\
M(m+n) &\geq M(n) + M(m) \\
M(n) &\geq n
\end{aligned}
$$

At Most Quadratic :

$$M(nm) \leq m^2 M(n)$$

Example 3.2.1

$$M(cn) \in \mathcal{O}(M(n))n^3 + nM(n) \in \mathcal{O}(n^3)$$

3.3 Fast Division With Remainder

Let $a = a_0 + \cdots + a_n x^n$, $b = b_0 + \cdots + b_m x^m \in F[x]$, where $a_n, b_m \neq 0$ and $m \leq n$.

Goal : Find $q(x)$ and $r(x)$ such that $a(x) = q(x)b(x) + r(x)$, $deg\, r < deg\, b$.

Let's assume $b_m = 1$

Reversion : Substitute $x = \frac{1}{y}$ and multiply by y^n. For $a = a_0 + \cdots + a_n x^n$, we have

$$
\begin{aligned}
y^n a(\frac{1}{y}) &= y^n\big(a_0 + a_1(\frac{1}{y}) + \cdots + a_n(\frac{1}{y^n})\big) \\
&= y^n a_0 + y^{n-1} a_1 + \cdots + a_n \\
&:= rev_n(a) = rev(a)
\end{aligned}
$$

Reversion of equation $a(x) + q(x)b(x) + r(x)$, we have

$$
\begin{aligned}
y^n a(\frac{1}{y}) &= y^n\big(q(\frac{1}{y})b(\frac{1}{y}) + r(\frac{1}{y})\big) \\
rev_n(a) &= rev_{n-m}(q) \cdot rev_m(b) + y^{n-m+1} rev_{m-1}(r)
\end{aligned}
$$

Goal : Solve equation for unknown $rev_{n-m}(q)$

$$rev_n(a) = rev_{n-m}(q) \cdot rev_m(b) \mod y^{n-m+1}$$

Example 3.3.1

Over $F[x], F = \mathbb{Q}$, $n = 3, m = 1$,

$$
\begin{aligned}
\underbrace{2x^3 + x^2 + 3x + 4}_{a} &= \underbrace{(q_2 x^2 + q_1 x + q_0)}_{q} \cdot \underbrace{(x-1)}_{b} + \underbrace{r_0}_{r} \\
\underbrace{4y^3 + 3y^2 + y + 2}_{rev_3(a)} &= \underbrace{(q_0 y^2 + q_1 y + q_2)}_{rev_2(q)} \cdot \underbrace{(1-y)}_{rev_1(b)} + y^3 \underbrace{r_0}_{rev_0(r)} \\
rev_2(q) &= (1-y)^{-1}(3y^2 + y + 1) \mod y^3
\end{aligned}
$$

Key idea : now work over a larger field $F[[y]]$(ring of infinite power series in Y, all infinite sums $c_0 + c_1 y + \cdots +$ with $c_i \in F$) instead of $F[y]$ to make sense of $(1-y)^{-1}$. What is $(1-y)^{-1}$ in $F[[y]]$?

Remark

$$\frac{1}{y} = 1 + y + y^2 + \cdots +$$

Example 3.3.2

$$\begin{aligned}
(1)(1-y) &\equiv 1 \mod y \\
(1+y)(1-y) &\equiv 1 \mod y^2 \\
(1+y+y^2)(1-y) &\equiv 1 \mod y^3
\end{aligned}$$

Back to our main equation :

$$\begin{aligned}
rev_2(q) &= (1-y)^{-1}(3y^2 + y + 1) \mod y^3 \\
&= (1+y+y^2)(3y^2+y+1) \mod y^3 \\
&= 2 + 3y + 6y^2 \mod y^3 \\
q &= rev_2\big(rev_2(q)\big) \\
&= 2x^2 + 3x + 6
\end{aligned}$$

Claim 3.3.1

Let $g = g_0 + g_1 x + \cdots \in F[[x]]$ have constant coefficient $g_0 = 1$. For any $h \in \mathbb{Z}_{>0}$, $\exists$ a unique $b \in F[x]$ with $deg\, b < k$ such that

$$bg \equiv 1 \mod x^k$$

Proof

$g = 1 + g_1 x + \cdots + g_{k-1}x^{k-1} \mod x^k$, let $b = b_0 + \cdots + b_{k-1}x^{k-1}$, then

$$(b_0 + b_1 + \cdots + b_{k-1}x^{k-1})(1 + g_1 x + \cdots + g_{k-1}x^{k-1}) \equiv 1 \mod x^k$$

Rewrite it in linear system:

$$\begin{bmatrix} 1 & & & & & \\ g_1 & 1 & & & & \\ g_2 & g_1 & 1 & & & \\ g_3 & g_2 & g_1 & 1 & & \\ \vdots & & & & & \\ g_{k-1} & g_{k-2} & g_{k-3} & g_{k-4} & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & & \\ b_1 & 1 & & & & \\ b_2 & b_1 & 1 & & & \\ b_3 & b_2 & b_1 & 1 & & \\ \vdots & & & & & \\ b_{k-1} & b_{k-2} & b_{k-3} & b_{k-4} & \cdots & 1 \end{bmatrix} = I_k$$

$$\begin{bmatrix} 1 & & & & & \\ g_1 & 1 & & & & \\ g_2 & g_1 & 1 & & & \\ g_3 & g_2 & g_1 & 1 & & \\ \vdots & & & & & \\ g_{k-1} & g_{k-2} & g_{k-3} & g_{k-4} & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 \\ b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{k-1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Using linear algebra, the system has a unique solution for b.

Problem : Given $g = g_0 + \cdots \in F[[x]]$ and $k \in \mathbb{N}$, find $h \in F[[x]]$ such that

$$hg \equiv 1 \mod x^k$$

Let $h_0, h_1, \cdots \in F[[x]]$ be such that

$$deg\, h_i < 2^i \text{ and } h_i g \equiv 1 \mod x^{2^i}$$

Then $h_0 = 1$.

Example 3.3.3

if $g = 1 - x$, then $h_0 = 1\,, h_1 = 1 + x\,, h_2 = 1 + x + x^2 + x^3$

Since $g_0 = 1$, we always have $h_0 = 1$.

How can we compute h_{i+1} from h_i?

Let $g^{-1} \in F[[x]]$ be the inverse of g, then

$$g^{-1} \equiv h_i \mod x^{2^i}$$

Multiply both sides by g, subtract right fro the left, have

$$1 - gh_i \equiv 0, \mod x^{2^i}$$

The LHS is divisible by x^{2^i}, set

$$r_i = \frac{1 - gh_i}{x^{2^i}} \in F[x]$$

Then $1 - gh_i = r_i \cdot x^{2^i}$ and $1 = gh_i + r_i x^{2^i}$, multiply both sides by g^{-1},

$$g^{-1} = h_i + x^{2^i} g^{-1} r_i$$

Take equation modulo $x^{2^{i+1}}$, have

$$\begin{aligned} h_{i+1} &= h_i + x^{2^i} h_i r_i \mod x^{2^{i+1}} \\ &= h_i + h_i(1 - gh_i) \\ &= 2h_i - gh_i^2 \mod x^{2^{i+1}} \end{aligned}$$

We have proven :

$$h_0 = 1,\; h_i = 2h_i - gh_i^2 \mod x^{2^{i+1}} \text{ for } i > 0$$

Picture working modulo $x^{2^{i+1}}$

Theorem 3.3.1

$$h_0 = 1,\; h_i = 2h_i - gh_i^2 = h_i + h_i(1 - gh_i) \mod x^{2^{i+1}} \text{ for } i > 0$$

Example 3.3.4

Given h_1, compute h_2

Let $h_2 = \underbrace{1 + b_1 x}_{=h_i \text{ (have it)}} + \underbrace{b_2 x^2 + b_3 x^3}_{\text{goal computation}} \in F[x]$

Theorem 3.3.2

$$h_2 = h_1 + h_1(1 - g h_1) \mod x^4$$

We rewrite the equation in matrix form:

$$\begin{aligned}
\begin{bmatrix} 1 \\ h_1 \\ h_2 \\ h_3 \end{bmatrix} &= \begin{bmatrix} 1 \\ h_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ h_1 & 1 & & \\ & h_1 & 1 & \\ & & h_1 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & & & \\ g_1 & 1 & & \\ g_2 & g_1 & 1 & \\ g_3 & g_2 & g_1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ h_1 \\ 0 \\ 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} 1 \\ h_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ h_1 & 1 & & \\ & h_1 & 1 & \\ & & h_1 & 1 \end{bmatrix} \left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ c_0 \\ c_1 \end{bmatrix} \right) \\
&= \begin{bmatrix} 1 \\ h_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ h_1 & 1 & & \\ & h_1 & 1 & \\ & & h_1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -c_0 \\ -c_1 \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ h_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & h_1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -c_0 \\ -c_1 \end{bmatrix} \\
G h_2 &= \begin{bmatrix} 1 & & & \\ g_1 & 1 & & \\ g_2 & g_1 & 1 & \\ g_3 & g_2 & g_1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ h_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ g_1 & 1 & & \\ g_2 & g_1 & 1 & \\ g_3 & g_2 & g_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & h_1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -c_0 \\ -c_1 \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ 0 \\ c_0 \\ c_1 \end{bmatrix} + \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & g_1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & h_1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -c_0 \\ -c_1 \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ 0 \\ c_0 \\ c_1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -c_0 \\ -c_1 \end{bmatrix}
\end{aligned}$$

Algorithm 4: Quadratic Newton Iteration

Input: $g \in F[x]$ with $g \mod x = 1\,, n = 2^r$
Output: $h \in F[x]$ such that $hg \equiv 1 \mod x^{2^r}$
$h_0 = 1$;
for $i = 0, 1, \cdots, r-1$ **do**
  $h_{i+1} = (2h_i - gh_i^2) \text{ rem } x^{2^{i+1}}$
end
return h_r;

Theorem 3.3.3

If $n = 2^r$, then $h_r = g^{-1} \mod x^n$ can be computed in $\mathcal{O}(M(n))$ field operations.

Proof
Cost to be compute h_{i+1} is $\leq 2\, M(2^{i+1} + 2 \cdot 2^{i+1}$

Total cost is

$$\begin{aligned} &2\sum_{i=0}^{r-1}\Big(M(2^{i+1}) + 2^{i+1}\Big) \\ =&\underbrace{2\sum_{i=0}^{r-1} M(2^{i+1})}_{S} + \underbrace{2\sum_{i=0}^{r-1} 2^{i+1}}_{<4n} \\ S =& \sum_{i=0}^{r-1} M(2^{i+1}) = M(2) + M(4) + \cdots + M(2^r) \end{aligned}$$

Superlinearity : use this to find that

$$M(ab) \geq aM(b) \text{ for any } a, b \in \mathbb{Z}_{\geq 1}$$

$$2^r = 2^i 2^{r-i} \;\Rightarrow\; M(2^r) \geq 2^{r-i} M(2^i) \;\Rightarrow\; M(2^i) \leq \frac{1}{2^{r-i}} M(2^r)$$

Therefore have

$$\begin{aligned} S &= \sum_{i=1}^{r} M(2^i) \\ &\leq M(2^r)\sum_{i=1}^{r}\frac{1}{2^{r-i}} \\ &= M(2^r)\sum_{i=0}^{r-1}\frac{1}{2^i} \\ &\leq M(2^r)\sum_{i=0}^{\infty}\frac{1}{2^i} \\ &\in \mathcal{O}\big(M(2^r)\big) \end{aligned}$$

Corollary 3.3.1

Let $a, b \in F[x]$ with $deg\, a = n,\ deg\, b = m,\ n \geq m$, then $q = a\, quo\, b$ can be computed in $M(n-m)$ operations from F.

q depends only on leading $n - m + 1$ coefficients of a and b.

Use recursion technique, then Newton iteration for $rev(b)$ over $F[[x]]$.

Corollary 3.3.2

For polynomials of degree at most n in F, division with remainder requires $\mathcal{O}(M(n))$ operations.

Remark
compute $r = a - qb$ in time $\mathcal{O}(M(n))$

What about integers?

Input : $a.b \in \mathbb{Z}$

Output :$q \in \mathbb{Z}$ such that $|a - qb| < |b|$

Reversion does not work because of carries!!

Idea : Use numerical Newton iteration. Compute approximation of $\frac{1}{b}$ over $\mathbb{R}$.

Example 3.3.5

$a = 34283749279327 42$, $b = 13432422423$, $a\, quo\, b$ will have about 7 decimal digits.

$$g := \frac{b}{10^{11}} \cong 0.1343242242$$

$$\begin{aligned}
h &= 10 & h \cdot g &= 1.343242242 \\
h &= 2h - gh^2 = 8.88218476133 & h \cdot g &= 0.8821847633 \\
h &= 2h - gh^2 = 7.341338287 & h \cdot g &= 0.9861185700 \\
h &= 2h - gh^2 = 7.443238215 & h \cdot g &= 0.999807331 \\
h &= 2h - gh^2 = 7.444673281 & h \cdot g &= 0.999999629
\end{aligned}$$

Should be good enough
$$\frac{7.44467321}{10^{11}} \cdot a = \underbrace{2552323}_{=quo\,(a,b)}.122$$

Corollary 3.3.3

Can do arithmetic in $R = F[x]/\big(p(x)\big)$ and $\mathbb{Z}/(p)$ in time

- $\mathcal{O}(M(deg\, p))$ field operations from F, or
- $\mathcal{O}(M(deg\, p))$ word operations from $\mathbb{Z}$.

3.4 p-adic Inversion Using Newton iteration

Algorithm 5: Integer Newton Iteration

Input: $f, g_0 \in R$ with $fg_0 \equiv 1 \mod p, l \in \mathbb{N}$
Output: $g \in R$ such that $gf \equiv 1 \mod p^l$
$r = \lceil \log l \rceil$;
for $i = 1, \cdots, r$ **do**
  $g_i = (2g_{i-1} - fg_{i-1}^2) \bmod p^{2^i}$
end
return g_r;

Example 3.4.1

$R = \mathbb{Z}$, compute inverse of 5 modulo 6561=3^8.

start with $g_0 = 1$ since $-1 \cdot 5 = 1 \mod 3$

$$
\begin{aligned}
g_1 &= 2g_0 - 5g_0^2 = 2 \mod 3^2 & 2 \cdot 5 &\equiv 1 \mod 3^2 \\
g_2 &= 2g_1 - 5g_1^2 = -16 \mod 3^4 & -16 \cdot 5 &\equiv 1 \mod 3^4 \\
g_3 &= 2g_2 - 5g_2^2 = -1312 \mod 3^8
\end{aligned}
$$

Method 2 : Euclidean

Compute $\begin{bmatrix} s & t \\ u & v \end{bmatrix} \begin{bmatrix} 5 \\ 6561 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Chapter 4

The Chinese Remainder Algorithm

4.1 Overview

R is a Euclidean Domain, let $m = m_0 \cdots m_{r-1}$, where $gcd(m_i, m_j) = 1, i \neq j$.

gcd condition $\Rightarrow m = lcm(m_0, \cdots, m_{r-1})$

Theorem 4.1.1 (Chinese Remainder Theorem)

$$\frac{R}{(m)} \cong \frac{R}{(m_0)} \times \frac{R}{(m_1)} \times \cdots \times \frac{R}{(m_{r-1})}$$

Example 4.1.1
$m = 7 \times 11 \times 13 = 1001$, $a = 1234$, then

$$\begin{array}{ccccr} \mathbb{Z} & \mapsto & \mathbb{Z}/((1001) & \cong & (\mathbb{Z}/(7), \mathbb{Z}/(11), \mathbb{Z}/(13)) \\ a & \mapsto & rem(a,m) & \mapsto & \big(rem(a,m_0), rem(a,m_1), \cdots, rem(a,m_{r-1})\big) \\ 1234 & \mapsto & 233 & \cong & (2,2,12) \end{array}$$

Example 4.1.2
If $m = 1001 = \times 11 \times 13$, $a = 233 \mod m$, $b = 365 \mod m$, then $a \mapsto (2,2,12)$ and $b \mapsto (1,2,1)$. then

$$\begin{aligned} rem(a+b,m) &: (2,2,12) + (1,2,1) = (3,4,0) \mapsto 598 \mod 1001 \\ rem(a*b,m) &: (2,2,12) * (1,2,1) = (2,4,12) \mapsto 961 \mod 1001 \end{aligned}$$

Goal: Given $v_0, v_1, \cdots, v_{r-1} \in R$, find an $f \in R$ such that $f \mod m_i \equiv v_i$ for $0 \leq i < r$.

$m = m_0 \cdots m_{r-1}$ and $f = v_0 s_0(\frac{m}{m_0} + v_1 s_1(\frac{m}{m_1}) + \cdots + v_{r-1}s_{r-1}(\frac{m}{m_{r-1}})$.

Can we construct the s_i's such that f is correct?

Consider $f \mod m_0$

- all terms $m/m_1, \cdots, m/m_{r-1}$ will vanish
- want $v_0 s_0(\frac{m}{m_0}) \equiv v_0 \mod m_0$, i.e. $s_0(\frac{m}{m_0}) \equiv 1 \mod m_0$

- we can choose s_0 such that $s_0(\frac{m}{m_0}) + (*)m_0 = 1$

Example 4.1.3
$m_0, m_1, m_2 = 7, 11, 13, v_0, v_1, v_2 = 2, 2, 12$

$$\begin{aligned}
&gcd(11 * 13, 7) = 1 = (-2)(11 \times 13) + (41)(7) = 1 \Rightarrow && L_0 = -2 \times (11 \times 13) = -286 \\
&gcd(7 \times 13, 11) = 1 = \cdots, && L_1 = 4 \times (7 \times 13) = 364 \\
&gcd(7 \times 11, 13) = 1 = \cdots, && L_2 = -1 \times (7 \times 11) = -77
\end{aligned}$$

So possibly f is given by

$$\begin{aligned}
f &= \underbrace{2}_{v_0} + \underbrace{(-2)}_{s_0} \times \underbrace{(11 \times 17)}_{m/m_0} + \underbrace{2}_{v_1} \times \underbrace{4}_{s_1} \times \underbrace{(7 \times 17)}_{m/m_1} + \underbrace{12}_{v_2} \times \underbrace{(-1)}_{s_1} \times \underbrace{(7 \times 11)}_{m/m_2} \\
&= -768 \\
&\equiv 233 \mod 1001
\end{aligned}$$

4.2 Small Refinement to Algorithm

Compute $c_i = v_i \cdot s_i \; rem \; m_i$.

$$\begin{aligned}
c_0 &= 2 \times (-2) \mod 7 = 3, , c_1 = 8, c_2 = 1 \\
f &= 3 \times (11 \times 13) + 8 \times (7 \times 13) + 1 \times (7 \times 11) \\
&= 1234 \\
&\equiv 223 \mod 1001
\end{aligned}$$

Claim 4.2.1
if each v_i in range $0, \cdots, M_i - 1$, then the cost is $\mathcal{O}\big((\log m)^2\big)$ word operations.

(a) compute $m = m_0 \cdots m_{r-1}$

(b) compute m/m_i for $0 \le i < r$

(c) compute s_i such that $s_i(\frac{m}{m_i}) + *m_i = 1$

(d) compute f

Each step bounded by $\mathcal{O}\big((\log m)^2\big)$ word operations.

$$rem(a, m) \mapsto \big(rem(a, m_0), rem(a, m_1), \cdots, rem(a, m_{r-1})\big)$$

Theorem 4.2.1
Both directions of CRT can be computed in time $\mathcal{O}\big((\log m)^2\big)$ bit operations.

4.3 Negative Numbers

- CRT still holds.
- Just change "system of representatives" modulo m

Example 4.3.1
modp and **mods** in Maple

Positive range : $0 \leq modp(a, m) \leq m - 1$

Symmetric range : $-\lfloor \frac{m-1}{2} \rfloor \leq mods(a, m) \leq \lfloor \frac{m}{2} \rfloor$

$mods(*, 7)$ maps to $\{-3, -2, -1, 0, 1, 2, 3\}$

$mods(*, 6)$ maps to $\{-2, -1, 0, 1, 2\}$

4.4 Variations of Chinese Remaindering

$rem(a, m_0)$, $rem(a, m_0, m_1)$, $rem(a, m_0, m_1, m_2), \cdots$

4.5 Matrix Radix Representator

Let $0 \leq a < m_0 m_1 \cdots m_{r-1} m_r m_i \in \mathbb{N}$, $m_i \in \mathbb{N}_{\geq 2}$(not necessarily real prime).

Claim 4.5.1
We can write a uniquely as

$$a = a_0 + a_1 m_0 + a_2 m_0 m_1 + \cdots + q_r m_0 \cdots m_{r-1}$$

with $0 \leq a_i < m_i$ for all i

This is a mixed radix representation

Example 4.5.1
$m_0 = 7, m_1 = 11, m_2 = 13$, $233 = 2 + (0)((7) + (3)(7 \times 11)$

4.6 Incremental Chinese Remaindering

Compute $\underbrace{rem(f, m_0)}_{v_0}$, $\underbrace{rem(f, m_0 m_1)}_{v_0, v_1}$, $\underbrace{(f, m_0 m_1 m_2)}_{v_0, v_1, v_2}$

Input : $M, m \in \mathbb{Z}$ with $m \perp M$ and $V, v \in \mathbb{Z}$, $0 \leq V < M, 0 \leq v < m$.

Output : $f \in \mathbb{Z}$ such that $0 \leq f < Mn$, $f \equiv V \mod M$ and $f \equiv v \mod m$. e.g. $M = m_0 m_1 \cdots m_{r-1}, m = m_r$

Method 1 (A2Q1)

Let s, t be such that $sM + tm = 1$

Return $modp\big(v \times (s \times M) + V \times (t \times m), m \times M\big)$

Chapter 5

Fast Interpolation and Evaluation

5.1 CRT revisited

We can regard CRT as an incremental reduction.

Picture!!!!

5.2 Recall Lagrange

$a = rem(f, m)$ where **Picture**

5.3 Fast Multi-point Evaluation

Given $f = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \in R[x]$ with degree $< n$ and $u_0, \cdots, u_{n-1} \in R$.

Find $f(u_0), \cdots, f(u_{n-1})$

Evaluation Another View

Recall for $f \in F[x]$, $rem(f, x-u) = f(u)$

Example 5.3.1
$(x^2 + 2x + 3)|_{x=1} = 6 = rem(x^2 + 2x + 3, x - 1)$

$$\begin{array}{r}
X + 3 \\
X - 1 \overline{\smash{\big)}\; X^2 + 2X + 3} \\
\underline{- X^2 \;\; + X \phantom{{}+3}} \\
3X + 3 \\
\underline{- 3X + 3} \\
6
\end{array}$$

$m = (x - u_0)\cdots(x - u_{n-1}) \in R[x]$, $u_i \neq u_j$ for $i \neq j$

$$\begin{array}{lcl} F[x]/(m) & \to & (F, \cdots, F) \\ \text{Evaluation} & \mapsto & (f(u_0), \cdots, f(u_{n-1})) \\ \text{Interpolation} & \hookleftarrow & (v_0, \cdots, v_{n-1}) \end{array}$$

Assume $n = 2^k$ for some k and degree of $f < n$

Main idea : Product tree

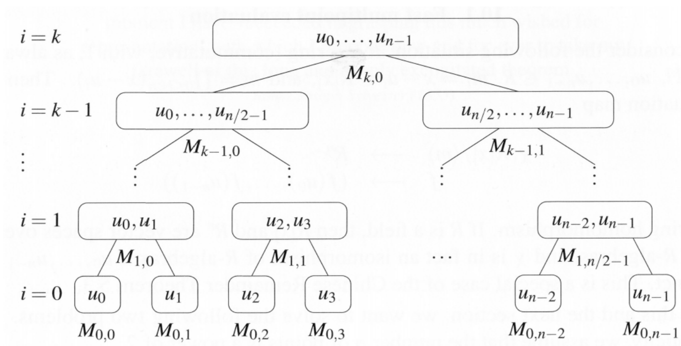

Figure 5.1: Product Tree

$$m = \underbrace{(m_0 m_1 \cdots m_{n/2-1})}_{M(k-1,0)} \underbrace{(m_{n/2} \cdots m_{k-1})}_{M(k-1,1)}$$

Cost of computing all the nodes?

$$\sum_{i=1}^{k} \frac{n}{2^i} M(2^i)$$

Superlinearity

- $M(n) \geq n$ • $M(n+m) \geq M(n) + M(m)$ • $M(mn) \geq mM(n)$

Cost of going down the subproduct tree?

Example 5.3.2

$rem(F, m_0) = rem\Big(rem\big(\cdots rem(F, M(k-1,0)) \cdots, m_0 m_1 m_2 m_3\big), m_0 m_1, m_0\Big)$

Let c be such that $cM(n)$ operations are sufficient to divide a polynomial of length $2n$ with a polynomial of degree n.

$$c \sum_{i=1}^{k} 2^{k-i} M(2^i) \in \mathcal{O}\big(M(n) \log n\big)$$

5.4 Recall Lagrange Interpolation

$m = \underbrace{(x-u_0)}_{m_0}\cdots\underbrace{(x-u_{n-1})}_{m_{n-1}}$

Let

$$s_i = (\tfrac{m}{m_i}|_{x=u_i})^{-1}$$
$$f = \underbrace{v_0 s_0}_{c_0}(m/m_0) + \cdots + \underbrace{v_{n-1}s_{n-1}}_{c_{n-1}}(m/m_{n-1})$$

Idea : Use Product Tree

$$\begin{aligned}
&(c_0, c_1, c_2, c_3, \cdots, c_{n-2}, c_{n-1}) && [n \text{ tuple}]\\
&(M_0c_1 + M_1c_0, M_2c_3 + M_3c_2, \cdots, M_{n-2}c_{n-1} + M_{n-1}c_{n-2}) && [n/2 \text{ tuple}]\\
&(M_2M_1(M_2c_3 + M_3c_2) + M_1M_2(m_0c_1 + m_1c_0), \cdots)
\end{aligned}$$

Example 5.4.1
Picture!!

Cost : $\mathcal{O}\big(M(n)\log n\big)$ field operations.

Question : how to recover the s_i's?

Consider formal derivative $m' = \frac{m}{x-u_0} + \cdots + \frac{m}{x-u_{n-1}}$

Example 5.4.2

$$Diff\big((x-1)(x-2)(x-3), x\big) = (x-2)(x-3) + (x-1)(x-3) + (x-1)(x-2)$$

Which gives $m'(u_i) = (\frac{m}{x-u_i})|_{x=u_i} = \frac{1}{s_i}$

Evaluation m' at n points $u_0, u_1, \cdots, u_{n-1}$ and repeat, which cost $\mathcal{O}\big(M(n)\log n\big)$

5.5 Fast Multi-modular Reduction

Let $m = m_0m_1\cdots m_{r-1}$ with $r = 2^k$, $n = deg\, m$, we can consider constructing a similar subproduct tree (same as Figure 5.1). The cost of computing nodes at level i from level $i-1$ is

$$\sum_d M(d) \le M(\sum_d d) = M(deg\, m)$$

Overall cost is $\mathcal{O}(M(n)\log r)$. Worst case $n = deg\, m$

5.6 Fast Chinese Remaindering

Let $s_i = rem((\frac{m}{m_i})^{-1}, m_i)$, $f = \underbrace{rem(v_0s_0, m_0)}_{c_0}\frac{m}{m_0} + \underbrace{rem(v_1s_1, m_1)}_{c_1}\frac{m}{m_1} + \cdots + \underbrace{rem(v_{r-1}s_{r-1}, m_{r-1})}_{c_{r-1}}\frac{m}{m_{r-1}}$

Question : How to get s_i's ?

Remark
$a \equiv b \mod m_i \iff am_i \equiv bm_i \mod m_i^2$

This is a neat trick to compute the s_i's

$$rem\big((\frac{m}{m_i})^{-1}, m_i\big) = rem\Big((\frac{rem(m, m_i^2)}{m_i})^{-1}, m_i\Big)$$

So we can use multi-modular reduction $m_0^2, m_1^2, \cdots, m_{r-1}^2$

Example 5.6.1

$rem(210, 44100) = 210$, $rem(210, 1225) = 210$, $rem(210, 49) = 14$, $\frac{14}{7} = 2$

Then $modp(\frac{1}{2}, 7) = 4 \ \Rightarrow \ 4 * (2 \cdot 3 \cdot 5) \mod 7 = 1$

5.7 Complexity Summary

Multiplication Time

- Over $R[x]$, n bound on degree, ring operations from R
- Over $\mathbb{Z}$, n bound on word-size, word operations (same as bit operations)

$$M(n) \in \mathcal{O}(n^2), \mathcal{O}(n^{1.59}), \mathcal{O}(n^{1+\epsilon}), \mathcal{O}\big(n \log n (\log\log n)\big)(\text{ for any } R)/\mathcal{O}(n \log n)(\text{ over } \mathbb{Z}, \text{ New })$$

- Superlinear, at most quadratic

1) $n \leq M(n)$ 2) $M(n) + M(m) \leq M(n+m)$ 3) $mM(n) \leq M(mn) \leq m^2 M(n)$

In time $\mathcal{O}(M(n))$, multiplication, inversion over $F[[x]]$ modulo x^n, division with remainder (over $F[x]$ and $\mathbb{Z}$)

$$nM(n^2) \text{ V.S } n^2 M(n)$$

5.8 Fast EEA

$sa + tb = y$, recall

$$\underbrace{\begin{bmatrix} & 1 \\ 1 & -4 \end{bmatrix}}_{Q_3} \underbrace{\begin{bmatrix} & 1 \\ 1 & -2 \end{bmatrix}}_{Q_2} \underbrace{\begin{bmatrix} & 1 \\ 1 & -1 \end{bmatrix}}_{Q_1} \begin{bmatrix} 91 \\ 63 \end{bmatrix} = \begin{bmatrix} 7 \\ \ \end{bmatrix}, \quad \frac{91}{63} = 1 + \frac{1}{2 + \frac{1}{4}}$$

Assume WLOG that $|b| \leq |a|$ and $n = \log a$

- Schonhage 72 : Compute all Q_i matrices in time $\mathcal{O}(M(n)\log n)$
- Main ingredient is **"Half-gcd" algorithm** :

Input : a, b both n bits long

Output : Unimodular $\underbrace{\begin{bmatrix} * & * \\ * & * \end{bmatrix}}_{U} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}$ down to at most $n/2$ bits long

Sketch of Half-gcd Algorithm

Write $a = a_1 2^{n/2} + a_0, b = b_1 2^{n/2} + b_0$

Compute "half-gcd" : $\begin{bmatrix} s_1 & t_1 \\ u_1 & v_1 \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} c_1 \\ d_1 \end{bmatrix}$ about $n/4$ bits long

get second subproblem $\begin{bmatrix} s_1 & t_1 \\ u_1 & v_1 \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} c_0 \\ d_0 \end{bmatrix}$ bounded by $3n/4$ bits long

Second Recursive call with next $n/2$ bits (combine subproblems to get U)

Let $H(n)$ be time for half-gcd

$$H(n) = 2H(\frac{n}{2}) + \mathcal{O}(M(n)) \;\Rightarrow\; H(n) \in \mathcal{O}(M(n)\log n)$$

Let $T(n)$ be the time to solve EEA

$$T(n) = T(\frac{n}{2} + \mathcal{O}(M(n)\log n) \;\Rightarrow\; T(n) \in \mathcal{O}(M(n)\log n)$$

5.9 "GCD-like" Operations

Introduce ad-h0c cost function

$$B(n) \in \mathcal{O}(M(n)\log n) \;\text{ OR }\; B(n) \in \mathcal{O}(n^2), \mathcal{O}(n^{1.59}), \mathcal{O}(n^{1+\epsilon})$$

In time $\mathcal{O}(B(n))$: extended gcd, evaluation/interpolation, multi-modular reduction Chinese remaindering, radix conversion(New), rational number reconstruction (New)

5.10 Radix Conversion

Given $f, p \in F[x]$, $deg\, f = n$, $deg\, p = m$, $k = \lfloor \frac{n}{m} \rfloor + 1$, find $a_0, \cdots, a_{k-1} \in F[x]$ such that

$$f = a_0 + a_1 p + \cdots + a_{k-1} p^{k-1}$$

with $deg\, a_i < deg\, p$.

Similar for $\mathbb{Z}$

Example 5.10.1
$a = 1234 = 2 + 1(7) + 4(7^2) + 3(7^3), p = 7$

base7:[2,1,4,3], base10:[4,3,2,1]

In Maple : convert (1234,base 7) to [2,1,4,3]

Let $f \in R[x]$ and monic $p \in R[x]$, $deg\, f = n$, $deg\, p = m$. Let $k = 2^t$, t minimal such that $2^t\, deg\, p \geq \lfloor \frac{n}{m} \rfloor + 1$

$$\begin{aligned} f &= a_0 + a_1 p + \cdots + a_{k-1}p^{k-1} \\ &= a_0 + \cdots + a_{k/2-1}p^{k/2-1} + \cdots + a_{k-1}p^{k-1} \\ &= \underbrace{(a_0 + \cdots + a_{k/2-1}p^{k/2-1})}_{rem(f,p^{k/2})} + p^{k/2} \underbrace{(a_{k/2} + \cdots + a_{k-1}p^{k/2-1})}_{quo(f,p^{k/2})} \end{aligned}$$

We can precompute $p, p^2, \cdots, p^{k/2}$ and solve the recurrence $T(n) \leq 2T(\frac{n}{2}) + \mathcal{O}(M(n))$

$$T(n) \in \mathcal{O}(B(n))$$

5.11 Rational Number Reconstruction

Given an image $a \in \mathbb{Z}$, a modulo $M \in \mathbb{Z}_+$, bounds $N, D \in \mathbb{Z}_+$ such that $2ND < M$

Find : signed fraction $n/d \in \mathbb{Q}$ with $n \perp d$ and $a = \frac{n}{d} \mod M$ with $|a| < N, d \leq D$

Notice if soln exists, it is unique

Cost : $\mathcal{O}(B(\log M))$ word operations

Example 5.11.1
$\frac{1234}{56789} = 400799450 \mod 10^{10}$

In Maple : **iratrecon**($\underbrace{4007994506}_{a}, \underbrace{10^{10}}_{M}, \underbrace{10^4}_{N}, 10^5$) gives $\frac{1234}{56789}$

5.12 Computation in Ring $\mathbb{Z}/<p>$

- $a + b$, $a - b$: $\mathcal{O}(\log p)$
- $a * b$: $\mathcal{O}(M(\log p))$
- Cost of a^{-1} : $\mathcal{O}(B(\log p))$.

Chapter 6

Exact Linear Algebra Over $\mathbb{Z}$ $\mathbb{Q}$ $\mathbb{Z}$x

6.1 Motivation

Computing expected hitting time in absorbing Markov Chains

Input : $A \in \mathbb{Z}[x]^{186\times 186}$, $deg\, A = 11$. All entries of the form $cx^k(1-x)^{1-k}$, $0 \leq k \leq 11, c \in \mathbb{Z}$

Output :$\alpha \in [0,1]$ such that $f(\alpha)$ is minimized, where $f \in \mathbb{Q}(x)$ is the sum if all entries in $(I-A)^{-1}$

Remark
Sum of all entries in $(I-A)^{-1}|_{x=\alpha}$ = Sum of all entries in $(I + A + A^2 + \cdots)|_{x=\alpha}$

Method 1 (fail)

- Sub different values of x into $(I-A)$, e.g. $B = (I-A)_{x=0.3}$
- Compute $I + B + B^2 + \cdots$ to high enough precision
- Sum up all entries to get an approximation of $f(0.3)$

Method 2 (fail)

- Compute $(I-A)^{-1}$ explicitly. e.g. if $n = 2, B + (I-A)^{-1} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$
- Get $f(x)$ = sum of entries

Method 3

Note that $f(x) = \begin{bmatrix} 1 & 1 & 1 & \cdots 1 \end{bmatrix} \left((I-A)^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right)$

e.g. $\begin{bmatrix} 1 & 1 \end{bmatrix} \left(\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} b_{11} + b_{12} \\ b_{21} + b_{22} \end{bmatrix} = b_{11} + b_{12} + b_{21} + b_{22}$

Use evaluation/ interpolation to recover

$$f(x) = \frac{N(x)}{D(x)},\ N(x), D(x) \in \mathbb{Z}[x]$$

Example 6.1.1
$D(2) = \det(I - A)_{x=2}$, $N(2) = \begin{bmatrix} 1 & 1 & \cdots 1 \end{bmatrix} \left((I-A)^{-1}|_{x=2}\right) D(2) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

6.2 Integer Matrix Determinant

Given $A \in \mathbb{Z}^{n\times n}$

Goal : $\det A$

What is $\det A$?

$$\det A = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i=1}^{n} A_{i,\sigma(i)}$$

$$|\det A| = \text{ volume of parallelepiled spanned by rows (columns) of } A$$

Example 6.2.1
$A_1 = \begin{bmatrix} 5 & \\ & 5 \end{bmatrix}$

$A_2 = \begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}$

- A non-singular over $\mathbb{Q}$ iff $\det A \neq 0$
- $\det A \neq 0 \ \Rightarrow Ax = b$ has exactly one solution

Cramer's Rule: Let $x = A^{-1}b = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, then $x_i = \frac{\det B_i}{\det A}$ where B_i is A with column i replaces by b

Example 6.2.2
!!!!

How big is $\det A$?

Hadamard's Bound : volume maximized when vectors orthogonal

$$|\det A| \leq \prod_{i=1}^{n} ||\text{row}_i\, A||$$

Where $||\text{row}_i\, A|| = (A_{i1}^2 + \cdots + A_{in}^2)^2$

We let $||A|| = \max |A_{ij}|$

Lemma 6.2.1

$$|\det A| \leq n^{n/2}||A||^n$$

Corollary 6.2.1

$A \in \mathbb{Z}^{n\times n}$ nonsingular, $b \in \mathbb{Z}^{n\times 1}$, then

- denominators in $A^{-1}b$ bounded by $n^{n/2}||A||^n$
- numerators in $A^{-1}b$ bounded in magnitude by $n^{n/2}||A||^{n-1}||b||$

Remark

- word length of entries in A is $\log ||A||$
- word length of $\det A$ is $\mathcal{O}(n \log ||A|| + n \log n)$

Computing $\det A$: Gaussian elimination over $\mathbb{Q}$

(e1) multiply a row of A by $c \neq 0 \Rightarrow \det A \to c \det A$

(e2) swap two different rows $\Rightarrow \det A \to -\det A$

(e3) add a multiple of one row to another $\Rightarrow \det A$ unchanged

Example 6.2.3

$$\begin{bmatrix} -34 & 4 & -34 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix} \to \begin{bmatrix} 1 & -\frac{2}{17} & 1 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix} \to \begin{bmatrix} 1 & -\frac{2}{17} & 1 \\ & 1 & \frac{68}{29} \\ & \frac{270}{17} & 24 \end{bmatrix} \textit{to} \begin{bmatrix} 1 & -\frac{2}{17} & 1 \\ & 1 & \frac{68}{29} \\ & & \frac{84}{29} \end{bmatrix}$$

Then $\det A = (-384/29 \times (-34) \times (174/17) = 4608$

$$\begin{bmatrix} -34 & 4 & -34 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix} \to \begin{bmatrix} -34 & 4 & -34 \\ 0 & -384 & -816 \\ 33 & 12 & 57 \end{bmatrix} \text{(row2=-34row2-19row1)}$$

6.3 Single Modular Approach

Choose a single number P such that $P > 2|\det A|$

Remark

If $P > 2|a|$, then $a = mods(a)$

- mop $A \in \mathbb{Z}^{n\times n} \Rightarrow A \in \mathbb{Z}^{n\times n}/(P)$
- compute $\det A$ over $\mathbb{Z}/(P)$
- reduce in symmetric range

Example 6.3.1
Hadamard's bound for $A = \begin{bmatrix} -34 & 4 & -34 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix}$: $|\det A| \leq \lfloor 3^{3/2} 57^3 \rfloor = 962291 := \beta$

$2\beta = 1924582$ choose $P = 1924619$ (first prime larger than 2β)

$$A = \begin{bmatrix} -34 & 4 & -34 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix} \rightarrow (modp) \begin{bmatrix} 1924585 & 4 & 1924585 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix}$$

$1415161 \times 1924585 \equiv 1 \mod P$

$$\begin{bmatrix} -34 & 4 & -34 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix} \rightarrow (modp) \begin{bmatrix} 1924585 & 4 & 1924585 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1811406 & 1 \\ 19 & 8 & 43 \\ 33 & 12 & 57 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1811406 & 1 \\ & 226436 & 24 \\ & 1811422 & 24 \end{bmatrix}$$

Then $mods\big((226436 * 24 - 1811422 * 24)1924585, P\big) = 4608$

Cost: If $\log P \in \mathcal{O}\big(n(\log n + \log ||a||)\big)$, the cost is

$$\mathcal{O}\Big(n^3 M\big(n(\log n + \log ||A||)\big) + nB\big(n(\log n + \log ||a||)\big)\Big) \text{ word operations}$$

Assuming standard arithmetic gives

$$\mathcal{O}\big(n^5(\log n + \log ||A||)^2\big)$$

Total size of input : $\mathcal{O}(n^2 \log ||A||)$

Total size of output : $\mathcal{O}\big(n(\log n + \log ||A||)\big)$

Intermediate size : $\mathcal{O}\big(n^3(\log n + \log ||A||)\big)$, which makes it a bad approach as it is larger than input output size

About A2:

Q3: if working modulo p be sure to use Maple's "Inert" function

Example 6.3.2
RemExpand, Normal, Gcd, Gcdex

Gcd(f,g,x) mod P_i

The command Gcdex(f,g,x,'s','t') mod p will assign a and t such that $sf + tg \equiv gcd(f, g) \mod p$

(a) Newton iteration steps showed look like $h_1 := Rem(Expand(\cdots) \mod p, x^2, x) \mod p;$

(b) $revf := Normal(subs(x = \frac{1}{x}, f) * x^n) \mod p;$

(c) Go ahead and use Gcdex as described above

Q1: see section6.1. You know result is in range $[0, M_m]$

Summary of Single Modulo Approach :

- $B = \lfloor n^{n/2} \rfloor ||A||^n$, $p =$ nextprime (2B+1)
- map $A \in \mathbb{Z} \Rightarrow Ap \in \mathbb{Z}_p$
- compute det Ap over $\mathbb{Z}_p$
- reduce in symmetric range

Cost: $\mathcal{O}\Big(n^3 M\big(n(\log n + \log ||A||)\big) + nB\big(n(\log n + \log ||A||)\big)\Big)$

If $M(n) = n^2$, then $\mathcal{O}\big(n^5(\log n + \log ||A||)^2\big)$ word operations. (large intermediate space)

6.4 Multiple "Small" Modulii Approach

- Choose $p = p_1 p_2 \cdots p_k$ such that $p > 2B$
- For $i = 1 \cdots k$, compute $A^i = rem(A, p_i)$, $d^i = det\, A^i$ over $\mathbb{Z}_{p_i}$
- Chinese Remainders to get d such that $d \equiv det\, A \mod p$
- Reduce d in symmetric range

Lemma 6.4.1
Let $\beta \in \mathbb{Z}_+$, if $l = 6 + \ln\ln\beta$, then $\prod_{2^{l-1}<p<2^l} p > 2\beta$

Example 6.4.1
$\beta = 10^{10^{12}}$, $\log_1 0\beta = 10^{12}$, i.e. β has about a terabyte of decimal digits, which needs about 50 GB to store.

$6 + \ln\ln 10^{12} < 35$, i.e. 35 bit prime numbers.

Thus the lemma says we can choose

- $\log p_i \in \Theta(\log n + \log\log ||A||)$
- $K \in \mathcal{O}\big(\frac{n(\log n + \log ||A||)}{\log n + \log\log ||A||}\big)$

Or (to simplify analysis), choose

- $\log p_i \in \Theta(\log n + \log ||A||)$
- $K \in \Theta(n)$

Compute d_i with $\mathcal{O}(n^3(\log n + \log ||a||)^2)$ word operations, reduction $\mod p_i$ has no effect since p_i is large.

Chinese Remainder images $\mathcal{O}\big((n(\log n + \log ||A||))^2\big)$ word operations.

Total cost (using standard arithmetic) : $\mathcal{O}\big(n^4(\log n + \log ||A||)^2\big)$

6.5 Non-singular System (Rationals) Solving

Input : $A \in \mathbb{Z}^{n\times n}$ and $b \in \mathbb{Z}^{n\times 1}$, $\det A \neq 0$. WLOG, if A is rational, we can scale to $\mathbb{Z}$

Output : Compute $A^{-1}b \in \mathbb{Q}^{n\times 1}$

Recall: $(\det A)A^{-1}b$ is over $\mathbb{Z}$ and $||(\det A)A^{-1}b|| \leq n^{n/2}||A||^{n-1}||b||$

Let $\alpha = \max(||A||, ||B||)$ for simplicity. Let $\beta = \lfloor n^{n/2}\alpha^n \rfloor$

6.6 Solving via Chinese Remaindering

(1) Choose small primes $p_1, \cdots, p_k$ such that $p = p_1 \cdots p_k > 2\beta$

• $\log p_i \in \Theta(\log n + \log \alpha)$ and $K \in \Theta(n)$

(2) Compute $(modp(\det A, p_i),\ modp((\det A)A^{-1}b, p_i))$ for $i = 1, \cdots, k$

• issue: bad primes make A^{-1} singular, $\Theta(n)$ bad primes, choose more primes.

(3) Chinese Remainder to get $(\det A, (\det A)A^{-1}b)$

Overall Cost : $\mathcal{O}(n^4(\log n + \log \alpha)^2)$ word operations using standard arithmetic.

$$\mathcal{O}(n^4(\log n + \log ||A|| + \log ||b||/n)^2) \text{ word operations}$$

Potential Problem : A might become singular modulo p_i

Example 6.6.1
Assume $Ax = b \Rightarrow \begin{bmatrix} 5 & 2 \\ 2 & 2 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, where $d = \det A = 6$, $u = (\det A)A^{-1}b = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$

Computing using above approach :

$p_1, p_2 = 3, 5,\ A^{(1)} = A \mod 3 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \in \mathbb{Z}_3^{2\times n}$, $d^{(1)} = \det A^{(1)} = 0 \in \mathbb{Z}_3$

$u^{(1)} = adj(A^{(1)})b \in \mathbb{Z}_3^{n\times 1} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, where $adj(A)$ always exists

$$adj(A) = \begin{cases} (\det A)A^{-1} & \text{if } A \text{ is invertible} \\ 0_{n\times n} & \text{if rank } \leq n-2 \\ \text{nonzero matrix of rank } < n & \text{if rank } A = n-1 \end{cases}$$

Example 6.6.2
$A = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}$, then $adj(A) = \begin{bmatrix} 6 & -4 \\ -3 & 2 \end{bmatrix}$

Idea: fix bad primes by replacing $(\det A)A^{-1}$ by $adj(A)$, keep the result in Chinese Remainder such as x are considered valid solutions.

6.7 Solving via Power Series Inversion

Given : $f = f_0 + f_1x + \cdots \in F[x]$, $n \in \mathbb{Z}_+$

Compute : $g = g_0 + \cdots g_{n-1}x^{n-1} \in F[x]$ such that $gf \equiv 1 \mod x^n$

Newton Iteration : For $i = 0, \cdots, \lceil \log_2 n \rceil$, compute $h_i \in F[x]$ s.t. $h_i f \equiv 1 \mod x^{2^i}$

Linear Variant : For $i = 0, 1 \cdots, n-1$, compute $g_0 = \frac{1}{f_0}$ and $g_i =$(A2Q4) for $i > 0$

Problem Variation : Suppose f satisfies $deg\, f := d << n$ and g such that $deg\, g << n$

$$[x\, x\, \cdots\, x]\, \underbrace{[xx \cdots x]}_{f} \equiv 1 \mod x^n$$

Goal : more efficient method to compute g in this setting.

Example 6.7.1

Re-analyse algorithm from A2Q4 in terms of n and d.

Claim 6.7.1

Let $f \in F[x]$ with $deg\, f = d$, $f_0 \neq 0$, let h be such that $h \equiv f^{-1} \mod x^k$ for some $k \geq 0$, then $f^{-1} = h + f^{-1} r x^k$ (*) for some $r = r_0 + r_1 + \cdots \in F[x]$, $(1 + x + x^2 + \cdots = (1-x)^{-1})$

Proof

Solve for unknown r, multiply both sides of (*) by f, then

$$1 = fh + rx^k \Rightarrow \ r = \frac{1 - fh}{x^k}$$

Corollary 6.7.1

$deg\, r \leq deg\, f + deg\, h - k$. by assumption $h_0, \cdots h_{k-1} = g_0, \cdots, g_{k-1}$ where $g = f^{-1}$

Question : How to get g_k?

Answer : Consider (*) modulo x^{k+1}, $g_k = g_0 r_0$

Goal : Solve $Ax = b$ for $A \in \mathbb{Z}^{n \times n}$(nonsingular) and $b \in \mathbb{Z}^{n \times 1}$ in $\mathcal{O}(n^3(\log n + \log ||a||)^2)$ bit operations

Revisit Newton Iteration

Given : $f \in F[x]$, $deg\, f = d$, $f_0 \neq 0$

Compute : $g = g_0 + g_1 x + \cdots \in F[x]$ such that $gf \equiv 1 \mod x^n$ for some $n >> d$

Suppose $h = rem(f^{-1}, x^k)$ for some $k < n$

Then $f^{-1} = h + f^{-1}(\frac{1-fh}{x^k}x^k$ (*) $deg\, r < d$

Thus $f^{-1} \equiv h + rem(f^{-1}, x^l) r x^k \mod x^{k+l}$

- Newton Iteration has $l = k$, (*) becomes $f^{-1} = 2h - hfh \mod x^{2k}$
- A2Q4 has $l = 1$, (*) becomes $f^{-1} = h + g_0 \underbrace{r_0}_{\text{coefficient of } x^{k+1} \text{ of } fh} x^k \mod x^{k+1}$
- New idea : use $l = d$ and $rem(f^{-1}, x^d)$ at each iteration

Let $p \in \mathbb{Z}_+$, every element of $S := \{\frac{a}{b} : a, b \in \mathbb{Z}, b \perp p\}$ has a unique $p-$adic expansion $\frac{a}{b} = c_0 + c_1 p + \cdots$ such that $c_0 + c_1 p + c_{i-1} p^{i-1} \equiv \mod (\frac{a}{b}, p^i)$ for all i.

Example 6.7.2

$modp(17/21, 10) = 7, modp(17/21, 100) = 77, modp(17/21, 1000) = 277$

then $17/21 = 7 + 7 \times 10 + 4 \times 10^2 + 0 \times 10^3 + 9 \times 10^4 + 1 \times 10^5 + \cdots$

Note we can represent truncated expansion $u := modp(\frac{a}{b}, p^i)$ as $(c_0, \cdots, c_{i-1}) \in (\mathbb{Z}_p)^i$, i.e. i small integers (e.g $modp(17/21, 10^5) = (7, 7, 4, 0, 9)$) OR $c_0 + c_1 p + \cdots + c_{i-1} p^{i-1} \in \mathbb{Z}_{p^i}$, one large integer in range $[0, p^i - 1]$

Rational Number Reconstruction

Given an image $a \in \mathbb{Z}$, a modulus $M \in \mathbb{Z}_+$ and bounds $N, D \in \mathbb{Z}_+$ such that $2ND < M$

Find signed fraction $\frac{n}{d} \in \mathbb{Q}$ with $n \perp d$, $a \equiv \frac{n}{d} \mod M$, $|n| \leq N$ and $d \leq D$

Example 6.7.3

$17/21 \equiv 90477 \mod 10^5 \rightarrow \text{iteration}(90477, 10^2, 10^2, 10^5) \rightarrow 17/21$

Application to solve $Ax = b$

$Ax = b \Rightarrow \begin{bmatrix} 3 & 5 & 1 \\ 2 & 4 & 3 \\ 1 & 5 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$, choose $p = 5$ (need $p \perp \det A$)

Compute p-adic expansion of $A^{-1}b$:

$$\begin{aligned} A^{-1}b &\equiv \begin{bmatrix} 0 \\ 3 \\ 2 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \\ 2 \end{bmatrix} 5 + \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} 5^2 + \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} 5^3 + \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} 5^4 \mod 5^5 \\ &\equiv \begin{bmatrix} 175 \\ 173 \\ 1737 \end{bmatrix} \mod 5^5 (= 3125) \end{aligned}$$

Then in Maple : $map(iratrecon, \begin{bmatrix} 175 \\ 173 \\ 1737 \end{bmatrix}, 3125, 30, 30) = \begin{bmatrix} 25/18 \\ -11/18 \\ 8/9 \end{bmatrix}$

- How to compute c_0?

Compute $B := rem(A^{-1}, p) = \begin{bmatrix} 1 & 0 & 3 \\ 1 & 4 & 4 \\ 3 & 0 & 1 \end{bmatrix}$ (can find $\mathcal{O}(n^3)$ modulo p by first reducing A modulo p, can choose prime randomly, highly probability that $p \perp \det A$ out of some $\mathcal{O}(n)$ primes

Note $BA \equiv I_3 \mod 5$, then $c_0 = modp(Bb, 5)$

- How to compute c_1?

Ansatz: $A^{-1}b \equiv c_0 + c_1 p \mod p^2$, need to solve this for c_1

Multiply both sides by A, we get:

$$\begin{aligned} b &\equiv Ac_0 + Ac_1 p \mod p^2 \\ b - Ac_0 &\equiv Ac_1 p \mod p^2 \\ Ac_1 &\equiv \frac{b - Ac_0}{p} (:= r \in \mathbb{Z}^{n\times 1}) \mod p \\ Ac_1 &\equiv r \mod p \\ c_1 &\equiv A^{-1} r \mod p \\ c_1 &\equiv Br \mod p \end{aligned}$$

As before : $A^{-1}b = c_0 + A^{-1}rp$

6.8 Dixon's Algorithm

Choose a single small prime $p \in \mathbb{Z}$ such that $p \perp \det A$

Note $p \perp \det A \Rightarrow p^l \perp \det A$ for any l. We can choose p randomly. p can also be a prime power, e.g. $p = 2^{64}$

We will compute $c_0, c_1, \cdots \in \mathbb{Z}_p^{n\times 1}$ such that $A^{-1}b = c_0 + c_1 p + \cdots$

p-adic solver : $A \in \mathbb{Z}^{n\times n}, b \in \mathbb{Z}^{n\times 1}$, let $\alpha = \max(||A||^{\frac{n-1}{n}}, ||b||^{1/n})$

i.e. $\log \alpha = \mathcal{O}(\log ||A|| + \log ||b||/n)$

Pre-int : Let $M = \lfloor n^{n/2}\alpha^n \rfloor$, let $k \in \mathbb{Z}$ minimal s.t. $p^k > 2M^2$ ($k \in \Theta(n)$)

(1) Lint : $B \leq$ inverse of $modp(A, p)$ over $\mathbb{Z}_p$

(2) Lift : for $i = 0$ to $k-1$ do

//invariant : $A^{-1}b = c_0 + \cdots + c_{i-1}p^{i-1} + A^{-1}rp^i$

$c_0 = modp(B, modp(r, p), p)$ // compute over $\mathbb{Z}_p$

$r = \frac{r - Ac_i}{p}$ // compute over $\mathbb{Z}$ exact

(3) Represent conversion :

$u = c_0 + c_1 p + \cdots + c_{k-1}p^{k-1}$

return $map(iratrecon, u, p^k, M, M)$

Cost :(1)$\mathcal{O}(n^3(\log p)^2)$ word operations, compute $modp(A, p)$ then $modp(A, p)^{-1}$ over $\mathbb{Z}/(p)$

(2) $\mathcal{O}(kn^2(\log p)^2)$ word operations

(3) $\mathcal{O}(n(k\log p)^2)$ word operations

Overall : $\mathcal{O}(n^3(\log n + \log \alpha)^2)$ word operations, use $k \in \Theta(n)$ and $\log p \in \Theta(\log n + \log \alpha)$

Implementation Notes :

(1) can use radix $p = (p_1, \cdots, p_k) = (p_1 p_1 \cdots p_k)$ word-size primes, account for large entries of A

(2) algorithm combines Chinese Remaindering and p-adic lifting

(3) reduce "all" work to level 3 BLAS.

Chapter 7

The Resultant And A Modular GCD Algorithm in $\mathbb{Z}$x

7.1 GCDs over $\mathbb{Q}[x]$ and $\mathbb{Z}[x]$

Question : do gcds in $\mathbb{Z}[x]$ always exist? how to compute gcds over $\mathbb{Z}[x]$? what is the relationship between gcds over $\mathbb{Q}[x]$ and $\mathbb{Z}[x]$?

Definition 7.1.1 (gcd over R)
Let $a, b \in R$. then c is a gcd of a and b if

1) $c|a$ and $c|b$. Note $a = cq_1, b = cq_2$ for $q_1, q_2 \in R$

2) if $d|a$ and $d|b$, then $d|c$, for all $d \in R$

Example 7.1.1
$\gcd(15x + 30, 18x^2 + 6x - 60) = 3x + 6$ with $q_1 = 5, q_2 = 6x - 10$

gcd(6,15)=3

Example 7.1.2
Compute $gcd(f, g)$ for $f, g \in \mathbb{Z}[x]$

$f = -7x^3 + 22x^2 - 55x - 94, g = 89x^2 - 54x$

Definition 7.1.2 (UFD)
R is a UFD if every element in R has a unique factorization into irreducible (up to unit and ordering)

Definition 7.1.3 (Associates, lu, normal, contant, primitive root)
Over $\mathbb{Z}$:

lu(a)=sign(a), normal(a)=$|a|$, thus always have lu(a)=sign(a)×normal(a)

gcd(a,b) will return mean normalized gcd of a and b

Over $\mathbb{Z}[x]$:

Now let $f = f_n x^n + \cdots + f_0 \in \mathbb{Z}[x]$

Content of f is defined by $cont(f) := gcd(f_0, f_1, \cdots, f_n) \in \mathbb{Z}$, note $cont(f_0) = gcd(f_0) = normal(f_0)$

primitive part of f defined by $f := cont(f) \times pp(f)$, $cont(pp(f)) = 1$

Example 7.1.3
$f(x) = 18x^3 - 42x^2 + 30x - 6$, $g(x) = -12x^2 + 10x - 2$

$cont(f) = gcd(18, -42, 30, 6) = 6$, $cont(g) = gcd(-12, 10, -2) = 2$

$pp(f) = 3x^3 - 7x^2 + 5x + 1$, $pp(g) = -6x^2 + 5x - 1$ (not normalized)

Useful extension of cont and pp to $\mathbb{Q}[x]$

Let $f = (\frac{a_0}{b}) +)\frac{a_1}{b})x + \cdots + (\frac{a_n}{b})x^n \in \mathbb{Q}[x]$ with common denominator b

Example 7.1.4
$cont(-3x - 9/2) = 3/2$, $pp(f) = f/cont(f) \in \mathbb{Z}[x]$

Note since $R[x]$ is a UFD, any two elements have unique gcd (up to leading unit), we extend lu to $R[x]$ via $lu(lc(f))$.

$f \in R[x]$ is normalized when its leading coefficient is , e.g. over $\mathbb{Z}[x]$, lc should be positive.

$gcd(f, g)$ is the unique normalized gcd in $R[x]$

Corollary 7.1.1
Let $f, g \in \mathbb{Z}[x]$ with $h = gcd(f, g)$, then

(1) $cont(h) = gcd(cont(g), cont(g))$ (2) $pp(h) = gcd(pp(f), pp(g))$

(3) $h/lc(h) \in \mathbb{Q}[x]$ is the monic gcd of f and g in $\mathbb{Q}[x]$

Application to Computation : gcd over $\mathbb{Q}[x] \iff$ gcd over $\mathbb{Z}[x]$

Lemma 7.1.1
$pp(gcd_{\mathbb{Q}[x]}(f, g)) = gcd_{\mathbb{Z}[x]}(pp(f), pp(g))$

Input : primitive $f, g \in \mathbb{Z}[x]$

Output : $h = gcd(f, g) \in \mathbb{Z}[x]$

(1) Compute monic $v = gcd_{\mathbb{Q}[x]}(f, g)$ using EEA over $\mathbb{Q}[x]$

(2) $b = gcd(lc(f), lc(g))$

(3) Return $pp(bv) \in \mathbb{Z}[x]$ (numbers get bigger)

Example 7.1.5
$f = 12x^2 + 20x + 3$, $g = -30x^2 - 47x - 3$

(1) $gcd_{\mathbb{Q}[x]}(f, g) = x + 3/2$

(2) $gcd(lc(f), lc(g)) = 6$

(3) $6 \cdot (x + 3/2) = 6x + 9$, thus $pp(6x + 9) = 2x + 3$

Example 7.1.6
Compute $gcd(f,g), f,g \in \mathbb{Z}[x]$, $f = -7x^3 + 22x^2 - 55x - 94, g = 89x^2 - 54x$

First try a modular approach, let $p = 5$

$\bar{f} = modp(f,5) = 3x^3 + 2x^2 + 1 \in \mathbb{Z}/(5)[x]$, $\bar{g} = modp(g,5) = 4x^2 + x \in \mathbb{Z}/(5)[x]$, $gcd(\bar{f},\bar{g}) = 1 \in \mathbb{Z}/(5)[x]$

Suppose $h \in \mathbb{Z}[x]$ is a common divisor of f and g in $\mathbb{Z}[x]$, then $f = q_1 h$, $g = q_2 h$ for $q_1, q_2 \in \mathbb{Z}[x]$, then over $\mathbb{Z}/(5)[x]$ have

$$f \mod 5 \equiv (q_1 \mod 5)(h \mod 5), \quad g \mod 5 \equiv (q_2 \mod 5)(h \mod 5)$$

Then $(h \mod 5) \mid (f \mod 5)$ and $(h \mod 5) \mid (g \mod 5)$, thus $h \mod 5$ is a constant

Suppose $A, b, c \in \mathbb{Z}^{100\times 100}$, $||A||, ||B||, ||C|| \leq 99$ and $B \equiv A^{-1}C \mod 2\cdot 10^6$

Question : Does $B = A^{-1}C$?

Answer : Consider that $AB \equiv C \mod 2\cdot 10^6$

$$||AB|| \leq n \cdot 99^2 < 2\cdot 10^6/2, \quad ||C|| < 2\cdot 10^6/2$$

Recall for $f = f_0 + f_1 x + \cdots + f_n x^n \in \mathbb{Z}[x]$, $||f||_\infty = \max_i |f_i|$, $||f||_1 = \sum_0 |f_i|$

Remark
For any $g, h \in \mathbb{Z}[x]$, we have

$$||gh||_\infty \leq ||g||_1 ||h||_1$$

where gh is expensive to compute, but the RHS is cheaper to compute.

Remark
Suppose $f, g, h \in \mathbb{Z}[x]$ with $gh \equiv f \mod P$, if $||g||_1||h||_1 < P/2$ and $||F||_\infty < P/2$, then

$$gh = f \text{ (without the mod)}$$

Back to the GCD problem, we have the key notations for $f \in R$ and R a UFD

- $f = lu(f) \cdot normal(f)$
- gcd means normalized gcd

Theorem 7.1.1 (Gauss's Thm)
R a UFD, then $R[x]$ is a UFD

Let $f \in R[x]$, extend "lu" to $R[x]$ via lu(f)=lu(lc(f)), and also f =cont(f)$\cdot$ pp(f)

Example 7.1.7
$f = 10x^3 - 42x^2 + 30x - 6$, $g = -12x^2 + 10x - 2$, then

cont(f)=gcd(18,-42,30,-6)=6, pp$(f) = 3x^3 - 7x^2 + 5x - 1$

and cont(g)=2, pp$(g) = -6x^2 + 5x - 1$

gcd over $\mathbb{Q}[x]$: gcd(f, g)=normal$_{\mathbb{Q}[x]}\Big(\text{gcd}_{\mathbb{Z}[x]}(\text{pp}(f), \text{pp}(g)\ )\Big)$

gcd Over $\mathbb{Z}[x]$: gcd(f, g)=gcd(cont(f), cont(g))$\cdot$ gcd (pp(f), pp(g))

Theorem 7.1.2
$\text{gcd}_{\mathbb{Z}[x]}\big(\ \text{pp}(f), \text{pp}(g)\ \big) = \text{pp}\big(\ \text{gcd}_{\mathbb{Q}[x]}(f, g)\big)$

7.2 Modular Algorithm for GCD over $\mathbb{Z}[x]$

$f, g \in \mathbb{Z}[x] \xrightarrow{\text{gcd over } \mathbb{Z}[x]} h = gcd(f, g) \in \mathbb{Z}[x] \xrightarrow{\text{mod } p} \text{normal } (h \mod p) \in \mathbb{Z}_p[x]$

$f, g \in \mathbb{Z}[x] \xrightarrow{\text{mod } p} \bar{f}, \bar{g} \in \mathbb{Z}_p[x] \xrightarrow{\text{gcd over } \mathbb{Z}_p} gcd(\bar{f}, \bar{g}) \in \mathbb{Z}_p[x]$

Does the diagram commute?

Example 7.2.1
$f = 3x^3 + 3x - x^2 - 1 \in \mathbb{Z}[x]$, $g = 3x^2 + 5x - 2 \in \mathbb{Z}[x]$, $h = gcd(f, g) = 3x - 1 \in \mathbb{Z}[x]$

Try modular approach : $gcd(\ f \mod 7,\ g \mod 7\ ) = x + 2$

lc(h) mush divide $b :=$ gcd(lc(f), lc(g))=3

multiply image by b, reduce in symmetric range, take p^p : $3x + 6 \equiv 3x - 1 \mod 7$

But gcd($f \mod 5$, $g \mod 5$)=$x^2 + 1$ (degree too large , i.e. 5 is a bad prime)

gcd($f \mod 3$, $g \mod 3$)=1 (degree too small, i.e. 3 is a bad prime)

Main question : For what prime does

$$normal_{\mathbb{Z}_p[x]}\big(gcd_{\mathbb{Z}[x]}(f, g) \mod p\big) = gcd_{\mathbb{Z}_p[x]}\big(\ f \mod p,\ g \mod p\ \big)$$

7.3 The Resultant

Polynomial multiplication is a linear map

Example 7.3.1
$g = -2 + 5x + 3x^2 \in \mathbb{Q}[x]$, $t = t_0 + t_1 x + \cdots \in \mathbb{Q}[x]$

Infinite vector space $g * t$: Picture!

Let $f, g \in F[x]$ nonzero, $n = deg\, f$, $m = deg\, g$, then $(-g)f + (f)g = 0$

Lemma 7.3.1
$gcd(f, g) \neq 1$ if and only if nozero s, t such that $sf + tg = 0$ with $deg\, s <= g$ and $deg\, t < f$

Proof
$\Rightarrow$ $deg\, h = gcd(f,g) > 0$, then $(-g/h)f + (f/h)g = 0$

$\Leftarrow$ assume $sf + tg = 0$ and $f \perp g$, then $sf = -tg$ and $f|t$ imppossible if $det\, t < deg\, f$

View multiplication $\begin{bmatrix} f & g \end{bmatrix} \begin{bmatrix} s \\ t \end{bmatrix}$ as a linear map

$f = 3x^3 - x^2 + 3x - 1$, $deg\, f = n = 3$, $g = 3x^2 + 5x - 2$, $\;deg\, g = m = 2$

$$\underbrace{\begin{bmatrix} 3 & & 3 & & \\ -1 & 3 & 5 & 3 & \\ 3 & -1 & -2 & 5 & 3 \\ -1 & 3 & & -2 & 5 \\ & -1 & & & -2 \end{bmatrix}}_{Syl(f,g)} \begin{bmatrix} s_1 \\ s_0 \\ t_2 \\ t_1 \\ t_0 \end{bmatrix} = \text{the vector representation of } sf + tg$$

Theorem 7.3.1
Let $f, g \in F[x]$ be nonzero, then

(1) $\gcd(f,g) = 1$ iff $\mathrm{Syl}(f,g)$ is invertible

(2) if $\gcd(f,g) = 1$ and $n + m \geq 1$, then EEA computes v such that $\mathrm{Syl}(f,g)\, v = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$

Definition 7.3.1 (res(f,g))
$\mathrm{res}(f,g)$=det $\mathrm{Syl}(f,g)$

Corollary 7.3.1
Let $f, g \in F[x]$ be nonzero, TFAE

(1) $\gcd(f,g) = 1$

(2) $\mathrm{res}(f,g) \neq 0$ iff $\mathrm{Syl}(f,g)$ invertible

(3) do not exist nonzero $s, t \in F[x]$ such that $sf + tg = 0$, $deg\, s < deg\, g$ and $deg\, t < deg\, f$

Example 7.3.2
$f = 3x^3 - x^2 + 3x - 1$, $g = 3x^2 + 5x - 2$, $h = gcd(f,g) = 3x - 1$

$\mathrm{res}(f,g)$=0, $\mathrm{res}(f/h, g/h)$=res($x^2 + 1, x + 2$)=det Syl (f,g)=$\begin{vmatrix} 1 & 1 & \\ 0 & 2 & 1 \\ 1 & & 2 \end{vmatrix} = 5$

7.4 Key steps of Modular GCD Algorithm

Input : $f, g \in R[x]$

Output : $h = gcd(f,g)$

(1) Assume f, g are primitive

e.g. work with pp(f) and pp(g)

deal with content separately

(2) Let $b := gcd(lc(f), lc(g)) \in \mathbb{Z}$

only choose moduli/primes p with $p \nmid b$

then p is "bad" iff $deg\, gcd(f \mod p, g \mod p) > deg\, h$

then we can choose p one big prime, or $p = p_1 \cdots p_k$ for small primes p_i

(3) Compute $v := gcd(f \mod p,\ g \mod p) \in \mathbb{Z}/(p)[x]$;

monic modular gcd, hopefully $v \equiv (1/lc(h))h \mod p$

$w := mods(b \cdot v, p) \in \mathbb{Z}[x]$; hopefully $w = (b/lc(h))h$ over $\mathbb{Z}[x]$

(4) Check that $w \mid bf$ and $w \mid bg$

(5) Compute $f^*, g^* \in \mathbb{Z}[x]$ with max-norm $< p/2$ such that

$$f^* w \equiv bf \mod p \ ,\ g^* w \equiv bg \mod p \tag{7.1}$$

(6) Check that (6.1) holds without mod, if

$||f^*||_1, ||w||_1 < p/2$ and $||bf||_\infty < p/2$

$||g^*||_1, ||w||_1 < p/2$ and $||bg||_\infty < p/2$

then return $pp(w)$

Otherwise, if (6.1) does not hold without mod, then

p was bad, i.e. $p \mid res(f/h, g/h)$

And/Or p not bad enough

Example 7.4.1

$f = 6x^3 + 9x^2 + 10x + 15$, $g = 10x^4 + 15x^3 + 4x^2 + 8x + 3$, $h = gcd(6, 10) = 2$, $p = 227$

$v = gcd(f \mod p, g \mod p) = x - 112$

$w = mods(b * v, p) = 2x + 3$

$f^* = Quo(f, v, x) \mod p = 6x^2 + 10$

$g^* = Quo(g, v, x) \mod p = 10x^3 + 4x + 2$

$f^* w = \underbrace{(6x^2 + 10)}_{||\cdot||_1 = 6} \underbrace{(2x + 3)}_{||\cdot||_1 = 5} = 12x^3 + 18x^2 + 20x + 30 \mod 227$

$g^* w = \underbrace{(10x^3 + 4x + 2)}_{||\cdot||_1 = 16} \underbrace{(2x + 3)}_{||\cdot||_1 = 5} = 20x^4 + 30x^3 + 8x^2 + 16x + 6 \mod 227$

Return $pp(2x + 3) = 2x + 3$

Pick $p = 229$

$v = Gcd(f \mod p, g \mod p) \mod p = x^2 - 77x + 54$

$w = mods(b * v, p) = 2x^2 + 75x + 108$

$f^* = Quo(f, v, x) \mod p = 6x + 13$

$g^* = Quo(g, v, x) \mod p = 10x^2 + 98x - 89$

$f^* w = \underbrace{(6x + 13)}_{||\cdot||_1 = 19} \underbrace{(2x^2 + 75x + 108)}_{||\cdot||_1 = 185} = 12x^3 + 18x^2 + 20x + 30 \mod 229$

$g^* w = \underbrace{(10x^2 + 98x - 89)}_{||\cdot||_1 = 197} \underbrace{(2x^2 + 75x + 108)}_{||\cdot||_1 = 185} = 20x^4 + 30x^3 + 8x^2 + 16x + 6 \mod 229$

Note the LHS norm is larger, i.e. step (5) fail.

Main Question : how to choose p ?

Theorem 7.4.1
Let $f, g \in \mathbb{Z}[x]$ with $n = \max(deg\, f, deg\, g) \geq 1$ and $A = \max(||f||_\infty, ||g||_{\infty||})$, let $h = gcd(f, g) \in \mathbb{Z}[x]$, then

(1) $|res(f/h, g/h)| \leq (n + 1)^n A^{2n}$

(2) $||f/h||_1 \cdot ||\frac{b}{lc(h)} h||_1 \leq (n + 1)^{1/2} 2^n Ab$

(3) $||g/h||_1 \cdot ||\frac{b}{lc(h)} h||_1 \leq (n + 1)^{1/2} 2^n Ab$

Option 1 : Classical Approach

Let $B = b(n + 1)^{1/2} 2^n A$, choose p randomly in range $2B < p < 4B$

7.5 From Integer to Polynomials

Recall Gauss's Thm, R a UFD leads to $R[x]$ a UFD

$R \to R[x] \to R[x, y] \to \cdots$

Example 7.5.1
$R = \mathbb{Z}_7[y]$, consider polynomials from $R[x]$

$f = \underbrace{(6y^4 + 2y^3 + 3y^2)}_{f_2} x^2 + \underbrace{(3y^4 + 6y^2 + 2y)}_{f_1} x + \underbrace{(2y^3 + 2y^2)}_{f_0}$

$g = \underbrace{(3y^3 + 6y^2 + 3y)}_{g_3} x^3 + \underbrace{(2y^2 + 4y + 2)}_{g_2} x^2 + \underbrace{(2y^3 + 5y^2 + 3y)}_{g_1} x + \underbrace{(6y^2 + y + 2)}_{g_0}$

$cont(f) = gcd(f_2, f_1, f_0) = y^2 + y$, $cont(g) = gcd(g_3, g_2, g_1, g_0) = y + 1$

$pp(f) = (y^2 + 4y)x^2 + (4y^2 + 3y + 5)x + 5y$, $pp(g) = (y^2 + y)x^3 + (3y + 3)x^2 + (3y^2 + y)x + 2y + 3$

gcd computation : $gcd_{R[x]}(f, g) = \underbrace{gcd(cont(f), cont(g))}_{(y+1)} \cdot \underbrace{gcd(pp(f), pp(g))}_{(yx+3)}$

Now consider $f, g \in R[y]$ with $R = \mathbb{Z}_7[x]$, then

$f = \underbrace{(6x^2+3x)}_{f_4} y^4 + \underbrace{(2x^2+2)}_{f_3} y^3 + \underbrace{(3x^2+6x+2)}_{f_2} y^2 + \underbrace{(2x)}_{f_1} y$

$g = (3x^3+2x)y^3 + (6x^3+2x^2+5x+6)y^2 + (3x^3+4x^2+3x+1)y + 2x^2 + 2$

$cont(f) = 1$, $cont(g) = 1$, $pp(f) = f$, $pp(g) = g$

gcd computation : $gcd_{R[y]}(f,g) = \underbrace{gcd_R(cont(f), cont(g))}_{1} \cdot \underbrace{gcd_{R[y]}(pp(f), pp(g))}_{xy^2+(x+3)y+3=(y+1)(yx+3)}$

7.6 Modular Algorithm For GCD Over $F[x,y]$

Input : Primitive $f, g \in F[x,y] = R[x]$ where $R = F[y]$ and F a field.

$deg_x f = n \geq deg_x g = m$ and $deg_y f, deg_y g \leq d$

(1) No need for number theoretic bound

e.g. $deg_y h \leq d$, $deg\, f/h = d - deg_y h$

(2) Easy to get bound on $deg_y res_x(f/h, g/h)$

$deg_y res_x(f/h, g/h) \leq (n+m)d$

Example 7.6.1

$f/h = (6y^2+3y)x + 3y^2$, $g/h = (3y+3)x^2 + 2y + 3 \in \mathbb{Z}_7[y][x]$

$$\begin{aligned} res_x(f/h, g/h, x) &= det\, Syl_x(f/h, g/h) \\ &= \begin{bmatrix} 6y^2+3y & & 3y+3 \\ 3y^2 & 6y^2+3y & 0 \\ & 3y^2 & 2y+3 \end{bmatrix} = y^2(y^2+2y+3)(y+2) \end{aligned}$$

Algorithm :

(1) Let $b = gcd\big(lc + x(f), lc_x(g)\big) \in R = F[y]$, let $L = d + 1 + deg_y b$, where $L > deg_y\, b\frac{h}{lc(h)}h$

(2) Compute $v_i = b(\alpha_i) \cdot gcd\big(f(x,\alpha_i), g(x,\alpha_i)\big)$ over $F[x]$ for $L, \alpha_i \in F$ such that $y - \alpha_i \nmid b$

i.e. detect (some) bad images and discard.

(3) Once L consistent images computed, interpolate to get $v \in R[x]$ such that

$$v \equiv v_i \mod (y - \alpha_i) \text{ for } 1 \leq i \leq L$$

(4) Check such that $v \mid bf$ and $v \mid bg$