Algoritmy počítačovej algebry II

Jaroslav Guričan

16. mája 2012

1 Úvod

Definícia $O(f)$ (zložitosť)

Násobenie prirodzených čísiel (algoritmus ceruzka, papier - $O(n^2)$, Karacuba - $O(n^{log_2 3})$, FFT - $O(nlog_2 n)$ - n je počet cifier čísiel, väčšinou sa pracuje s prípadom, keď majú obe čísla približne rovnakú veľkosť)

Násobenie a delenie polynómov (delenie so zvyškom): v oboch prípadoch potrebujeme asi n^2 operácií v poli F (resp. v okruhu R, samozrejme, delenie nad okruhom môžeme robiť vo všeobecnosti len vtedy, keď vieme deliť "vedúcim" koeficientom deliteľa.) n je väčší zo stupňov polynómov vstupujúcich do algoritmu. Presnejšie, na vynásobenie polynómov stupňa m a n nad F potrebujeme $O((m+1)(n+1))$ operácií nad F a ich vydelenie so zvyškom $O((m-n+1)(n+1))$ (predpokladáme, že $m \geq n$, inak je výsledok delenia prvý polynóm). Tie "+1" v uvedených výrazoch sú kvôli polynómom stupňa 0.

Euklidov algoritmus na číslach je o niečo lepší:

Veta 1 *Nech a, b sú dve prirodzené čísla. Položme $n = \max\{a, b\}$. Potom štandardný euklidov algoritmus na výpočet* $\mathrm{nsd}\,(a, b)$ *potrebuje najviac $log_2 n$ krokov.*

Dôkaz. Nech $n = a \geq b$, položme $a_0 = a, a_1 = b$ a postupnosť delení so zvyškom z euklidovho algoritmu:

$$\begin{aligned} a_0 &= q_0 a_1 + a_2 \\ a_1 &= q_1 a_2 + a_3 \\ &\dots \\ a_k &= q_k a_{k+1} \quad \text{(posledný zvyšok je 0)} \end{aligned}$$

Keďže všetky čísla v uvedených iteráciách sú celé čísla, platí, že $a_0 \geq a_1 \geq \cdots \geq a_{k+1}$, $q_0 \geq 1, \dots, q_{k-1} \geq 1$ a $q_k \geq 2$.

Preto

$$\begin{aligned} a_0 &\geq a_1 + a_2 \geq 2a_2 \\ a_1 &\geq a_2 + a_3 \geq 2a_3 \\ &\dots \\ a_{k-1} &\geq a_k + a_{k+1} \geq 2a_{k+1} \end{aligned}$$

Vynásobením týchto nerovností dostaneme

$$a_0 a_1 \dots a_{k-1} \geq 2^k a_2 a_3 \dots a_{k+1}$$

t.j. $a_0 a_1 \geq 2^k a_{k+1} \geq 2^k$. Keďže $n = a_0 \geq a_1$, dostávame $n^2 \geq 2^k$, po zlogaritmovaní $2 \log_2 n \geq k$, t.j. k je $O(\log n)$. □

Zaujímavý je binárny euklidov algoritmus, založený na zápise čísiel do binárnej sústavy. Uvedieme si príklad: Nech $a = 728_{10} = 1011011000_2$, $b = 220_{10} = 11011100_2$. Urobíme si tabuľku výpočtu $\mathrm{nsd}\,(a, b)$, v tabuľke budeme všetky čísla písať v dvojkovej sústave.

a	b	k pre 2^k	poznámka
1011011000	11011100		obe sú deliteľné 2^2, lebo končia 00
10110110	110111	2	vykrátili sme 2^2, lebo končia 00
1011011	110111		prvé sme vydelili 2, lebo končí 0 a druhé je s 2 nesúdeliteľné
110111	100100		druhé do 1. stĺpca, rozdiel prvé-druhé do druhého stĺpca
110111	1001		druhé sme vydelili 4, lebo končí 00 a druhé je s 2 nesúdeliteľné
1001	101110		druhé do 1. stĺpca, rozdiel prvé-druhé do druhého stĺpca
10111	1001		druhé sme vydelili 2 a vymenili sme stĺpce
1001	1110		druhé do 1. stĺpca, rozdiel prvé-druhé do druhého stĺpca
1001	111		druhé sme vydelili 2
111	10		druhé do 1. stĺpca, rozdiel prvé-druhé do druhého stĺpca
111	1		druhé sme vydelili 2

Tabuľka 1:

Keďže máme v 2. stĺpci číslo 1, nepárny spoločný deliteľ je 1, t.j. celkový nsd je $4 = 2^2$ z tretieho stĺpca.

Euklidov algoritmus na polynómoch.
Často sa môže hodiť euklidov algoritmus pre polynómy - je to úloha, ktorá sa môže vyskytovať napríklad pri zjednodušovaniach výrazov. V tomto prípade budeme predpokladať, že operácie v poli vieme urobiť rýchlo, t.j. v konštantnom čase (jeden strojový takt).

Veta 2 *Nech a, b sú dva polynómy nad poľom $F[x]$. Položme $n = \max\{\mathrm{st}\,(a), \mathrm{st}\,(b)\}$. Potom štandardný euklidov algoritmus na výpočet* $\mathrm{nsd}\,(a, b)$ *potrebuje najviac n^2 operácií v poli F.*

Dôkaz. Nech $n = \mathrm{st}\,(a) \geq \mathrm{st}\,(b)$, položme $a_0 = a, a_1 = b$ a postupnosť delení so zvyškom z euklidovho algoritmu:

$$\begin{aligned} a_0 &= q_0 a_1 + a_2 \\ a_1 &= q_1 a_2 + a_3 \\ &\dots \\ a_k &= q_k a_{k+1} \quad \text{(posledný zvyšok je 0)} \end{aligned}$$

Poďme sa pozrieť na jednotlivé iterácie. Na delenie

$$a_i = a_{i+1} q_{i+1} + a_{i+2}$$

potrebujeme čas (počet operácií nad poľom F) rovnaký ako na násobenie $a_{i+1} q_{i+1}$, t.j.

$$T_{i+1} \leq c \cdot \mathrm{st}\,(a_{i+1}) \mathrm{st}\,(q_{i+1})$$

pre vhodnú konštantu c. Celkový čas teda je

$$T = T_1 + \cdots + T_n \leq c \cdot \sum_{i=1}^{n} \mathrm{st}(a_i)\mathrm{st}(q_i) \leq c \cdot n \sum_{i=1}^{n} \mathrm{st}(q_i)$$

ale

$$\sum_{i=1}^{n} \mathrm{st}(q_i) = \sum_{i=1}^{n} \mathrm{st}(a_{i-1}) - \mathrm{st}(a_i) = \mathrm{st}(a_0) - \mathrm{st}(a_n) \leq \mathrm{st}(a_0) \leq n$$

Spolu teda vidíme, že čas (počet operácií nad poľom F) je ohraničený číslom $c \cdot n^2$, t.j. $T \leq c \cdot n^2$ $\square$

Často je potrebné nájsť nsd dvoch prvkov ako lineárnu kombináciu týchto dvoch prvkov. Na tento účel sa dá použiť tzv. rozšírený euklidov algoritmus, ktorý ide podľa štandardného euklidovho algoritmu, len sa stále pokúša udržať informáciu o tom, aká lineárna kombinácia pôvodných prvkov je aktuálny zvyšok.

Platí $a_0 = 1.a_0 + 0.a_1$, $a_1 = 0.a_0 + 1.a_1$. V iterácii, nech $a_{i-1} = x_{i-1}a_0 + y_{i-1}a_1$, $a_i = x_i a_0 + y_i a_1$, vypočítame $a_{i-1} = q_i a_i + a_{i+1}$, t.j.

$$\begin{aligned} a_{i+1} &= a_{i-1} - q_i a_i = x_{i-1}a_0 + y_{i-1}a_1 - q_i(x_i a_0 + y_i a_1) = \\ &= (x_{i-1} - q_i \cdot x_i)a_0 + (y_{i-1} - q_i \cdot y_i)a_1 \end{aligned}$$

Keďže posledný krok v euklidovom algoritme už má tvar $a_{n-1} = q_n a_n$ ($a_{n+1} = 0$), nsd je vlastne číslo a_n, t.j. rozšírený euklidov algoritmus pracuje takto:

1 Nastav $x_0 = 1$, $y_0 = 0$, $x_1 = 0$, $y_1 = 1$

2 Pri iterácii vypočítaj $x_{i+1} = x_{i-1} - q_i \cdot x_i$, $y_{i+1} = y_{i-1} - q_i \cdot y_i$, kde q_i je číslo z rovnice $a_{i-1} = q_i a_i + a_{i+1}$.

Tento krok podľa popisu zabezpečuje to, že $a_{i+1} = x_{i+1}a_0 + y_{i+1}a_1$

1. opakuj tento krok, kým nedostaneš $a_{n+1} = 0$, vtedy je a_n nsd a $a_n = x_n a_0 + y_n a_1$.

Pseudokód pre tento algoritmus je nasledujúci:

```
function extended_gcd(a, b)
    x := 0    lastx := 1   // x je x_1, lastx je x_0
    y := 1    lasty := 0   // y je y_1, lasty je y_0
    while b <> 0
        temp := b
        quotient := a div b   // quotient je q_i
        b := a mod b
        a := temp

        temp := x
        x := lastx-quotient*x   // lastx je x_{i-1},
                                // x vpravo je x_i, x vľavo je x_{i+1}
        lastx := temp

        temp := y
```

```
        y := lasty-quotient*y   // lasty je y_{i-1},
                                // y vpravo je y_i, y vľavo je y_{i+1}
        lasty := temp
    return {lastx, lasty, a}    // a je nsd, lastx, lasty
                                // sú príslušné koeficienty
end
```

Tento algoritmus je potrebný o.i. pre výpočet inverzného prvku nad poľom typu Z_p, prípadne v poli typu $F[x]/(f)$ pre ireducibilný polynóm $f \in F[x]$.

Čínska veta o zvyškoch pre euklidovské okruhy (CRT, chinesse remainder theorem).

Veta 3 *Nech $(E, +, \cdot)$ je euklidovský okruh s ohodnotením g, $m_1, \ldots, m_n$ sú nenulové nesúdeliteľné prvky z E, $a_1, \ldots, a_n$ sú ľubovoľné prvky z E. Potom systém kongruencií*

$$\begin{aligned} x &\equiv a_1 \mod m_1 \\ x &\equiv a_2 \mod m_2 \\ &\cdots \\ x &\equiv a_n \mod m_n \end{aligned}$$

má riešenie.

Ak je $(E, +, \cdot) = (Z, +, \cdot)$, tak existuje práve jedno také riešenie x, že

$$0 \leq x < m_1 \cdot m_2 \cdot \ldots \cdot m_n$$

.

Ak je $(E, +, \cdot) = (F[x], +, \cdot)$, tak existuje práve jedno také riešenie $f(x)$, že

$$\mathrm{st}(f) < \mathrm{st}(m_1) + \mathrm{st}(m_2) + \cdots + \mathrm{st}(m_n)$$

.

Dôkaz. Riešenie x budeme hľadať postupnými iteráciami. Urobme delenie so zvyškom $a_1 = q_1 a_1 + r_1$ (zvyšok r_1 je buď 0 alebo má ohodnotenie menšie ako m_1, t.j. $g(r_1) < g(m_1)$). Položíme $x_1 = r_1$. Takto sme získali riešenie prvej kongruencie.

Upravme x_1 tak, aby sme zachovali to, že máme riešenie prvej kongruencie, ale aby sme získali riešenie aj druhej kongruencie. To sa dá urobiť tak, že nájdeme vhodné $c \in E$, aby $x_2 = x_1 + c \cdot m_1$ bolo riešením druhej kongruencie (x_2 tohoto tvaru je riešením prvej kongruencie pre úplne ľubovoľné c).

Keďže m_1 a m_2 sú nesúdeliteľné, existujú $s, t \in E$ také, že $1 = m_1 s + m_2 t$ a teda $m_1 s \equiv 1 \mod m_2$ (s, t sa dajú získať pomocou rozšíreného euklidovho algoritmu). Podmienku $x_1 + c \cdot m_1 \equiv a_2 \mod m_2$ zabezpečíme tak, že urobíme $c \cdot m_1 \equiv a_2 - x_1 \mod m_2$, potom prenásobením prvkom s získame $c \cdot m_1 s \equiv (a_2 - x_1)s \mod m_2$ a teda $c \equiv (a_2 - x_1)s \mod m_2$. Za c teda stačí zobrať zvyšok $(a_2 - x_1)s$ pri delení číslom m_2.

Predpokladajme teraz, že už máme riešenie x_j prvých j kongruencií ($2 \leq j < n$). Urobme $x_{j+1} = x_j + c \cdot m_1 m_2 \ldots m_j$. Podobne ako v predošlom prípade je x_{j+1} riešením prvých j kongruencií pre ľubovoľné $c \in E$. Znovu, m_{j+1}

je nesúdeliteľné so súčinom $m_1 \dots m_j$ a teda existujú také $s, t \in E$, že $1 = (m_1 \dots m_j)s + m_{j+1}t$, t.j. $(m_1 \dots m_j)s \equiv 1 \mod m_{j+1}$ a potom môžeme počítať podobne ako v predošlom prípade:

$x_{j+1} \equiv a_{j+1} \mod m_{j+1}$ prepíšeme na $x_j + c \cdot m_1 m_2 \dots m_j \equiv a_{j+1} \mod m_{j+1}$, odpočítaním získame

$c \cdot m_1 m_2 \dots m_j \equiv a_{j+1} - x_j \mod m_{j+1}$, prenásobením prvkom s získame

$c \cdot m_1 m_2 \dots m_j s \equiv (a_{j+1} - x_j)s \mod m_{j+1}$ a teda $c \equiv (a_{j+1} - x_j)s \mod m_{j+1}$. Čiže opäť stačí za c zobrať zvyšok $(a_{j+1} - x_j)s$ pri delení prvkom m_{j+1}.

Fakt, že riešenie vieme nájsť aj s obmedzením na veľkosť (pri celých číslach), prípadne stupeň (pri polynómoch) sa ľahko dokáže indukciou. Jednoznačnosť riešenia pri danom obmedzení na veľkosť (stupeň) je tiež ľahko vidieť. □

Špeciálny prípad, keď je $E = F[x]$ - okruh polynómov nad poľom F: zoberme $m_i = x - b_1$, $b_i \in F$ pre $i = 1, \dots, n$. V tomto prípade sú m_i, m_j nesúdeliteľné práve vtedy, keď $b_i \neq b_j$. Ak je $a_i \in F$, kongruencia $f(x) \equiv a_i \mod (x - b_i)$ je ekvivalentná s faktom, že $f(b_i) = a_i$ a teda pre tento prípad CRT je vlastne veta o interpolácii - ak máme po dvoch rôzne prvky $b_1, \dots, b_n \in F$ a ľubovoľné $a_1, \dots, a_n \in F$, potom existuje práve jeden polynóm $f(x) \in F[x]$ stupňa menej ako n taký, že pre všetky $i = 1, \dots, n$ platí $f(b_i) = a_i$. Podľa dôkazu má polynóm $f(x)$ tvar $f(x) = a_1 + c_1(x - b_1) + c_2(x - b_1)(x - b_2) + \dots + c_{n-1}(x - b_1)(x - b_2) \dots (x - b_{n-1})$, kde $c_1, \dots, c_{n-1}$ sú prvky z F - lebo sú to zvyšky pri delení polynómami $(x - b_2), \dots, (x - b_n)$. Uvedený tvar (interpolačného) polynómu $f(x)$ sa nazýva Newtonov interpolačný polynóm.

Tu je pseudokód pre všeobecné nájdenie prvku x z predošlej vety.

```
function crt_interpolation(a, m) // a je pole a_1,...,a_n
                                 // m je pole m_1,...m_n
                                 // nesúdeliteľných prvkov
1.  x := a_1 mod m_1   // čiže zvyšok pri delení prvkom m_1
    M := 1
    n:= length (a)  // dĺžka poľa a a/alebo m
    for k:= 2 to n
2.      M := M*m_{k-1}
3.      s := M^{-1} mod m_k      // s z dôkazu, vypočítať sa dá
                                 // napr. pomocou extend_gcd
4.      c := (a_k-x)*s mod m_k   // c z dôkazu
5.      x := x + c*M             // nová iterácia
    end_for
    return x   //
end
```

V prípade použitia CRT na interpoláciu polynómov nad poľom F, t.j. keď $a_i \in F$ a $m_i = x - b_i$, kde $b_i \in F$ sú po dvoch rôzne prvky je dobre si uvedomiť, ako fungujú niektoré operácie, najmä

Lema 4 *Nech $f(x) \in F[x]$, $a \in F$. Potom $f(x) \mod (x - a) = f(a)$.*

a

Lema 5 *Nech $f(x) \in F[x]$, $a \in F$, pričom a nie je koreň $f(x)$, t.j. $f(a) \neq 0$. Potom $f(x)^{-1} \mod (x - a) = f(a)^{-1}$. ($f(a)$ je prvok poľa F a preto existuje jeho inverzný prvok).*

Použitím týchto liem vieme pre tento prípad prepísať predošlý algoritmus do tvaru

```
function newton_interpolation(a, b) // a je pole a_1,...,a_n, b je pole
                                    // b_1,...b_n navz. rôznych
                                    //  prvkov, čiže ako m_i budeme
                                    // brať polynómy x - b_i
1.  f(x) := a_1                     // toto je a_1 mod (x-b_1)
    M(x) := 1
    n:= length (a)  // dĺžka poľa a a/alebo m
    for k:= 2 to n
2.      M(x) := M(x)*(x-b_{k-1})
3.      s := M(a_k)^{-1}     // podľa lemy je to M(x) mod (x-b_k)
4.      c := (a_k-f(a_k))*s    // podľa lemy je to (a_k-f(x))*s mod (x-b_k)
5.      f(x) := f(x) + c*M(x)             // nová iterácia
    end_for
    return f(x)   //
end
```

Urobme si analýzu, koľko operáciií v poli F musíme urobiť, aby sme vypočítali interpolačný polynóm podľa tohoto algoritmu.

V riadku 1 nerobíme žiadne operácie. Cyklus for ... end_for robíme $n-1$ krát pre hodnoty $2, \ldots, n$.

Pri vstupe do cyklu pre dané k má polynóm $M(x)$ stupeň $k-2$, v riadku 2 ho násobíme polynómom $(x - b_{k-1})$, na čo potrebuje $k-1$ násobení (násobenie prvkom 1 nepočítame ako operácie) a $k-1$ sčítaní, čiže $O(k)$ operácií. Po vynásobení má $M(x)$ stupeň $k-1$.

V riadku 3 teda dosadzujeme prvok a_k do polynómu stupňa $k-1$, pri použití hornerovej schémy potrebujeme $2k$ operácií, vypočítame jeden inverzný prvok v F, t.j. spolu tiež $O(k)$ operácií.

V riadku 4 pracujeme s polynómom $f(x)$, ktorý má v tomto mieste stupeň $k-2$, do polynómu $f(x)$ dosadzujeme prvok a_k, t.j. $O(k)$ operácií, urobíme ešte $a_k - f(a_k)$ a potom to vynásobíme prvkom s, čiže pouzijeme ešte 2 operácie, to nič nezmení na tom, že na riadok 4 potrebujeme $O(k)$ operácií.

V riadku 5 robíme vynásobenie polynómu $M(x)$ stupňa $k-1$ prvkom c a pričítanie polynómu $f(x)$, ktorý je stupňa $k-2$, t.j. spolu $O(k)$ operácií. Po skončení riadku 5 je v danej iterácii cyklu stupeň polynómu $f(x)$ $k-1$.

Teda na vykonanie jednej iterácie cyklu for ... end_for potrebujeme $O(k)$ operácií, nech je ohraničenie $C \geq 0$, t.j. pre ľubovoľné k, $k = 2, \ldots, n$ potrebujeme operácií $\leq Ck$. Potom počet operácií potrebných na vykonanie celého algoritmu je $\leq C(2 + 3 + \cdots + n) < C\frac{n(n+1)}{2}$, čiže $O(n^2)$.

2 Schémy počítania pomocou CRT

Varianty počítania pomocou CRT - schémy SHI, MHI (Single, Multiply Homomorphic Image), pre polynómy S(M)HI cez koeficienty, dosadenie-výpočet-interpolácia Kladné koeficienty (nesymetrické reprezentácie), ľubovoľné celočíselné koeficienty (symetrické reprezentácie). Vo všetkých prípadoch sa budeme snažiť vypočítať hodnotu výrazu $\mathbf{e}(v_1, \ldots, v_n)$, čo bude okruhový výraz (výraz

- expression, preto sme použili písmeno **e**) v hodnotách (vstupoch) $v_1, \ldots, v_n$, ktoré sú z daného okruhu.

2.1 MHI schéma pre $F[x]$, špeciálny prípad je $Z_p[x]$

V tejto časti je $F[x]$ okruh polynómov. Túto schému budeme nazývať aj evaluačno-interpolačná.

Budeme potrebovať odhad na stupeň výsledku pre dané vstupy, t.j. $k < st(\mathbf{e}(v_1, \ldots, v_n))$. Ako tento odhad získať - to závisí od konkrétnej úlohy, kde sa táto metóda použije.

1. krok: Zvoľme k navzájom rôznych prvkov $a_1, \ldots, a_k$ poľa F . (Ak pole F nemá k prvkov, máme smolu.) Podľa predpokladu sú $v_1, \ldots, v_n$ polynómy v premennej x a preto pre každé relevantné i, l je $v_i(x) \equiv v_i(a_l) \mod x - a_l$. Aj **e** je okruhový výraz nad $F[x_1, \ldots, x_n]$ a preto pre každé relevantné l je $\mathbf{e}(v_1(x), \ldots, v_n(x)) \equiv \mathbf{e}(v_1(a_l), \ldots, v_n(a_l)) \mod x - a_l$.

Preto pre $i = 1, \ldots, k$ vypočítame hodnotu $d_i = \mathbf{e}(v_1(a_i), \ldots, v_n(a_i)$ (evaluácia).

2. krok: Nájdeme polynóm $f(x) \in F[x]$ stupňa najviac k taký, že pre všetky $i = 1, \ldots, k$ platí $f(a_i) = d_i$ (tu sa použije CRT pre polynómy - Newtonova interpolačná metóda).

2.2 MHI schéma pre Z s celočíselnou CRT

Potrebujeme vypočítať hodnotu výrazu $\mathbf{e}(x_1, \ldots, x_n) \in Z[x_1, \ldots, x_n]$ vo vstupných hodnotách $a_1, \ldots, a_n \in Z$.

Odhadneme horné ohraničenie výsledku, t.j. nájdeme číslo B také, aby $|\mathbf{e}(a_1, \ldots, a_n)| < B$ (toto je opäť ad-hoc záležitosť, ktorá závisí od výrazu **e** a od vstupných hodnôt).

1. krok: Teraz máme 2 možnosti. Ak vieme, že výsledná hodnota $\mathbf{e}(x_1, \ldots, x_n)$ bude nezáporná, zvolíme istý počet navzájom nesúdeliteľných čísiel $m_1, \ldots, m_k$ tak, aby $B \leq m_1. \cdots .m_k$.

Ak je možné, že výsledok, t.j. $\mathbf{e}(x_1, \ldots, x_n)$ môže byť aj kladný aj záporný a vopred nevieme, ktorá z týchto možností nastane, voľbu musíme urobiť tak, aby $2B \leq m_1. \cdots .m_k$.

Pre $i = 1, \ldots, k$ vypočítame hodnoty $e_i = \mathbf{e}(a_1 \mod m_i, \ldots, a_n \mod m_i)$ mod m_i - t.j. každé $e_i \in Z_{m_i}$.

2. krok: Pomocou algoritmu z dôkazu CRT nájdeme (jediné) $a \in Z$, ktoré spĺňa všetky kongruencie $a \equiv e_i \mod m_i$ (pre $i = 1, \ldots, k$) a tiež podmienku $0 \leq a < m_1. \cdots .m_k$. Ak vieme, že výsledok je nezáporný, je a hľadaný výsledok, t.j. $a = \mathbf{e}(a_1, \ldots, a_n)$. Ak vopred nepoznáme znamienko výsledku, t.j. naša voľba $m_1, \ldots, m_k$ bola taká, aby $2B \leq m_1. \cdots .m_k$ a dostali sme hodnotu $a \geq B$, správny výsledok je záporný a preto $\mathbf{e}(a_1, \ldots, a_n) = -(m_1. \cdots .m_k - a)$.

Príklad: Nájdite riešenie systému rovníc

$$\begin{aligned} 9x_1 + 3x_2 - 3x_3 &= 1 \\ -x_1 + 4x_2 + 5x_3 &= 3 \\ 3x_1 - x_2 - 7x_3 &= 6 \end{aligned}$$

Táto úloha v skutočnosti je úloha na výpočet v $Q[x]$, lebo riešenia linerárneho systému s celočíselnými koeficientami vo všeobecnosti budú racionálne čísla,

$x_1 = \frac{a_1}{b_1}$, $x_2 = \frac{a_2}{b_2}$, $x_3 = \frac{a_3}{b_3}$, pričom $a_1, \ldots, a_3 \in Z$, $b_1, \ldots, b_3 \in N_0$. To ale znamená, že ju môžeme chápať ako úlohu nad Z. Ale keď budeme hľadať tieto riešenia (napr. gaussovo eliminačnou metódou) nad poliami Z_{p_i} (vieme, že systém rovníc môžeme riešiť nad poľom), ako riešenia dostaneme prvky zo Z_{p_i}, čo nie sú primárne zlomky, respektíve, každý prvok v Z_{p_i} vieme interpretovať ako zlomok, pre každý menovateľ vieme nájsť práve jeden vhodný čitateľ, aby tento zlomok ako výraz v poli Z_{p_i} interpretoval naše číslo. Preto je dôležité povedzme nájsť nielen výsledky, ale aj správne menovatele.

Pri tom nám pomôže Crammerovo pravidlo: riešenie $(x_1, \ldots, x_3)$ systému rovníc môžeme nájsť pomocou známych vzorcov $x_i = \frac{\det(A_i)}{\det(A)}$, kde A je matica systému a A_i je matica, ktorá vznikne z A nahradením $i-$teho stĺpca stĺpcom pravých strán systému.

Ak je teda $d = \det(A) \neq 0$ a $(x_1, \ldots, x_3)$ je riešenie nášho systému nad nejakých poľom Z_p (ktoré získame napr. elimináciou), položíme $y_i = dx_i$ a teda máme riešenie v tvare zlomkov $x_i = \frac{y_i}{d}$. (V skutočnosti sme teda úlohu zmenili na úlohu nájsť $\det(A)$, $\det(A_1) = y_1$, $\det(A_2) = y_2$, $\det(A_3) = y_3$, čo sú úlohy nad Z.) Odhadneme $B = 590$ (viď nižšie), výsledky $a_1, \ldots, a_3$, $b_1, \ldots, b_3$ ($b_1 = b_2 = b_3 = \det(A)$) môžu byť aj kladné aj záporné, nájdeme (málo) prvočísiel tak, aby $2.590 \leq p_1 . \cdots . p_k$, napr. $7.11.17 = 1309 > 2.590$

Zoberme teraz rozšírené matice našej sústavy nad 7, 11 a 17:

$$A'_7 = \left(\begin{array}{ccc|c} 2 & 3 & 4 & 1 \\ 6 & 4 & 5 & 3 \\ 3 & 6 & 0 & 6 \end{array}\right) \quad A'_{11} = \left(\begin{array}{ccc|c} 9 & 3 & 8 & 1 \\ 10 & 4 & 5 & 3 \\ 3 & 10 & 4 & 6 \end{array}\right) \quad A'_{17} = \left(\begin{array}{ccc|c} 9 & 3 & 14 & 1 \\ 16 & 4 & 5 & 3 \\ 3 & 16 & 7 & 6 \end{array}\right)$$

Po eliminácii (na trojuholníkový redukovaný tvar) dostaneme, vždy nad "správnym poľom":

$$B'_7 = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right) \quad B'_{11} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 6 \end{array}\right) \quad B'_{17} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 8 \\ 0 & 1 & 0 & 12 \\ 0 & 0 & 1 & 13 \end{array}\right)$$

a $\det(A_7) = 4$, $\det(A_{11}) = 4$ a $\det(A_{17}) = 3$.

Čiže nad Z_7 je riešenie $(2, 0, 1)$ a preto $x_1 = \frac{2.4}{4} = \frac{1}{4}$, $x_2 = \frac{0.4}{4} = \frac{0}{4}$ a$x_3 = \frac{1.4}{4} = \frac{4}{4}$, t.j. v tomto prípade máme $(\det(A_1) \mod 7, \det(A_2) \mod 7, \det(A_3) \mod 7) = (y_1, y_2, y_3) = (1, 0, 4)$ a $\det(A) \mod 7 = d = 4$,

nad Z_{11} je riešenie $(6, 3, 6)$ a preto $x_1 = \frac{6.4}{4} = \frac{2}{4}$, $x_2 = \frac{3.4}{4} = \frac{1}{4}$ a $x_3 = \frac{6.4}{4} = \frac{2}{4}$, t.j. v tomto prípade máme $(y_1, y_2, y_3) = (2, 1, 2)$ a $d = 4$,

nad Z_{17} je riešenie $(8, 12, 13)$ a preto $x_1 = \frac{8.3}{3} = \frac{7}{3}$, $x_2 = \frac{12.3}{3} = \frac{2}{3}$ a $x_3 = \frac{13.3}{3} = \frac{5}{3}$, t.j. v tomto prípade máme $(y_1, y_2, y_3) = (7, 2, 5)$ a $d = 3$.

Teraz teda potrebujeme nájsť $\det(A_1)$, $\det(A_2)$, $\det(A_3)$ a $\det(A)$ nad Z tak, aby

$\det(A_1)$	$\det(A_2)$	$\det(A_3)$	$\det(A)$	
1	0	4	4	mod 7
2	1	2	4	mod 11
7	2	5	3	mod 17
211	903	277	1159	v Z
211	-406	277	-150	v Z, so znamienkom

Riešenie nad Q teda je

$$x_1 = \frac{211}{-150} = -\frac{211}{150}, \quad x_2 = \frac{-406}{-150} = \frac{203}{75}, \quad x_3 = \frac{277}{-150} = -\frac{277}{150}.$$

Ak je v niektorom riadku $\det(A) = 0$ (ale nie vo všetkých troch), to principiálne nevadí. Na nájdenie $\det(A_1)$, $\det(A_2)$, $\det(A_3)$ by sme ale nemohli použiť riešenie systému nad príslušným Z_p, lebo tam také riešenie nemusí existovať, a ak existuje, nie je jednoznačne určené. V takom prípade môžeme vypočítať príslušné determinantny $\det(A_1)$, $\det(A_2)$, $\det(A_3)$ priamo.

Ešte sa vráťme k odhadu, ktorý sme použili pri výpočte ($|B| < 590$). Tento odhad sme získali pomocou Hadamarovej nerovnosti, ktorá hovorí, že ak $A = ||a_{ij}||_{n\times n}$ je štvorcová matica a $||\alpha_i|| = \sqrt{a_{i1}^2 + \cdots + a_{in}^2}$ sú dĺžky riadkov ako vektorov (pracujeme s reálnymi číslami), tak platí:

$$|det(A)| \leq ||\alpha_1||.||\alpha_2||.\cdots.||\alpha_n||$$

(Objem $n-$rozmerného kvádra je menší alebo rovný ako súčin dĺžok jeho strán, rovnosť platí v prípade kolmých stien.)

Keďže $\det(A) = \det(A^T)$, pri odhade môžeme použiť aj stĺpce (môžeme skúsiť oboje a zobrať lepší výsledok).

V našom prípade sme použili stĺpce. Pre $\det(A)$ platí:

$$\left|\det\begin{pmatrix} 9 & 3 & -3 \\ -1 & 4 & 5 \\ 3 & -1 & 7 \end{pmatrix}\right| \leq \sqrt{81+1+9}\sqrt{9+16+1}\sqrt{9+25+49} = \sqrt{91.26.83} = \sqrt{196378} = 443.15$$

Dĺžka vektora pravej strany je $\sqrt{1+9+36} = \sqrt{46}$ a preto je vidieť, že keď budeme počítať odhad pre $\det(A_2)$, tak dostaneme väčšiu hodnotu ako pri odhade pre ostatné determinanty (na získanie správneho odhadu, ktorý "pokryje" všetky determinanty treba spomedzi dĺžok stĺpcov "vyhodiť" najmenšiu a ostatné vynásobiť. Toto je trik, ktorý sa v prípade práce s riadkami nedá uplatniť). V našom prípade teda bude "najnepriaznivejší" odhad $\det(A_2) \leq \sqrt{91.46.83} < 590$.

Pre "reálne" použitie uvedenej metódy sa samozrejme "nehráme" s malými číslami, preto sa zvyčajne volia prvočísla pi tak, aby boli menšie ako je slovo počítača na ktorom sa pracuje, ale čo najbližšie k nemu, t.j. napr. pre 32 bitové počítače (2^32 je asi 4.10^9) sa prvočísla volia tak, aby napr. $10^9 < p_i < 4.10^9$ - aby sa v poli Z_{p_i} dalo počítať rýchlo, a aby to "zachytávalo" čo najväčší rozsah).

Dá sa ukázať, že uvedená metóda pracuje v čase $O(m^2n^3 + mn^4)$, kde n je počet rovníc (premenných) a m je "presnosť" koeficientov sústavy rovníc - maximum z presností všetkých koeficientov sústavy, (presnosť čísla je počet "cifier", v tomto prípade ale môžeme brať cifry veľkosti menej ako 10^9 a teda ak vieme absolútne hodnoty čísiel v rozšírenej matici sústavy dobre ohraničiť číslom D, tak m je zhruba $\log_{10^9} D$, čo je v podstate $\frac{\text{počet dekadických cifier } D}{9}$). V tomto odhade sa využíva fakt, že n je podstatne menšie ako 10^9, v praxi sa možno vyskytnú sústavy rovníc s počtom premenných (a rovníc) do 50, väčšie sústavy sa zriedka počítajú presne (t.j. ako zlomky), pre väčšie sústavy sa používajú iteračné numerické metódy, počítajúce riešenia s istou presnosťou.

2.3 MHI schéma pre $Z[x]$ s použitím CRT

Potrebujeme vypočítať hodnotu výrazu $\mathbf{e}(x_1, \ldots, x_n) \in Z[x_1, \ldots, x_n]$ vo vstupných hodnotách $a_1(x), \ldots, a_n(x) \in Z[x]$. Výsledok bude polynóm $f(x) = f_m x^m + \cdots + f_1 x + f_0$.

Najprv musíme získať odhady na možné koeficienty výsledku, t.j. B také, aby pre všetky $i = 0, \ldots, m$ platilo $|f_i| < B$. (toto je opäť ad-hoc záležitosť, ktorá závisí od výrazu **e** a od vstupných hodnôt).

1. krok: Teraz máme 2 možnosti. Ak vieme, že koeficienty výsledku budú nezáporné, zvolíme istý počet navzájom nesúdeliteľných čísiel $m_1, \ldots, m_k$ tak, aby $B \leq m_1.\cdots.m_k$.

Ak je možné, že koeficienty výsledku môžu byť aj kladné aj záporné, (t.j. niektorý z $f_0, \ldots, f_m$ môže byť aj kladný a niektorý z $f_0, \ldots, f_m$ záporný a/alebo vopred nevieme, či takáto možnosť nastane), voľbu musíme urobiť tak, aby $2B \leq m_1.\cdots.m_k$. Budeme používať nasledujúce označenie: ak $a(x) = a_n x^n + \cdots + a_1 x + a_0$, tak $a(x) = a_n x^n + \cdots + a_1 x + a_0 \mod m$ (pre $m \in N$) znamená

$$(a_m \mod m)x^m + \cdots + (a_1 \mod m)x + (a0 \mod m)$$

1. krok: Pre $i = 1, \ldots, k$ vypočítame hodnoty $e_i(x) = e(a_1(x) \mod m_i, ..., a_n(x) \mod m_i) \mod m_i$ - t.j. každé $e_i(x) \in Z_{m_i}[x]$, povedzme

$$e_i(x) = e_{im}x^m + e_{im-1}x^{m-1} + \cdots + e_{i1}x + e_{i0}, \quad 0 \leq e_{ij} < m_i$$

Tu (bez újmy na všeobecnosti) predpokladáme, že výsledky $e_i(x)$ vo všetkých $Z_{m_i}[x]$ majú rovnaký stupeň m, ak nie, doplníme ich členmi s nulovými koeficientami tak, aby sme mali rovnaký "formálny" stupeň.

2. krok: Pomocou algoritmu z dôkazu CRT nájdeme pre každé $i = 0, \ldots, m$ (jediné) $f_i \in Z$, ktoré spĺňa všetky kongruencie $f_i \equiv e_{ij} \mod m_j$ (pre $j = 1, \ldots, k$) a tiež podmienku $0 \leq f_i < m_1.\cdots.m_k$. Ak vieme, že všetky výsledné koeficienty majú byť kladné, sú f_i hľadané koecicienty. Ak vopred nepoznáme znamienka koeficientov, t.j. naša voľba $m_1, \ldots, m_k$ bola taká, aby $2B \leq m_1.\cdots.m_k$ a dostali sme hodnotu $f_i \geq B$, správny koeficient je záporný a preto ho zmeníme na $f_i := -(m_1.\cdots.m_k - f_i)$.

2.4 MHI schéma pre $Z[x]$ s použitím interpolačno-evaluačnej schémy v $Z_p[x]$

V tomto prípade riešime rovnakú úlohu ako v paragrafe 2.3. V metóde, ktorú sme tam použili nie je špecifikované, ako sa majú v prvom kroku vypočítať hodnoty $e_i(x) = \mathbf{e}(a_1(x) \mod m_i, \ldots, a_n(x) \mod m_i) \mod m_i$ v $Z_{m_i}[x]$. Teraz namiesto všeobecných čísiel $m_1, \ldots, m_k$ použijeme prvočísla $p_1, \ldots, p_k$, vďaka čomu sú Z_{p_i} polia a preto v nich môžeme použiť počítanie podľa metódy z paragrafu 2.1.

Všetky ostatné časti výpočtu urobíme rovnako, ako v paragrafe 2.3.

Príklad: Vypočítajme determinant matice $A(x) = \begin{pmatrix} 1+3x & 4-2x \\ 3-x & 8+5x \end{pmatrix}$

Normálny výpočet nám dá výsledok $\det(A) = (1+3x)(8+5x) - (3-x)(4-2x) = 13x^2 + 39x - 4$.

Použime ohraničenie na koeficienty výsledku $B = 50$, keďže sa vo výsledku môžu objaviť aj kladné koeficienty, aj záporné koeficienty, musíme zvoliť prvočísla tak, aby ich súčin bol viac ako $2B = 100$, napr. $p_1 = 3$, $p_2 = 5$ a $p_3 = 7$, $3.5.7 = 105 > 100$.

Ak označíme $A_3(x) = A(x) \mod 3$, $A_5(x) = A(x) \mod 5$, $A_7(x) = A(x) \mod 7$ (všetky prvky matice prevedieme do príslušného $Z_p[x]$), dostaneme

$$A_3(x) = \begin{pmatrix} 1 & 1+x \\ 2x & 2+2x \end{pmatrix} \quad A_5(x) = \begin{pmatrix} 1+3x & 4+3x \\ 3+4x & 3 \end{pmatrix} \quad A_7(x) = \begin{pmatrix} 1+3x & 4+5x \\ 3+6x & 1+5x \end{pmatrix}$$

Teraz vypočítame $\det(A_3(x))$, $\det(A_5(x))$ a $\det(A_7(x))$ pomocou evaluačno-interpolačnej metódy z 2.1. Vieme, že $\det(A)$ je polynóm stupňa nanajvýš 2, preto v každom poli použijeme 3 argumenty, 0, 1, 2:

Pre prvočíslo 3 máme (tu si treba uvedomiť, že pre polynómy stupňa nanajvýš 2 vieme použiť ľubovoľné pole, ktoré má aspoň 3 prvky, t.j. už Z_3 je v poriadku):

$$A_3(0) = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \quad \text{t.j.} \quad \det A_3(0) = 2$$

$$A_3(1) = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \quad \text{t.j.} \quad \det A_3(1) = 0$$

$$A_3(2) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{t.j.} \quad \det A_3(2) = 0$$

Pri interpolácii teda v hľadáme v $Z_3[x]$ polynóm $f(x)$ druhého stupňa taký, aby $f(0) = 2$, $f(1) = 0$ a $f(2) = 0$, čo je $2 + x^2$.

Pre prvočíslo 5 máme:

$$A_5(0) = \begin{pmatrix} 1 & 4 \\ 3 & 3 \end{pmatrix} \quad \text{t.j.} \quad \det A_5(0) = 1$$

$$A_5(1) = \begin{pmatrix} 4 & 2 \\ 2 & 3 \end{pmatrix} \quad \text{t.j.} \quad \det A_5(1) = 3$$

$$A_5(2) = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} \quad \text{t.j.} \quad \det A_5(2) = 1$$

Pri interpolácii teda v hľadáme v $Z_5[x]$ polynóm $f(x)$ druhého stupňa taký, aby $f(0) = 1$, $f(1) = 3$ a $f(2) = 1$, čo je $1 + 4x + 3x^2$.

Pre prvočíslo 7 máme:

$$A_7(0) = \begin{pmatrix} 1 & 4 \\ 3 & 1 \end{pmatrix} \quad \text{t.j.} \quad \det A_7(0) = 3$$

$$A_7(1) = \begin{pmatrix} 4 & 2 \\ 2 & 6 \end{pmatrix} \quad \text{t.j.} \quad \det A_7(1) = 6$$

$$A_7(2) = \begin{pmatrix} 0 & 0 \\ 1 & 11 \end{pmatrix} \quad \text{t.j.} \quad \det A_7(2) = 0$$

Pri interpolácii teda v hľadáme v $Z_7[x]$ polynóm $f(x)$ druhého stupňa taký, aby $f(0) = 3$, $f(1) = 6$ a $f(2) = 0$, čo je $3 + 4x + 6x^2$.

Čiže na výpočet hľadaného polynómu $f(x) = f_0 + f_1x + f_2x^2$ nad Z teraz budeme riešiť 3 úlohy pomocou algoritmu z CRT, zaznačíme si v tabuľke (vždy jeden stĺpec zodpovedá jednej úlohe)

$2 + x^2$	2	0	1	mod 3
$1 + 4x + 3x^2$	1	4	3	mod 5
$3 + 4x + 6x^2$	3	4	6	mod 7
	f_0	f_1	f_2	v Z

Pomocou CRT dostaneme

$f_0 = 101$, čo je viac ako $B = 50$ a preto musíme zobrať $f_0 = -(105 - 101) = -4$

$f_1 = 39$, čo je menej ako $B = 50$, takže je v poriadku

$f_2 = 13$, čo je menej ako B = 50, takže je v poriadku

Hľadaný výsledok teda je det $\begin{pmatrix} 1+3x & 4-2x \\ 3-x & 8+5x \end{pmatrix} = -4 + 39x + 13x^2$, čo je "potvrdením" nášho pôvodného výpočtu.

Pri nejakých príležitostiach používať aj iný spôsob "zdvihu" ako pomocou CRT, jedna z ďalších metód je použitie $p-$adických zápisov čísiel a tzv. Henselovej lemy (sú známe varianty lineárna a kvadratická). Túto metódu popíšeme a použijeme pri algoritme na faktorizáciu polynómov nad Z (kde budeme vychádzať z faktorizácie daného polynómu vo vhodnom $Z_p[x]$).

3 Algoritmy na násobenie celých čísiel

3.1 Karacubov algoritmus

Tento algoritmus bol prvýkrát popísaný v roku 1963. Budeme násobiť čísla a, b, pričom budeme predpokladať, že sú približne rovnako veľké a že majú párny počet cifier v $g-$adickom zápise (výhodné g môže byť 10^{10}, čo je najväčšia mocnina 10 pod 2^{32}). Ak to tak nie je, doplňme zo začiatku čísla a, b nulami.

Nech potom $a = kB + l$, $b = sB + t$, kde l, t sú "prvé" polovice cifier, k, s sú druhé polovice cifier a B je správna mocnina čísla g. (napr. ak $g = 10$, $a = 345$, $b = 2313$, najprv budeme a chápať ako 0345, B bude 10^2 lebo polovica cifier z oboch čísiel a, b je 2 a teda $a = 3 \cdot 100 + 45$, $b = 23 \cdot 100 + 13$).

Karacubov algoritmus je založený na rovnici

$$a \cdot b = (kB + l)(sB + t) = ksB^2 + (kt + ls)B + lt$$

a fakte, že $(k - l)(s - t) = ks - ls - kt + lt$, t.j.

$$a \cdot b = (kB + l)(sB + t) = ksB^2 + (ks + lt - (s - t)(k - l))B + lt$$

čo znamená, že na výpočet uvedeného súčinu potrebujeme urobiť 3 násobenia (ks, lt a $(k - l)(s - t)$) namiesto 4 ako pri prvom vzorci.

Predpokladajme, že počet cifier oboch čísiel a, b je 2^p. V takom prípade môžeme uvedený "trik" urobiť rekurzívne pomocou algoritmu s nasledujúcim pseudokódom:

```
function karacuba(a, b,p)  // a= a_{n-1}a_{n-2}...a_1a_0;
                           // b= b_{n-1}b_{n-2}...b_1b_0; n=2^p
1  if (p == 0) return a*b  // cisla a,b su "jednociferne"
   else
    begin
2     q:= p-1;
3     k := a_{p-1}a_{p-2}...a_{q+1}a_q   l := a_{q-1}a_{q-2}...a_1a_0
4     s := b_{p-1}b_{p-2}...b_{q+1}b_q   t := b_{q-1}b_{q-2}...b_1b_0
5     u := karacuba(k,s,q)
6     v := karacuba(l,t,q)
7     w := karacuba((k-l),(s-t),q)
```

```
8     return u*B^2+(u+v-w)*B+v
    end
```

Tu je dobre si uvedomiť, že vynásobiť číslom B^2 a/alebo číslom B znamená pripísať za číslo istý počet núl a sčítanie $n-$ciferných čísiel vieme urobiť v čase $O(n)$, sčítaní podľa uvedeného algoritmu robíme 6p ($p = \log n$) (v riadku 7 dve, v riadku 8 čtyri), aj keď my tu sčítavame možno $2n$ ciferné čísla, stále počet ohraničený hranicou $O(n \log n)$.

Spočítajme počet násobení (násobíme vždy len čísla menšie ako B), ktoré potrebujeme urobiť.

Podľa algoritmu platí:

$$M(a, b, p) = M(k, s, p-1) + M(l, t, p-1) + M((k-l), (s-t), p-1)$$

Vzhľadom na charakter práce algoritmu, vidíme, že na spočítanie počtu násobení vlastne potrebujeme len jeden vstupný údaj a to je číslo p (=$\log_2 n$), takže budeme písať len

$$M'(p) = M'(p-1) + M'(p-1) + M'(p-1) = 3M'(p-1)$$

t.j.

$$M'(p) = 3^p M'(0)$$

Vieme, že $M'(0) = 1$ a preto

$$M'(p) = 3^p$$

.

$M'(p)$ je počet násobení, ktoré potrebujeme na vynásobenie dvoch $n-$ciferných čísiel, kde $p = \log_2 n$, čiže ak tento počet násobení prepíšeme ako funkciu M'' čísla n (t.j. $M''(n) = M'(log_2 n)$), dostaneme

$$M''(n) = 3^{\log_2 n} = (2^{\log_2 3})^{\log_2 n} = 2^{(\log_2 3)\cdot(\log_2 n)} = 2^{\log_2 n^{\log_2 3}} = n^{\log_2 3}$$

3.2 Rýchla diskrétna Fourierova transformácia

Iný spôsob násobenia veľkých čísiel je založený na tom, že sa na číslo dívame ako na funkčnú hodnotu polynómu. T.j. ak $a = a_{n-1}a_{n-2}...a_1a_0$, a $a = b_{n-1}b_{n-2}...b_1b_0$ sú zápisy čísiel v $g-$adicekj sústave, potom môžeme pracovať s polynómami $a(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2}...a_1x + a_0$, $b(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2}...b_1x + b_0$ a vieme, že $a = a(g)$, $b = b(g)$ a preto ak položíme $c(x) = a(x)b(x)$, tak $c(g) = a \cdot b$.

Preto jedna z možností, ako násobiť čísla je vynásobiť príslušné polynómy a do výsledku dosadiť číslo g. Celočíselné polynómy je vhodné násobiť pomocou MHI pre $Z[x]$ s pomocou evaluačno-interpolačnej schémy v $Z_p[x]$ podľa 2.4. Preto sa začneme zaoberať násobením polynómov v poli $Z_p[x]$ (i keď zo začiatku je jednoduchšie sa neobmedzovať poľom, prvý príklad na ktorom budeme demonštrovať dynamiku nových myšlienok urobíme nad poľom komplexných čísiel C).

3.2.1 Násobenie v $Z_p[x]$ pomocou evaluačno-interpolačnej schémy I

$a(x) = a_{n-1}x^{n-1} + \cdots + a_1x + a_0, b(x) = b_{n-1}x^{n-1} + \cdots + b_1x + b_0 \in F[x]$. Zvoľme $2n$ rôznych prvkov $a_1, \ldots, a_{2n} \in F$.

Vieme, že polynóm $c(x) = a(x) \cdot b(x)$ má stupeň $\leq 2n-2$, takže ho nájdeme pomocou interpolácie v $2n$ rôznych bodoch (stačilo by $2n-1$). Vypočítajme funkčné hodnoty

$$f_1 = a(a_1), \ldots, f_{2n} = a(a_{2n})$$
$$g_1 = b(a_1), \ldots, g_{2n} = b(a_{2n})$$

a potom čísla

$$d_1 = f_1g_1, \ldots, d_{2n} = f_{2n}g_{2n}$$

a pomocou Newtonovej interpolácie nájdime interpolačný polynóm $c(x) = c_{2n-2}x^{2n-2} + c_{2n-2}x^{2n-2}...c_1x + c_0$ taký, že

$$c(a_1) = d_1, \ldots, c(a_{2n}) = d_{2n}$$

Vidíme, že

1. Výpočet čísiel $f_1, \ldots, g_{2n}$ pomocou hornerovej schémy je $4nO(n)$
2. výpočet čísiel $d_1, \ldots, d_{2n}$ je $2n$ násobení
3. interpolácia v $2n$ bodoch je $O((2n)^2)$.

Tento algoritmus teda potrebuje $8O(n^2) = O(n^2)$ operácií, podľa veľkosti konštanty sa zdá, že je vlastne horší ako štandardný algoritmus na násobenie polynómov.

3.2.2 FFT - Forward Fourier Transform

V tejto časti si ukážeme, ako sa dá za vhodných okolností podstatne urýchliť výpočet dosadenia väčšieho počtu prvkov. Myšlienka spočíva v nájdení vhodnej vlastnosti množiny prvkov, ktoré dosadzujeme do polynómu.

Povieme, že množina $S = \{a_1, ..., a_{2n}\}$ je symetrická, ak $0 \notin S$ a $s \in S \Rightarrow -s \in S$ (jeden z predpokladov bude, že nebudeme pracovať s poliami v ktorých je $1 + 1 = 0$ a teda pre nenulové s je $s \neq -s$). Budeme predpokladať, že máme symetrickú množinu S zapísanú v tvare $S = \{\pm s_1 \ldots, \pm s_n\}$

Predpokladajme, že $\mathrm{st}(a(x) \leq 2n-1$ (ide o formálny stupeň, t.j. niektoré najvyššie mocniny môžu mať nulové koeficienty a takto potrebujeme, aby mal daný polynóm $2n$ "členov, sčítancov, termov" - možno s nulovými koeficientami).

Rozpíšme tento polynóm v nasledujúcom tvare

$$\begin{aligned}
&a(x) = a_0 + a_1 + a_{2n-1}x^{2n-1} + a_{2n-2}x^{2n-2} = \\
&= (a_0 + a_2x^2 + a_4x^4 + \cdots a_{2n-2}x^{2n-2}) + (a_1x + a_3x^3 + a_5x^5 + \cdots a_{2n-1}x^{2n-1}) = \\
&= (a_0 + a_2x^2 + a_4x^4 + \cdots a_{2n-2}x^{2n-2}) + x(a_1 + a_3x^2 + a_5x^4 + \cdots a_{2n-1}x^{2n-2}) = \\
&= (a_0 + a_2y + a_4y^2 + \cdots a_{2n-2}y^{n-1}) + x(a_1 + a_3y + a_5y^2 + \cdots a_{2n-1}y^{n-1})
\end{aligned}$$

kde $y = x^2$ Ak položíme $u(y) = a_0 + a_2y + a_4y^2 + \cdots a_{2n-2}y^{n-1}$ a $v(y) = a_1 + a_3y + a_5y^2 + \cdots a_{2n-1}y^{n-1}$, tak dostávame

$$a(x) = u(y) + xv(y)$$

Koľko operácií budeme potrebovať na výpočet funkčných hodnôt $f_1 = a(s_1), f_2 = a(-s_1) \ldots, f_{2n-1} = a(s_n, f_{2n} = a(-s_n))$?

Použime výpočet typu

$$\begin{aligned} f_1 &= a(s_1) = & u(s_1^2) + s_1 v(s_1^2) \\ f_2 &= a(-s_1) = & u((-s_1)^2) - s_1 v((-s_1)^2) \\ & & \ldots \\ f_{2n-1} &= a(s_n) = & u(s_n^2) + s_n v(s_n^2) \\ f_{2n} &= a(-s_n) = & u((-s_n)^2) + s_n v((-s_n)^2) \end{aligned}$$

Dôležité je, že $s_1^2 = (-s_1)^2$ ($\ldots$ $s_n^2 = (-s_n)$), to znamená, že ak vypočítame čísla $u_i = u(s_i^2)$, $v_i = v(s_i^2)$ a $d_i = s_i v_i$, tak

$$\begin{aligned} f_1 &= u_1 + d_1 \\ f_2 &= u_1 - d_1 \\ & \ldots \\ f_{2n-1} &= u_n + d_n \\ f_{2n} &= u_n - d_n \end{aligned}$$

Ak teda označíme počet násobení, ktoré potrebujeme urobiť, aby sme vypočítali funkčné hodnoty polynómu $a(x)$ v bodoch $S = \{\pm s_1 \ldots, \pm s_n\}$ znakom $M(S)$, tak z uvedenej schémy dostaneme

$$M(S) = 2M(S') + n$$

pričom $S' = \{s_1^2 \ldots, s_n^2\}$. Člen $2M(S')$ pochádza z toho, že do polynómov $u(y), v(y)$ dosadzujeme prvky z $n-$prvkovej množiny S' a člen n pochádza z násobení $d_i = s_i \cdot v(s_i^2)$, ktorých musíme urobiť n. Vzhľadom na to, že množina S má $2n$ prvkov a S' má n prvkov, môžeme poslednú rovnicu napísať s týmito argumentami, t.j.

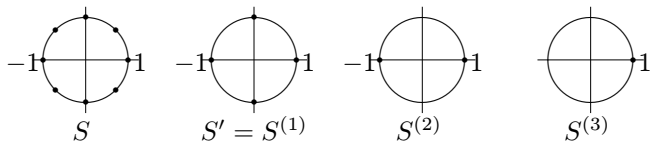
$$M(2n) = 2M(n) + n \tag{1}$$

Skutočná dynamika uvedeného postupu sa ukáže vtedy, ak bude množina S *dedične* symetrická, t.j. ak počet prvkov množiny bude mocnina 2, t.j. 2^k a bude symetrická nielen množina S, ale aj $S' = S^{(1)} = \{s_1^2 \ldots, s_n^2\}$ (ktorá má 2^{k-1} prvkov) a aj množina $S^{(2)} = \{s_1^4 \ldots, s_n^4\} = \{a^2; a \in S'\}$ (ktorá má 2^{k-2} prvkov) je symetrická, $\ldots$

Vďaka symetrii množiny S' totiž budeme môcť použiť uvedený trik aj pri dosadzovaní prvkov z množiny S' do polynómov $u(y)$ a $v(y)$, t.j. tieto "rozdelíme" každý na dva a pri každom použijeme rovnakú schému. Dedičná symetria potom umožní robiť tento trik "rekurzívne", až kým sa nedostaneme ku polynómu stupňa 0, t.j. konštante, kedy vlastne nič nenásobíme, ale len vrátime hodnotu nultého koeficienta.

Ukážme si príklad na vhodnú množinu S pre komplexné čísla. Chceme dosadzovať do polynómu stupňa $7 = 2^3 - 1$. Zoberme "základné" riešenie ω rovnice $x^8 = 1$ v komplexných číslel, t.j. $\omega = \cos(45^o) + i.\sin(45^0)$ a $S = \{1 = \omega^0, \omega, \omega^2, \ldots, \omega^7\}$. Zrejme $1 = -\omega^4$ a preto aj $\omega^1 = -\omega^5$, $\omega^2 = -\omega^6$ a $\omega^3 = -\omega^7$,

t.j. S je symetrická množina. Ale $S^{(1)} = S' = \{a^2; a \in S\} = \{1, \omega^2, \omega^4, \omega^6\}$ je tiež symetrická, množina $S^{(2)} = \{a^4; a \in S\} = \{1, \omega^4\} = \{\pm 1\}$ je tiež symetrická a nakoniec máme množinu $S^{(3)} = \{a^8; a \in S\} = \{1\}$ - táto už má len jeden prvok, ktorý ale dosadzujeme do polynómov stupňa 0 (majú jeden "člen").



S $\quad S' = S^{(1)}$ $\quad S^{(2)}$ $\quad S^{(3)}$

Pre pole $Z_{17}[x]$ môžeme pre polynómy stupňa 16 zobrať množinu $S = \{1, 16, 2, 15, 3, 14, 4, 13, 5, 12, 6, 11, 7, 10, 8, 9\}$. Potom $S' = S^{(1)} = \{1, 4, 9, 16, 8, 2, 15, 13\} = \{1, 16, 4, 13, 2, 15, 8, 9\}$, $S^{(2)} = \{1, 16, 4, 13\}$, $S^{(3)} = \{1, 16\}$, $S^{(4)} = \{1\}$.

Pre polynómy stupňa najviac 7 zoberieme množinu $S = \{1, 4, 9, 16, 8, 2, 15, 13\}$, čo je vlastne množina S' z predošlého príkladu.

V oboch týchto príkladoch je využitý fakt, že $17 = 2^4 + 1$, t.j. grupa $(Z_{17} \setminus \{0\}, \odot)$ je 16 prvková cyklická grupa. Preto S môže obsahovať všetky prvky zo $Z_{17} \setminus \{0\}$ (je to vlastne množina všetkých riešení rovnice $x^{16} = 1$ v Z_{17}, $S^{(1)}$ je množina všetkých riešení rovnice $x^8 = 1$ v Z_{17}, $S^{(2)}$ je množina všetkých riešení rovnice $x^4 = 1$ v Z_{17}, $S^{(3)}$ je množina všetkých riešení rovnice $x^2 = 1$ v Z_{17} a nakoniec $S^{(4)}$ je množina všetkých riešení rovnice $x^1 = 1$ v Z_{17}).

$$f(x) = 1 + 3x + x^2 + 9x^3 + 12x^4 + 6x^5 + 15x^6 + 2x^7$$

za x dosadzujeme z $S = \{1, 16, 4, 13, 2, 15, 8, 9\} = \{\pm 1, \pm 4, \pm 2, \pm 8\}$

$1 + x^2 + 12x^4 + 15x^6$ $\qquad$ $3x + 9x^3 + 6x^5 + 2x^7$

$1 + y + 12y^2 + 15y^3$ $\qquad$ $x(3 + 9y + 6y^2 + 2y^3)$

za y dosadzujeme z $S = \{1, 16, 4, 13\} = \{\pm 1, \pm 4\}$

$1 + 12y^2$ $\quad$ $y + 15y^3$ $\quad$ $3 + 6y^2$ $\quad$ $9y + 2y^3$

$1 + 12z$ $\quad$ $y(1 + 15z)$ $\quad$ $3 + 6z$ $\quad$ y$(9 + 2z)$

za z dosadzujeme z $S = \{1, 16\} = \{\pm 1\}$

$1 \quad 12z$ $\quad$ $1 \quad 15z$ $\quad$ $3 \quad 6z$ $\quad$ $9 \quad 2z$

$1 \quad z(12)$ $\quad$ $1 \quad z(15)$ $\quad$ $3 \quad z(6)$ $\quad$ $9 \quad z(2)$

za w (ktoré nevidieť, lebo polynómy sú stupňa 0) dosadzujeme z $S = \{1\}$

Takže postupne dostávame hodnoty (teraz ideme v opačnom poradí)

$1 \quad 12$ $\quad$ $1 \quad 15$ $\quad$ $3 \quad 6$ $\quad$ $9 \quad 2$

t.j. (dosadenie $\{\pm 1\}$)

1 ± 12 $\quad$ 1 ± 15 $\quad$ 3 ± 6 $\quad$ 9 ± 2

$13_1,\ 6_{-1}$ $\quad$ $16_1,\ 3_{-1}$ $\quad$ $9_1,\ 14_{-1}$ $\quad$ $11_1,\ 7_{-1}$

t.j. (dosadenie $\{\pm 1, \pm 4\}$, vieme, že $1 = (\pm 1)^2$, $-1 = (\pm 4)^2$)

$13 \pm 16,\ 6 \pm 4 \cdot 3$ $\quad$ $9 \pm 11,\ 14 \pm 4 \cdot 7$

$12_1,\ 14_{-1},\ 1_4,\ 11_{-4}$ $\quad$ $3_1,\ 15_{-1},\ 8_4,\ 3_{-4}$

t.j. (dosadenie $\{\pm 1, \pm 4, \pm 2, \pm 8\}$, vieme, že $1 = (\pm 1)^2$, $-1 = (\pm 4)^2$, $4 = (\pm 2)^2$, $-4 = (\pm 8)^2$)

$12 \pm 3,\ 14 \pm 4 \cdot 15,\ 1 \pm 2 \cdot 8,\ 11 \pm 8 \cdot 3$

$15_1,\ 9_{-1},\ 6_4,\ 5_{-4},\ 0_2,\ 2_{-2},\ 1_8,\ 4_{-8}$

Čísla s indexami znamenajú funkčné hodnoty, t.j. napr. 15_1 vlastne znamená $f(1) = 15, \ldots$ Pri dosadzovaní sme v každom riadku okrem prvého potrebovali 4 násobenia ($4 = \frac{\mathrm{st}\, f}{2}$), takéto riadky sú 3, $3 = \log_2(\mathrm{st}\, f + 1)$. V prvom riadku nič nenásobíme, dosadzujeme do polynómov stupňa 0. Spolu je to 12 násobení oproti 64, ktoré potrebujeme, keď použijeme hornerovu schému.

Pozrime sa na to, v ktorých konečných poliach sa dá nájsť rozumne veľká dedične symetrická množina. Nech Z_p je konečné pole, napíšme $p = 2^n m + 1$, kde m je nepárne číslo. Vieme, že $(Z_p \setminus \{0\}, \odot)$ je cyklická grupa, ktorá má $2^n m$ prvkov. Nech α je jej generátor, položme $\omega = \alpha^m$. Rád prvku ω je 2^n. Platí

nasledujúce tvrdenie

Veta 6 *Množina*

$$S = [\omega] = \{1, \omega, \omega^2, \ldots, \omega^{2^n-1}\}$$

je dedične symetrická množina.

Dôkaz. Dokážeme nasledujúce tvrdenia, z ktorých toto tvrdenie jednoducho vyplýva

1. Nech $1 \leq k \leq n$ a prvok β má rád 2^k. Potom je množina

$$[\beta] = \{1, \beta, \beta^2, \ldots, \beta^{2^k-1}\}$$

   symetrická.

2. Ak $1 \leq k \leq n$, tak ω^{2^k} má rád 2^{n-k}

Ak totiž platia obe tieto tvrdenia, tak množiny $S = [\omega]$, $S' = S^{(1)} = [\omega^2]$, $S^{(2)} = [(\omega^2)^2] = [\omega^{2^2}]$, $S^{(3)} = [(\omega^4)^2] = [\omega^{2^3}]$,..., $S^{(n-1)} = [(\omega^{2^{n-2}})^2] = [\omega^{2^{n-1}}] = \{-1, 1\}$ všetko symetrické množiny, presne ako to vyžaduje definícia.

ad 1) Označme $2N = 2^k$. V poli má rovnica $x^2 = 1$ dve riešenia, ± 1. Keďže $(\beta^N)^2 = \beta^{2N} = 1$ a $\beta^N \neq 1$, tak $\beta^N = -1$. Vidíme, že pre $i = 0, \ldots N-1$ platí $\beta^{N+i} = \beta^N \odot \beta^i = (-1) \odot \beta^i = -\beta^i$. Množina $[\beta]$ je teda symetrická.

ad 2) Rád prvku ω je 2^n. Preto $(\omega^{2^k})^{2^{n-k}} = \omega^{2^k 2^{n-k}} = \omega^{2^n} = 1$. T.j. 2^{n-k} by mohol byť rád prvku ω^{2^k}, len ešte treba skontrolovať, že pre $1 < l < 2^{n-k}$ je $(\omega^{2^k})^l \neq 1$. Ale očividne, $1 < l < 2^{n-k}$ znamená, že $2^k < (2^k) \cdot l < 2^k \cdot 2^{n-k} = 2^n$ a teda $(\omega^{2^k})^l = 1$ je v spore s tým, že rád ω je 2^n. Takže skutočne je rád prvku ω^{2^k} číslo 2^{n-k}. □

Tu je pseudkód algoritmu využívajúceho uvedené myšlienky:
vstup:

polynóm $f(x)$, $\mathrm{st}(f) < 2^n$. Polynóm bude zadaný ako pole `f[i]`, pričom `f[i]` je koeficient pri x^i pre $i = 0, \ldots, 2^n - 1$

číslo n

$\omega \in F$ taká, že v danom poli F má rád 2^n (v pseudókode použijeme pismeno `w`)

výstup: pole `A[i]` veľkosti 2^n, ktoré obsahuje príslušné funkčné hodnoty, t.j. `A[i]`$=f(\omega^i)$ pre $i = 0, \ldots, 2^n - 1$.

```
function fft(f,w,n)
begin
     if (n = 0)
       begin
          A[0] := f[0]
          return A
       end
     else
       begin
         // rozdel polynom f na polynomy u, v
```

```
        j := 0
        for i=0 to 2^n-1 step 2
          begin
           u[j] := f[i]
           v[j] := f[i+1]
           j := j+1
          end
        // u je pole reprezentujuce cast s parnymi exponentami
        // so substituciou y=x**2
        // v je pole reprezentujuce cast s neparnymi exponentami
        // "posunuta" o jedna so substituciou y=x**2
        k:= n-1
        U:=fft(u,w^2,k)
        V:=fft(v,w^2,k)
        // skombimovanie vysledkov
        for i from 0 to 2^k-1
          begin
            tmp:= w^i*V[i]  // RIADOK S NASOBENIM
            A[j]:= U[i]+tmp
            A[k+j]:= U[i]-tmp
          end
      end
      return A

end
```

Použime teraz vzťah (1) na určenie počtu násobení, ktoré potrebujeme na výpočet funkčných hodnôt polynómu $f(x)$ stupňa 2^m-1 v prvkoch $1, \omega, \omega^2, \ldots, \omega^{2^m-1}$ (a ω je prvok rádu 2^m, samozrejme.)

$$\begin{aligned} M(2^m) &= 2M(2^{m-1}) + 2^{m-1} = \\ &= 2(2M(2^{m-2}) + 2^{m-2}) + 2^{m-1} = 2^2 M(2^{m-2}) + 2 \cdot 2^{m-1} = \\ &= 2^2(2M(2^{m-3}) + 2^{m-3}) + 2^{m-1} = 2^3 M(2^{m-3}) + 3 \cdot 2^{m-1} = \\ &\quad \ldots \\ &= 2^m M(2^0) + m \cdot 2^{m-1} = 2^m M(1) + m \cdot 2^{m-1} = \\ &= m \cdot 2^{m-1} \end{aligned}$$

pričom posledná rovnica platí vďaka tomu, že $M(1) = 0$, lebo na dosadenie do polynómu stupňa 0 nepotrebujeme žiadne násobenie. V logaritmickom tvare teda dostávame tvrdenie

$$M(n) = O(n \log n)$$

Takže teraz vieme, že za dobrých okolností vieme číslo z bodu 1 v časti 3.2.1 urobiť pomoocu $O(n \log n)$ operácií v poli.

3.2.3 Interpolácia pomocou Fourierovej trasfmormácie, FFI

V tejto časti sa pozrieme na druhú "podúlohu", ktorú potrebujeme rýchlo riešiť, aby sa dala využiť myšlienka z 3.2.1. Veľkým šťastím je, že FFT poskytuje po istej preformulácii rýchly algoritmus aj na riešenie tejto časti úlohy.

Štandardná úloha (polynomiálnej) interpolácie sa dá v jazyku matíc formulovať nasledovne: majme $n+1$ prvkov $a_0, a_1, \ldots, a_n$ z poľa F, urobme tzv. Vandermondovu maticu

$$V(a_0, a_1, \ldots, a_n) = \begin{pmatrix} 1 & a_0 & a_0^2 & a_0^3 & \ldots & a_0^n \\ 1 & a_1 & a_1^2 & a_1^3 & \ldots & a_1^n \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & a_n & a_n^2 & a_n^3 & \ldots & a_n^n \end{pmatrix}$$

Predpokladajme, že $a_0, a_1, \ldots, a_n$ sú po dvoch rôzne. Ak zobiereme ľubovoľných $n+1$ prvkov $b_0, b_1, \ldots, b_n \in F$ (môžu byť aj všetky rovnaké), podľa vety o interpolácii vieme, že existuje taký polynóm $f(x) = f_n x^n + \cdots + f_1 x + f_0$, že pre všetky $i = 0, \ldots, n$ platí, že $f(a_i) = b_i$, čo z maticového pohľadu znamená, že rovnica

$$V(a_0, a_1, \ldots, a_n) \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} \tag{2}$$

má vždy riešenie $x_0 = f_0, \ldots, x_n = f_n$ a preto je matica $V(a_0, a_1, \ldots, a_n)$ regulárna.

Úloha, ktorú sme riešili v predošlom paragrafe sa v maticovom jazyku dá formulovať takto:

Je daný polynóm $f(x) = f_n x^n + \cdots + f_1 x + f_0$. Pre dané (navzájom rôzne) prvky $a_0, a_1, \ldots, a_n$ vypočítaj súčin

$$V(a_0, a_1, \ldots, a_n) \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}$$

(lebo je vidieť, že vynásobením dostávame $b_0 = f(a_0), \ldots, b_n = f(a_n)$.) O tomto násobení vieme, že v prípade použitia všeobecnej množiny prvkov $a_0, a_1, \ldots, a_n$ a ho vieme urobiť pomocou $O(n^2)$ násobení (hornerova schéma, prípadne použitie štandartných vzorcov na násobenie matíc), keď použijeme dedične symetrickú množinu $a_0, a_1, \ldots, a_n$, vieme to urobiť pomocou $O(n \log n)$ operácií (FFT).

Nájsť interpolačný polynóm znamená nájsť riešenie rovnice (2), vďaka regulárnosti Vandermondovej matice jej riešenie môžeme formálne popísať vzorcom

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} = (V(a_0, a_1, \ldots, a_n))^{-1} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} \tag{3}$$

Hlavný trik umožňujúci využitie FFT spočíva v tom, že pre dobré dedične symetrické množiny vieme inverznú maticu $(V(a_0, a_1, \ldots, a_n))^{-1}$ vypočítať veľmi efektívne, platí totiž veta

Veta 7 *Predpokladajme, že v predošlých úvahách je $n+1 = 2^N$ (t.j. mocnina dvojky) Nech $\omega \in F$ je taký, že* $\mathrm{rad}(\omega) = 2^N$. *Keďže* $[\omega] = \{1, \omega, \omega^2, \ldots, \omega^{2^N-1}\}$, *budeme namiesto* $V(1, \omega, \ldots, \omega^{2^N-1})$ *písať* $V([\omega])$.

Platí:

$$V(\omega)V(\omega^{-1}) = 2^N I$$

kde I je jednotková matica rozmeru $2^N \times 2^N$, t.j.

$$(V(\omega))^{-1} = \frac{1}{2^N} V(\omega^{-1})$$

a čo je dôležité, $[\omega^{-1}]$ je dedične symetrická množina.

Dôkaz. Jediné, čo je dôležité je uvedomiť si, ako vyzerá matica $V(\omega^{-1})$. Kvôli zjednodušeniu, počítajme riadky a stĺpce od nuly, t.j. vrchný riadok bude nultý, ľavý stĺpec bude nultý stĺpec. Potom $j-$tý stĺpec matice $V(\omega^{-1})$ ($j = 0, \ldots, 2^N - 1$) má tvar

$$\begin{pmatrix} 1 = ((\omega^{-1})^0)^j \\ (\omega^{-1})^j \\ \vdots \\ ((\omega^{-1})^{2^N-1})^j \end{pmatrix}$$

Nech teraz $i = j$, počítajme teda $i, i-$tý prvok v súčine $V(\omega)V(\omega^{-1})$. Bude to

$$\sum_{k=0}^{2^N-1} (\omega^i)^k ((\omega^{-1})^k)^i = \sum_{k=0}^{2^N-1} \omega^{(ik-ki)} = \sum_{k=0}^{2^N-1} 1 = 2^N$$

Nech teraz $i \neq j$, počítajme $i, j-$tý prvok v súčine $V(\omega)V(\omega^{-1})$. Bude to

$$\sum_{k=0}^{2^N-1} (\omega^i)^k ((\omega^{-1})^k)^j = \sum_{k=0}^{2^N-1} \omega^{(ik-kj)} = \sum_{k=0}^{2^N-1} \omega^{k(i-j)} = \frac{(\omega^{i-j})^{2^N} - 1}{\omega^{i-j} - 1} = \frac{0}{\omega^{i-j} - 1} = 0$$

lebo $(\omega^{i-j})^{2^N} = \omega^{(i-j)2^N} = (\omega^{2^N})^{i-j} = 1^{i-j} = 1$ a menovateľ v zlomku je určite nenulový.

Súčin $V(\omega)V(\omega^{-1})$ je preto diagonálna matica, ktorá má na diagonále prvky 2^N. □

Vráťme sa teraz ku vyjadreniu (3). Keď sme počítali dosadenie pre prvky z množiny $[\omega]$, podľa vety pre interpoláciu dostaneme

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} = (V([\omega])^{-1} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix} = \frac{1}{2^N} V([\omega^{-1}] \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}$$

kde ale výraz

$$V([\omega^{-1}] \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}$$

predstavuje výpočet dosadenia hodnôt z dedične symetrickej množiny $[\omega^{-1}]$ do polynómu

$$b(x) = b_n x^n + \cdots + b_1 x + b_0 = b_{2^N-1} x^{2^N-1} + \cdots + b_1 x + b_0$$

a to je niečo, čo vieme riešiť pomocou FFT použitím $O(n \log n)$ operácií. Zložky výsledného vektora ešte musíme predeliť číslom 2^N, ale to pridá len n operácií, takže celkový odhad $O(n \log n)$ sa tým nepokazí.

3.2.4 Prvočísla vhodné pre FFT/FFI

Aby sme mohli používať diskrétnu FFT/FFI, je vhodné mať malú zásobu polí - napr. v tvare Z_p, v ktorých sa dá robiť násobenie rozumne veľkých polynómov. Pre 32 bitové počítače budeme používať 10-miestne prvočísla (lebo $2^{32}-1$ je desaťmiestne číslo), vždy je vhodné v poli nájsť aj prvok α s vlastnosťou $\mathrm{rad}(\alpha) = p-1$ (primitívna odmocnina z jednotky, t.j. α je generátor multiplikatívnej grupy $Z_p \setminus \{0\}$. Pomocou prvku α ľahko nájdeme prvok ω potrebný pre FFT/FFI). Prvočísla v tabuľke budeme vždy hľadať v tvare $p = 2^n m + 1$, do tabuľky uvedieme číslo p, exponent n a "najmenšiu" primitívnu odmocninu z 1.

Našťastie, primitívne korene sa dajú nájsť s pravdepodobnosťou $\frac{3}{\pi^2}$, čo je asi 0.3, takže stačí začať prehľadávať čísla $2,\ 3,\ 4, \ldots$, kým nenájdeme primitívnu odmocninu z 1, šanca je vysoká, takže by nemalo trvať dlho ju nájsť. Tiež je dôležité si pripomenúť, že overiť, či $\alpha \in Z_p$ je primitívna odmocnina z 1 nie je ťažké, platí totiž jednoduchá veta, ktorej jedna implikácia je úplne zrejmá a dôkaz druhej je jednoduché cvičenie.

Veta 8 *$\alpha \in Z_p \setminus \{0\}$ je primitívna odmocnina z 1 (generátor multiplikatívnej grupy $Z_p \setminus \{0\}$) práve vtedy, keď pre všetky prvočíselné delitele q čísla $p-1$ je $\alpha^{\frac{p-1}{q}} \neq 1$.*

Vzhľadom na, že nájsť prvočíselný rozklad desať (dvadsať) miestneho čísla nie je problém, poskytuje táto veta naozaj efektívny nástroj na nájdenie prvku α.

Pre 64 bitové počítače je ideálne hľadať prvočísla 20 miestne, na hľadanie primitívnej odmocniny z 1 aj v tomto prípade stačí použiť uvedené kritérium.

32 bitové počítače:

p	$2^n * m + 1$	n	α
2130706433	$2^{24} * 127 + 1$	24	3
2114977793	$2^{20} * 2017 + 1$	20	3
2113929217	$2^{25} * 63 + 1$	25	5
2099249153	$2^{21} * 1001 + 1$	21	3
2095054849	$2^{21} * 999 + 1$	21	11
2088763393	$2^{23} * 249 + 1$	23	5
2077229057	$2^{20} * 1981 + 1$	20	3
2070937601	$2^{20} * 1975 + 1$	20	6
2047868929	$2^{20} * 1953 + 1$	20	13
2035286017	$2^{20} * 1941 + 1$	20	10
2013265921	$2^{27} * 15 + 1$	27	31

64 bitové počítače:

p	$2^n * m + 1$	n	α
15564440312192434177	$2^{59} * 27 + 1$	59	5
10232178353385766913	$2^{57} * 71 + 1$	57	3
10808639105689190401	$2^{57} * 75 + 1$	57	7
13690942867206307841	$2^{57} * 95 + 1$	57	3
17726168133330272257	$2^{57} * 123 + 1$	57	7

V prípade 64 bitových procesorov môžeme teda vo všetkých poliach Z_p pre p z poslednej tabuľky násobiť dva polynómy stupňa menšieho ako 2^{56} - t.j. ich stupeň je asi sedemnásťmiestne číslo (lebo $(2^{56}-1)+(2^{56}-1) < 2^{57}$).

3.3 Rýchle násobenie celých čísiel - 3 prime algorithm

V tejto časti popíšeme algoritmus na vynásobenie dvoch prirodzených čísiel zložený na FFT/FFI schéme.

Algoritmus bude subasymptotický, t.j. bude pracovať len pre čísla v obmedzenom rozsahu, ktorý upresníme v priebehu formulácie algoritmu.

Myšlienka algoritmu je nasledujúca:

Zoberme $B = 10^9$ (pre 32 bitové procesory) alebo $B = 10^{19}$ pre 64 bitové procesory, majme dve čísla a, b zapísané v $B-$adickom zápise, t.j. $a = a_{n-1}B^{n-1} + \cdots + a_1 B + a_0$. Hodnotu n upresníme neskôr.

T.j. pre polynómy $a(x) = a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ a $b(x) = b_{n-1}x^{n-1} + \cdots + b_1 x + b_0$ platí $a = a(B)$, $b = b(B)$. Potom ak $c(x) = a(x) \cdot b(x)$, tak $ab = c(B)$.

To znamená, potrebujeme vynásobiť polynómy s celočíselnými (nezápornými) koeficientami $a(x)$ a $b(x)$.

Celý postup teraz predvedieme pre 32 bitové procesory.

Vyberme k vhodných prvočísiel $B < p_1, \ldots, p_k < 2^{32}$ (k upresníme neskôr) a urobme polynómy $a_1(x) = a(x) \mod p_1 \in Z_{p_1}[x]$, $\ldots$, $a_k(x) = a(x) \mod p_k \in Z_{p_k}[x]$, podobne pre $b(x)$. V Vďaka podmienke $B < p_1, \ldots, p_k$ sa polynómy $a(x)$ sa pri homomorfizme do príslušných Z_{p_i} nezmenia.

Pomocou FFT/FFI vynásobme $d_i(x) = a_i(x)b_i(x)$ v okruhoch $Z_{p_i}[x]$. To vieme spraviť pomocou $kO(n \log n)$ operácií. Ak je teraz

$$d_i(x) = d^i_{2n-2}x^{2n-2} + \cdots + d^i_1 x + d^i_0$$

pre $i = 1, \ldots, k$ (i má - hlavne na pravej strane rovníc - význam indexu, nie exponentu), tak koeficienty $c_0, c_1, \ldots, c_{2n-2}$ polynómu $c(x) = a(x)b(x)$ sú určené kongruenciami

$$\begin{array}{llll|llll|c|llll} c_0 & \equiv d^1_0 & \mod p_1 & & c_1 & \equiv d^1_1 & \mod p_1 & & & c_{2n-2} & \equiv d^1_{2n-2} & \mod p_1 \\ \cdots & & & & \cdots & & & & \cdots & \cdots & & \\ c_0 & \equiv d^k_0 & \mod p_k & & c_1 & \equiv d^k_1 & \mod p_k & & & c_{2n-2} & \equiv d^k_{2n-2} & \mod p_k \end{array}$$

V skutočnosti budeme všetky polynómy $c(x), d_1(x), \ldots, d_k(x)$ považovať za polynómy stupňa $2n-1$. Každý z uvedených systémov kongruencií vieme vyriešiť (a teda nájsť čísla $c_0, c_1, \ldots, c_{2n-1}$) pomocou konečného počtu operácií, ktorý závisí od čísla k (povedzme l), t.j. ak poznáme všetky koeficienty d^j_i polynómov d_i, tak koeficienty $c_0, c_1, \ldots, c_{2n-1}$ vieme nájsť použitím nie viac ako $l(2n)$, t.j. $O(2n) = O(n)$ operácií.

Posledný krok algoritmu je dosadenie B do polynómu $c(x)$, t.j. výpočet $c = c(B) = ab$. Toto tiež vieme urobiť pomocou nie viac ako $O(n \log n)$ operácií.

Ešte potrebujeme určiť vhodný rozsah, kedy vieme tento algoritmus použiť. Najdôležitejšie je zistiť vhodný počet prvočísiel, ktoré potrebujeme použiť.

Jedno obmedzenie dostaneme tak, že sa pozrieme, aké veľké koeficienty môže nadobudnúť polynóm $c(x)$. Pre koeficienty c_i platí

$$c_i = \sum_{k=0}^{i} a_k b_{i-k} < \sum_{k=0}^{i} B \cdot B < (i+1)B^2 < nB^2$$

Ak teda budeme pracovať so vstupnými číslami a, b ktoré sú v $B-$adickom zápise menej ako B cifier, stále pracujeme s číslami, ktoré majú v dekadickom zápise $9B = 9 \cdot 10^9$ cifier. V skutočnosti je toto obmedzenie "slabšie" z obmedzení, ktoré pri našich voľbách musíme urobiť. Jednu vec však vidíme. Ak zvolíme 3 prvočísla s vlastnosťou $B < p_1 < p_2 < p_3$, tak $c_i < p_1 p_2 p_3$ (t.j. pri kongruenciách vždy riešime systém 3 kongruencií na každý koeficient c_i polynómu $c(x)$). Pozrime sa teraz na obmedzenia, ktoré vyplývajú z prvočísiel, ktoré máme k dispozícii podľa našich tabuliek.

Aby sme na výpočet súčinu $d_i = a_i(x)b_i(x)$ pre $i = 1,\ 2,\ 3$ mohli použiť FFT/FFI, potrebujeme, aby polynómy $a_i(x),\ b_i(x)$ mali stupeň menej ako 2^{23}, lebo podľa tabuľky máme k dispozícii 3 prvočísla s exponentom dvojky 24 a viac. Toto obmedzenie je podstatnejšie, lebo samozrejme $2^{23} < B = 10^9$. Takže v skutočnosti, pri použití 32 bitového procesora môžeme pracovať s číslami $a,\ b$, ktoré majú v $B-$adickom zápise menej ako 2^{23} cifier, t.j. v dekadickom zápise je to $9 \cdot 2^{23}$, čo je asi 72 miliónov dekadických cifier.

Obmedzenie pre 64 bitové procesory je teda podľa tabuľky dané počtom 2^{56} $B-$adických cifier, kde $B = 10^{19}$, čo je $19 \cdot 2^{56}$ dekadických cifier (viac ako $13 \cdot 10^{18}$).

4 Faktorizácia polynómov

Veľmi dôležitou vetou je tzv. fundamentálna veta aritmetiky. Jej trochu zovšeobecnená verzia hovorí o tom, že v euklidovskom okruhu sa každý prvok (ktorý nedelí jednotku) dá napísať ako súčin ireducibilných prvkov, pričom takýto rozklad je až na poradie (a asociovanosť) jednoznačný. Veta má základný význam pri štúdiu teórie deliteľnosti, štruktúry euklidovských okruhov a algebraických štruktúr s ním zviazaných.

Táto veta platí pre okruh celých čísiel $(Z, +, \cdot)$, pre okruh polynómov $F[x]$ nad ľubovoľným poľom (oba sú euklidovské okruhy), pre okruh polynómov s celočíselnými koeficientami $(Z[x], +, \cdot)$ (ktorý už nie je euklidovským okruhom, dokonca ani okruhom hlavných ideálov).

Štandartné dôkazy tejto vety sú však nekonštruktívne a preto sa z nich dosť dobre nedá vyrobiť algoritmus na hľadanie takéhoto rozkladu. Bežne sú známe postupy napr., ako hľadať lineárne faktory polynómov s celočíselnými (a tým zároveň aj s racionálnymi) koeficientami a tým rozkladať polynómy do stupňa 3. Keď treba rozložiť polynóm vyššieho stupňa, väčšinou sa používajú sa ad hoc spôsoby.

Na nájdenie koreňov polynómu nad málo prvkovým poľom môžeme použiť metódu úplného prebratia všetkých prvkov. Tým opäť získame rozklad polynómov do 3. stupňa.

Keďže $Q(\sqrt{2})$ je pole, v euklidovskom okruhu $Q(\sqrt{2})[x]$ tiež platí fundamentálna veta aritmetiky, t.j. každý polynóm s koeficientami tvaru $a+b\sqrt{2}$, a, $b \in Q$ sa tiež dá jednoznačne rozložiť na súčin ireducibilných polynómov s takýmito koeficientami. Vďaka Hilbertovej vete takýto rozklad existuje aj pre pre polynómy s viacerými neznámymi. Pre takéto typy polynómov sa v základnom kurze pre študentov odborného a učiteľského štúdia matematiky neuvádzajú žiadne metódy na faktorizáciu, hoci aj veľmi jednoduchých polynómov (ak nepočítame niekoľko bežných vzorcov typu $x^3 + y^3 = (x + y)(x^2 - xy + y^2)$).

Faktorizácia polynómov je veľmi dôležitý pomocník (podproblém) pri riešení viacerých úloh, ako napr. pri zjednodušovaní zložitých výrazov, rozklade na parciálne zlomky, hľadaní riešení rovníc. Menej známe je to, že rozklad polynómov nad $Q(\alpha)$, kde α je algebraické číslo nad Q, je dôležitá súčasť algoritmu na symbolické integrovanie.

Cieľom tohto príspevku je poskytnúť základný prehľad o algoritmoch používaných v tejto oblasti. Kvôli obmedzenému rozsahu sa sústredíme len na polynómy s jednou premennou. I keď žiadny z dôkazov, prezentovaných v tomto článku neprekračuje rozsah učebnice [3], nemožno ich označiť za elementárne. Pri čítaní je možné sa najprv oboznámiť s popisom algoritmov a potom si prípadne prečítať, prečo to vlastne všetko funguje.

Pred popisom jednotlivých algoritmov je zaujímavé zamyslieť sa nad možnými hranicami tohoto prístupu. Platí zaujímavá veta, ktorej zmysel bude asi zrejmý aj pre čitateľov, ktorí nie sú podrobne oboznámení s jednotlivými pojmami, ktoré sa v nej vyskytujú. (pozri napr. [2], [5]).

Veta 9 *Existuje rekurzívne spočítateľné pole K konečnej charakteristiky p (t.j. existujú algoritmy na výpočet súčtu, rozdielu, súčinu a podielu prvkov v K), také, že existuje algoritmus na faktorizáciu v okruhu polynómov $K[x]$ a ktoré má vlastnosť: existuje $\alpha \in K$ také, že pre $K(\alpha^{1/p})[x]$ neexistuje algoritmus na faktorizáciu.*

V celom nasledujúcom texte bude $\gcd(a, b)$ označovať najväčší spoločný deliteľ prvkov a, b (čo môžu byť celé čísla, alebo polynómy).

5 Rozklad polynómov z okruhu $Z[x]$: elementárne algoritmy

Na úvod uvedieme dva algoritmy založené na veľmi jednoduchej myšlienke deliteľnosti, v ktorých sa využívajú len bežné postupy. Treba ale povedať, že v oboch algoritmoch potrebujeme nájsť celočíslené delitele daného čísla. Toto samozrejme vôbec nie je ľahká úloha, stále sa jej venuje v matematickej literatúre veľká pozornosť. Uvedené algoritmy však ukazujú, ako pomocu programu dobre počítajúceho s celými číslami môžeme faktorizovať polynómy. V prípade polynómov malých stupňov s malými koeficientami môžeme použiť obyčajnú kalkulačku s nejakou formou Eratostenovho sita na nájdenie rozkladu.

5.1 Kroneckerov algoritmus

Kroneckerov algoritmus je asi prvý algoritmus, ktorý rieši problém faktorizácie polynómov. Je to jediný algoritmus, ktorý možno nájsť v bežne dostupnej lite-

ratúre (pozri [10]). Keďže je pomerne neefektívny, uvedieme len jeho neformálny popis.

Ak $f(x) = g(x)h(x)$, pričom všetky polynómy f, g, h majú celočíselné koeficienty, potom pre ľubovoľné celé číslo k platí $g(k)|f(k)$, $h(k)|f(k)$.

Ak je $st(f) = n$, tak aspoň jeden z polynómov g, h má stupeň $\leq [n/2]$. Keď chceme zistiť, či existuje polynóm stupňa $l \leq [n/2]$, ktorý je deliteľom polynómu f, môžeme postupovať nasledovným spôsobom: Zvolíme si $l+1$ navzájom rôznych celých čísiel $n_0, \ldots, n_l$. Zvolíme nejaký systém deliteľov $g_0, \ldots, g_l$ čísiel $f(n_0), \ldots, f(n_l)$, t.j. $g_i|f(n_i)$.

Pomocou nejakej interpolačnej metódy (Newton, Lagrange) skonštruujeme polynóm g taký, že $g(n_0) = g_0, \ldots, g(n_l) = g_l$. Ak má skonštruovaný polynóm g celočíselné koeficienty, je kandidátom na deliteľa polynómu f. Vydelením zistíme, či deliteľom je alebo nie je.

Ak nájdeme nejaký deliteľ, vydelíme ním pôvodný polynóm a potom postupujeme s hľadaním deliteľov dvoch polynómov nižšieho stupňa.

Keďže vieme prejsť všetky systémy deliteľov čísiel $f(n_0), \ldots, f(n_l)$, v konečnom počte krokov dostaneme rozklad polynómu f na ireducibilné polynómy (alebo dokážeme, že f je sám ireducibilný).

Uvedený algoritmus sa určite dá vylepšiť mnohými spôsobmi, ale vždy bude veľmi neefektívny.

5.2 Algoritmus pomocou interpolácie v jednom bode

Nech $f(x) = a_n x^n + \cdots + a_1 x + a_0$ je polynóm s celočíselnými koeficientami (vedúci koeficient a_n budeme vždy považovať za nenulový). Označme $|f| = \max\{|a_i|\,;\, i = 0, \ldots, n\}$. Kľúčové poznatky pre tento algoritmus sú skryté v nasledujúcich troch tvrdeniach:

Veta 10 *Nech $f(x) = a_n x^n + \cdots + a_1 x + a_0 \in C[x]$. Nech $\alpha \in C$ je koreň $f(x)$. Potom $|\alpha| < \frac{|f|}{|a_n|} + 1$.*

Dôkaz tejto vety je získame jednoduchým použitím vzorca na súčet geometrickej postupnosti.

Dôsledok 11 *Nech $0 < A \in N$, $m \in N$ a $2A + 1 \leq k \in N$. Potom existuje najviac jeden polynóm $f(x) = a_n x^n + \cdots + a_1 x + a_0 \in Z[x]$ taký, že $|f| \leq A$ a $f(k) = m$.*

Dôkaz. Koeficienty hľadaného polynómu (ak existuje) dostaneme postupným delením čísla m číslom k, pričom zvyšky (koeficienty počnúc a_0) budeme brať v intervale $\langle -A, A\rangle$. Za chvíľu uvedieme presný algoritmus.

Jednoznačnosť: nech f, $g \in Z[x]$ majú požadovanú vlastnosť. Potom $||f - g|| \leq 2A$. Ak je $f(x) - g(x) = b_l x^l + \cdots + b_1 x + b_0 \neq 0$, tak pre celé číslo b_l platí $|b_l| \geq 1$ a preto $2A/|b_l| + 1 \leq 2A + 1$. Podľa predošlej vety polynóm $f - g$ teda nemôže mať koreň k (lebo $k \geq 2A + 1$), iba ak je identicky rovný nule. Čiže $f = g$. □

Tento dôsledok hovorí o istom zvláštnom tvare g-adického vyjadrenia čísla m, ale tiež o tom, že za istých okolností je polynóm jednoznačne určený jednou svojou hodnotou. Dokonca bod, v ktorom potrebujeme poznať hodnotu nezávisí od stupňa polynómu, ktorý chceme "zrekonštruovať". Ak poznáme horné hraničenie absolútnych hodnôt polynómu, je tento jednoznačne určený svojou

hodnotou v ľubovoľnom "dostatočne veľkom" bode. Nepotrebujeme $n+1$ hodnôt ako pri klasickej interpolácii.

Znova, nech $f(x) = a_n x^n + \cdots + a_1 x + a_0$ je polynóm s celočíselnými koeficientami (vedúci koeficient a_n budeme vždy považovať za nenulový). Označme $||f|| = \sqrt{a_0^2 + \cdots + a_n^2}$.

Veta 12 (Landau-Mignotteova nerovnosť) *Nech $f(x) = a_n x^n + \cdots + a_1 x + a_0$, $g(x) \in Z[x]$, nech $g|f$ v okruhu $Z[x]$. Potom pre koeficient g_i polynómu g stupňa k platí $|g_i| \leq \binom{k}{i} ||f||$.*

Toto tvrdenie poskytuje práve to, čo potrebujeme na použitie predošlého dôsledku pri hľadaní deliteľov daného polynómu, a to je horné ohraničenie možných koeficientov každého deliteľa. Pre podrobnosti a dôkaz pozri [4], [7], [8] a tiež [6].

Poznámka. Podstatne slabší odhad vieme získať použitím Vietových vzťahov, stačí si uvedomiť, že každý koreň polynómu g je koreňom polynómu f, použitím Vietových vzťahov a vety 10 dostaneme jednoduchý odhad v tvare $|g_i| \leq \binom{k}{i} A^i$, kde $A = |f| + 1$ (dokonca, ak f nie je normovaný, dá sa použiť $A = \frac{|f|}{|a_n|} + 1$.

Teraz popíšeme, ako bude pracovať algoritmus.

Rozlíšime dva prípady: 1. vedúci koeficient daného polynómu je rovný 1, t.j polynóm je normovaný a 2. polynóm nie je normovaný.

Algoritmus 1

Nech teda $f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$. Našou úlohou je nájsť "kandidátov" na delitele tohoto polynómu. Bez ujmy na všeobecnosti predpokladajme, že $f(x)$ nemá celočíselné korene (tie totiž vieme pomerne ľahko nájsť). To znamená, že pre stupeň d deliteľa platí $2 \leq d \leq \frac{n}{2}$. Deliteľ $g(x)$ (ak existuje) bude tiež normovaný polynóm a bude mať tvar $g(x) = x^d + b_{d-1}x^{d-1} + \cdots + b_1 x + b_0$, pričom pre každý koeficent b_i je splnená nerovnosť

$$|b_i| \leq \binom{d}{i} ||f||$$

Označme $A = \max\{\binom{d}{i} ||f||;\ i = 0, \ldots, d\}$ a zvoľme si nepárne číslo $k \geq 2A+1$. Spomedzi deliteľov x čísla $f(k)$ vyberme tie, pre ktoré platia nasledujúce zrejmé obmedzenia:

$$\begin{array}{rcccll} k^2 - Ak - A & \leq & x & \leq & k^2 + Ak + A & \text{alebo} \\ k^3 - Ak^2 - Ak - A & \leq & x & \leq & k^3 + Ak^2 + Ak + A & \text{alebo} \\ & & \vdots & & & \\ k^d - Ak^{d-1} - \cdots - Ak - A & \leq & x & \leq & k^d + Ak^{d-1} + \cdots + Ak + A & \end{array} \tag{4}$$

Takto dostaneme istý počet čísiel, deliteľov čísla $f(k)$, z ktorých budeme "rekonštruovať" polynómy $g(x)$, možné delitele pôvodného polynómu. Nech teda a je deliteľ čísla $f(k)$, ktorý vyhovuje jednému z uvedených obmedzení. Polynóm $g(x) = x^d + b_{d-1}x^{d-1} + \cdots + b_1 x + b_0$ (jeho koeficienty) skonštruujeme podľa algoritmu:

```
e:= a
for i:=0 while e<>0 do
b_i:= e mod_sym k
   e:=(e-b_i) div k
end_for
```

kde x mod_sym k je zvyšok pri delení čísla x číslom k z intervalu $\langle -(k-1)/2, (k-1)/2\rangle$ (tzv. symetrický zvyšok, preto je v označení _sym). Ak sú koeficienty z intervalu $\langle -A, A\rangle$, je to kandidát na deliteľa, ak nie, rovno ho "zahodíme". Pre nájdený (ak sme ho nezahodili) polynóm $g(x)$ zistíme, či delí $f(x)$, ak áno vydelíme a pokračujeme v rozklade dvoch vzniknutých deliteľov (keď preberáme čísla a od najmenšieho k najväčšiemu, stačí rozkladať podiel $f(x)/g(x)$). Vďaka predpokladu, že $f(x)$ nemá (racionálne) korene, netreba kontrolovať tie čísla a, pre ktoré platí obmedzenie $k - A \leq a \leq k + A$.

Príklad. Nech $f(x) = x^7 - x^5 + 6x^4 + x^3 - 3x^2 + 8x + 3$. Potom $||f|| = 13$. Podľa Landauovej-Mignotteovej nerovnosti máme $A = 3.13 = 39$, lebo $\binom{3}{1} = \binom{3}{2} = 3$ (delitele hľadáme do stupňa 3). Kvôli jednoduchosti, napriek tomu zvolíme $A = 15$ a $k = 33 > 2 \cdot 15 + 1$, vypočítame $f(33) = 42586456047 = 3 \cdot 269 \cdot 4409 \cdot 11969$. Fakt, že $f(x)$ nemá celočíselné korene pokladajme za overený.

Zaujímavé delitele (pozri nerovnosti (4)) sa nachádzajú v intervaloch $\langle 579, 1599\rangle$ (tu by vyšiel ako deliteľ kvadratický polynóm) alebo $\langle 19092, 52782\rangle$ (kubický polynóm). V prvom intervale je kandidát $3 \cdot 269 = 807$. V druhom intervale je tiež jediný kandidát, a to $3 \cdot 11969 = 35907$. Nasledujúca tabuľka ukazuje vypočítané polynómy:

číslo	stupeň	polynóm	$k-$adický zápis	delí $f(x)$
807	2	$x^2 - 9x + 15$	$807 = 33^2 - 9 \cdot 33 + 15$	nie
35907	3	$x^3 - x + 3$	$35907 = 33^3 - 33 + 3$	áno

Prvý polynóm nevyhovuje, lebo jeho absolútny člen 15 nedelí 3. Druhý polynóm delí $f(x)$, podiel je $x^4 + 3x + 1$. Z predošlého postupu vyplýva, že tento polynóm je ireducibilný (nemá racionálny koreň a teda ho nedelia polynómy 1. a tretieho stupňa a už vieme, že jediný (tu nie sme celkom korektní, lebo naša voľba A bola 15 namiesto správnych 39) kandidát druhého stupňa $x^2 - 9x + 15$ ho nedelí). Preto

$$x^7 - x^5 + 6x^4 + x^3 - 3x^2 + 8x + 3 = (x^3 - x + 3)(x^4 + 3x + 1)$$

je rozklad na ireducibilné polynómy.

Venujme teraz niekoľko slov druhému prípadu, keď polynóm, ktorý chceme faktorizovať nie je normovaný.

Nech $f(x) = a_n x^n + \cdots + a_1 x + a_0$, $a_n \neq \pm 1$. V princípe máme dve možnosti, čo budeme robiť. Prvá spočíva v nájdení všetkých (kladných) deliteľov čísla a_n a zopakovaní uvedeného algoritmu pre každý z týchto deliteľov, ktorý sa použije ako vedúci koeficient kandidátov na delitele. Pritom samozrejme treba vhodne poopraviť nerovnosti (4) (náhrada vedúceho koeficienta 1 príslušným deliteľom).

Druhá možnosť spočíva v nasledovnom "triku": (predpokladajme, že všetky koeficienty polynómu sú navzájom nesúdeliteľné, ak to tak nie je, vydelením

najväčším spoločným deliteľom koeficientov získame nový polynóm, ktorý už má túto vlastnosť).

Algoritmus 2

a) Vyrobme polynóm $g(x) = a_n f(x)$.

b) Budeme hľadať deliteľ $u(x)$ s vedúcim koeficientom a_n ($u(x)$ má byť deliteľ $g(x)$) podľa algoritmu 1 s nasledovnou modifikáciou: nerovnosti (4) zmeníme tak, že všetky obmedzenia budú mať vedúci koeficient a_n.

c) Nech $u(x) = u_l x^l + \cdots + u_1 x + u_0$ je nájdený deliteľ. Pripomeňme si, že $u_l = a_n$. Položme $c = \gcd\{u_0, \ldots, u_l\}$ (toto číslo sa bežne nazýva *obsah (content)* polynómu $u_l x^l + \cdots + u_1 x + u_0$ a polynóm

$$\overline{u}(x) = \frac{u_l}{c} x^l + \frac{u_{l-1}}{c} x^{l-1} + \cdots + \frac{u_1}{c} x + \frac{u_0}{c}$$

sa nazýva *primitívna časť* polynómu $u_l x^l + \cdots + u_1 x + u_0$. Nech ďalej $v(x) = g(x)/u(x)$ a nech d a $\overline{v}(x)$ sú po rade obsah a primitívna časť polynómu $v(x)$.

Polynómy $\overline{u}(x), \overline{v}(x)$ sú deliteľmi polynómu $f(x)$, presnejšie $\overline{u}(x) \cdot \overline{v}(x) = f(x)$. Totiž

$$a_n f(x) = u(x)v(x) = c \cdot \overline{u}(x) \cdot d \cdot \overline{v}(x) = (c \cdot d) \cdot \overline{u}(x)\overline{v}(x)$$

Keďže[1] $\overline{u} \cdot \overline{v}$ je primitívna časť polynómu $u \cdot v$ a podľa dohody zo začiatku týchto úvah je primitívna časť polynómu $g(x)$ polynóm $f(x)$, dostávame, že $\overline{u}(x) \cdot \overline{v}(x) = f(x)$.

Keď použijeme tento trik, stačí použiť túto modifikáciu algoritmu 1 len raz, lebo podobne ako v prípade 1, tiež vopred poznáme vedúci koeficient. Platíme za to tým, že sa zvýšia možné absolútne hodnoty koeficientov deliteľa a my musíme pracovať s väčšími číslami A a k.

6 Rozklad polynómov z okruhu $Z_p[x]$

V tejto časti sa budeme venovať dvom algoritmom pre faktorizáciu polynómov s koeficientami z poľa Z_p. Prvý z nich, Berlekampov, je plnohodnotný algoritmus na rozklad. Druhý, tzv. distinct degree algoritmus nerozloží vždy daný polynóm na ireducibilné faktory, poskytne iba čiastočný rozklad, na dokončenie treba použiť napr. Berlekampov algoritmus. Je ale jednoduchší a pre Berlekampov algoritmus je zaujímavé každé zníženie stupňa rozkladaného polynómu.

Uvedené algoritmy sa dajú použiť ako podúlohy aj v špeciálnych, modulárnych algoritmoch na faktorizáciu polynómov s celočíselnými koeficientami.

V celej tejto časti bude p prvočíslo. Bez újmy na všeobecnosti môžeme predpokladať, že polynóm $f(x)$, ktorý ideme faktorizovať je normovaný a budeme hľadať len normované ireducibilné faktory.

6.1 Berlekampov algoritmus

Základná myšlienka Berlekampovho algoritmu spočíva v "preložení" úlohy faktorizácie do jazyka lineárnej algebry, kde potom budeme riešiť systém lineárnych rovníc. Spomínaný preklad zabezpečujú nasledujúce dve vety.

[1] na dôkaz tohto tvrdenia sa dajú použiť podobné argumenty ako tie, ktoré sú použité pri dôkaze Eisensteinovho kritéria ireducibility polynómov (pozri napr. [3], článok 7.4)

Prvá z nich je vlastne tzv. čínska veta o zvyškoch. Budeme používať jej formuláciu pre okruh polynómov $F[x]$, ako ho spomíname v poznámke za vetou 3.

Pravidlá pre počítanie v Z_p a malá Fermatova veta umožňujú dokázať dôležitú vetu:

Veta 13 *Nech $(E, +, \cdot)$ je komutatívny okruh charakteristiky p. Takýto okruh je zároveň vektorový priestor nad Z_p. Nech T: $E \to E$ je zobrazenie dané predpisom $T(a) = a^p$. Potom*

1. *T je okruhový homomorfizmus*
2. *T je lineárne zobrazenie vektorového priestoru E*

Dôkaz.

1. pre sčítanie: $(a+b)^p = \sum_{i=0}^{p} \binom{p}{i} \times (a^{p-i}b^i) = a^p + b^p$, lebo binomické koeficienty $\binom{p}{1}, \ldots, \binom{p}{p-1}$ sú deliteľné číslom p. Vďaka tomu, že charakteristika nášho okruhu je p je teda každé $\binom{p}{i} \times (a^{p-i}b^i)$ pre $i = 1, \ldots, p-1$ nulové (tiež by sme mohli binomické koeficienty počítať priamo v Z_p).

   pre násobenie to triviálne vyplýva z komutatívnosti
2. linearita: ešte treba dokázať, že pre $a \in Z_p$, $u \in O$ je $(a \cdot u)^p = a \cdot u^p$. Toto vyplýva z malej Fermatovej vety, ktorá hovorí, že pre prvky $a \in Z_p$ platí $a^p = a$.

□

Príklad okruhu spĺňajúceho predpoklady tejto vety je $(Z_p[x], +, \cdot)$. Veta potom okrem iného tvrdí, že pre ľubovoľný polynóm $u(x) \in Z_p[x]$ platí rovnosť $u(x)^p = u(x^p)$.

Ukážeme si, ako preložiť úlohu faktorizácie na úlohu riešenia systému lineárnych rovníc. Nech $f(x) = f_1(x). \cdots .f_r(x) \in Z_p[x]$, je rozklad (nateraz pre nás neznámy) na ireducibilné polynómy. Budeme naviac predpokladať, že $f_1, \ldots, f_r$ sú navzájom rôzne - takýto polynóm $f(x)$ sa nazýva "bez štvorcov". Ak polynóm $g(x) \in Z_p[x]$ nie je "bez štvorcov", potom je buď f $p-$ta mocnina nejakého polynómu — čo vieme ľahko rozoznať pomocou lineárneho zobrazenia T, nenulový koeficient je len pri mocnine, ktorej exponent je deliteľný číslom p a vtedy budeme namiesto f priamo pracovať radšej s jeho $p-$tou odmocninou, ktorú získame tak, že vydelíme každý exponent číslom p, ak treba urobíme to viac krát — alebo polynóm $\frac{g(x)}{\gcd(g(x), \mathrm{D}\, g(x))}$ je bez štvorcov a obsahuje vo svojom rozklade niektoré z ireducibilných polynómov nachádzajúcich sa v rozklade $g(x)$, ale s exponentami 1 ($\mathrm{D}g(x)$ je formálna derivácia polynómu $g(x)$).

Úloha je nájsť polynómy $f_1, \ldots, f_r$. Nech $s_1, \ldots, s_r$ sú prvky zo Z_p. Z čínskej vety o zvyškoch vyplýva, že existuje práve jeden polynóm $u(x)$, $\mathrm{st}(u) < \mathrm{st}(f)$ taký, že

$$\begin{array}{rcll} u(x) & \equiv & s_1 & \mod f_1 \\ & \vdots & & \\ u(x) & \equiv & s_r & \mod f_r \end{array} \tag{5}$$

Uvedené kongruencie hovoria, že $f_i(x) \mid u(x) - s_i$. Keďže $\mathrm{st}(u(x)) < \mathrm{st}(f(x))$, je $\gcd(f(x), u(x) - s_i)$ vlastný deliteľ $f(x)$. Navyše, ak $s_i \neq s_j$, $f_i(x)$ nedelí $u(x) - s_j$ a teda f_i delí $\gcd(f(x), u(x) - s_i)$ a nedelí $\gcd(f(x), u(x) - s_j)$.

Pre takéto $u(x)$ pre každé prípustné i platí

$$u(x)^p \equiv s_i^p = s_i \equiv u(x) \qquad \mod f_i$$

t.j.

$$u(x)^p \equiv u(x) \qquad \mod f \tag{6}$$

Naopak, nech $u(x) \in Z_p[x]$ je taký, že $u(x)^p \equiv u(x) \mod f$, t.j. $u(x)^p - u(x) = u(x)(u(x) - 1).\cdots.(u(x) - (p-1))$ je deliteľný polynómom $f(x)$. Pre každý ireducibilný faktor f_i existuje práve jeden prvok $s_i \in Z_p$ taký, že $f_r \mid u(x) - s_i$. Teda existuje r prvkov $s_1, \ldots, s_r \in Z_p$ takých, že platia všetky kongrencie sústavy (5).

Čiže hľadať riešenie $u(x)$ sústavy (5) je ekvivalentné s hľadaním riešenia kongruencie (6). (pozri [1])

Nech $\mathrm{st}(f(x)) = n$. Veta 13 hovorí, že riešenia kongruencie (6) sú vlastné vektory lineárnej transformácie T pre vlastnú hodnotu 1, presnejšie, vektor koeficientov $(u_0, \ldots, u_{n-1})$ hľadaného polynómu $u(x)$ je riešenie lineárnej sústavy rovníc s maticou $Q - I$, kde

$$Q = \begin{pmatrix} q_{0,0} & q_{0,1} & \cdots & q_{0,n-1} \\ q_{1,0} & q_{1,1} & \cdots & q_{1,n-1} \\ \vdots & \vdots & & \vdots \\ q_{n-1,0} & q_{n-1,1} & \cdots & q_{n-1,n-1} \end{pmatrix}, \tag{7}$$

a prvky Q sú určené rovnicami $x^{pk} \equiv q_{n-1,k}x^{n-1} + \cdots + q_{1,k}x + q_{0,k} \mod f$, $k = 0, 1, \ldots, n-1$ (pozor, "čísla" určené v predošlej kongruencii idú do príslušného stĺpca). (T je teraz transformácia vektorového priestoru odvodeného od faktorového okruhu $U = Z_p[x]/(f)$.)

Poznámka: Vektorový priestor U nad Z_p môžeme stotožniť s množinou $\{g(x) \in Z_p[x];\ \mathrm{st}(g) < \mathrm{st}(f)\}$ a podobne každý vektorový priestor $U_i = Z_p[x]/(f_i)$ môžeme stotožniť s množinou $\{g(x) \in Z_p[x];\ \mathrm{st}(g) < \mathrm{st}(f_i)\}$.

Z čínskej vety o zvyškoch vyplýva, že zobrazenie $\phi : U \to U_1 \times \cdots \times U_r$ dané predpisom

$$g(x)\phi = (g(x) \mod f_1, \ldots, g(x) \mod f_r)$$

je bijekcia, overiť lineárnosť tohto zobrazenia tiež nie je problém. Zobrazenie ϕ je teda izomorfizmus vektorových priestorov $\phi : U \cong U_1 \times \cdots \times U_r$.

Označme $W = \{u(x) \in U;\ u(x)^p = u(x)\}$ a $W_i = \{u(x) \in U_i;\ u(x)^p = u(x)\}$. Rovnosť typu $u(x)^p = u(x)$ vo formulách definujúcich W a W_i je rovnosť v príslušnom okruhu U a U_i; v $Z_p[x]$ ju môžeme nahradiť kongruenciou $u(x)^p \equiv u(x) \mod f$, prípadne $u(x)^p \equiv u(x) \mod f_i$. Z vety 13 vyplýva, že W je podpriestor U a W_i je podpriestor U_i. Podrobná analýza dôkazu ekvivalencie medzi riešeniami sytému (5) a kongruencie (6) by viedla k dôkazu toho, že aj $\phi : W \cong W_1 \times \cdots \times W_r$ — je to dokonca okruhový izomorfizmus. Naviac, každý podpriestor W_i je (nad Z_p) jednorozmerný vektorový priestor, totiž okruh U_i je na základe ireducibility f_i pole. Rovnica $u(x)^p = u(x)$ je rovnica typu $y^p = y$

pre prvky tohto poľa a má v ňom teda najviac p koreňov. Pole Z_p je podpole U_i. Pomocou malej Fermatovej vety zistíme, že každý prvok zo Z_p je koreňom tejto rovnice. To znamená, že množina W_i má práve p prvkov a je to teda jednorozmerný podpriestor U_i.

Preto $\dim(W)$ = počet ireducibilných faktorov polynómu $f(x)$ (ktorý je podľa predpokladu "bez štvorcov"). Podľa predošlých úvah je W množina riešení systému rovníc s maticou $Q - I$. Ak je táto množina riešení jednorozmerný priestor, je polynóm $f(x)$ ireducibilný.

Berlekampov algoritmus môžeme popísať nasledovne:

Algoritmus 3

1) Zisti, či je $f(x)$ bez štvorcov. Ak nie, zisti, či je $p-$ta mocnina, ak áno, pracuj s $p-$tou odmocninou. Ak nie je $p-$ta mocnina, pracuj s $\frac{f}{\gcd(f,\mathrm{D}f)}$.
2) Vypočítaj maticu Q.
3) Nájdi bázu riešení sústavy $Q-I$. Jeden z prvkov bázy bude vždy $(1,0,\ldots,0)$ - prvky zo Z_p (konštantné polynómy) sú vždy riešením rovnice (6). Počet prvkov bázy je počet ireducibilných faktorov polynómu f.
4) Pre každý prvok $s \in Z_p$ vypočítaj $\gcd(f, u - s)$, kde u je nekonštantný polynóm zodpovedajúci netriviálnemu prvku bázy určenej v predošlom bode. Netriviálne výsledky sú vlastné delitele polynómu $f(x)$. Tento bod treba robiť postupne pre všetky prvky s a polynómy u. Ak nenájdeme úplný rozklad, pustíme algoritmus rekurzívne na rozumný zoznam vypočítaných vlastných deliteľov (musí to skončiť, lebo sa vždy zníži stupeň).
5) Ak pôvodný polynóm nebol bez štvorcov, pre každý ireducibilný faktor zisti najvyšší exponent, s ktorým sa ešte nachádza v pôvodnom polynóme.

Najdrahšia časť tohto algoritmu je bod 4, zaujímavý je tiež efektívny spôsob nájdenia matice Q. Analýze bodu 4 sa budeme viac venovať v pokračovaní tohto článku.

6.2 Distinct degree algoritmus

Druhý algoritmus na (čiastočnú) faktorizáciu polynómov z okruhu $Z_p[x]$ je založený na nasledujúcom zovšeobecnení známej vety:

Veta 14 *Polynóm $x^{p^r} - x \in Z_p[x]$ je práve súčin všetkých normovaných ireducibilných polynómov zo $Z_p[x]$, ktorých stupeň delí číslo r.*

Dôkaz. Toto tvrdenie plynie z faktu, že konečné pole, ktoré má p^i prvkov je izomorfné s rozkladovým poľom polynómu $x^{p^i} - x$ spojeného so základnými vlastnosťami viacnásobných algebraických rozšírení poľa.

Presnejšie, derivácia $\mathrm{D}(x^{p^r} - x) = -1$. Preto $\gcd(x^{p^r} - x, \mathrm{D}(x^{p^r} - x)) = 1$, polynóm $x^{p^r} - x$ je bez štvorcov.

Nech $f \in Z_p[x]$ je ireducibilný polynóm, ktorý delí $x^{p^r} - x$. Nech $\mathrm{st}(f) = d$. Označme F rozkladové pole polynómu $x^{p^r} - x$ (t.j. najmenšie pole, ktoré obsahuje všetky jeho korene). Vieme, že F má p^r prvkov. Z deliteľnosti $f \mid x^{p^r} - x$ vyplýva, že všetkých d koreňov polynómu f sa nachádza v F. Nech α je koreň f, potom stupeň rozšírenia $Z_p(\alpha)$ nad Z_p, t.j. $[Z_p(\alpha) : Z_p]$ je d (lebo f je minimálny polynóm prvku α). Už vieme, že $Z_p(\alpha)$ je podpole F a teda $[F : Z_p(\alpha)].[Z_p(\alpha) : Z_p] = [F : Z_p] = r$. Odtiaľ dostávame, že $d \mid r$.

Naopak, nech f je ireducibilný a $\mathrm{st}(f) = d \mid r$. Nech α je koreň f. Potom $0 \neq \alpha \in Z_p[x]/(f)$. Rád prvku α v grupe $(Z_p[x]/(f) - \{0\}, \cdot)$ je $p^d - 1$ a preto

$\alpha^{p^d} = \alpha$. Potom aj $\alpha^{p^r} = \alpha$, lebo $d \mid r$. Každý koreň f je koreňom $x^{p^r} - x$, t.j. $f \mid x^{p^r} - x$. $\square$
Pretože x, $x-1$, $\cdots$, $x-(p-1)$ sú práve všetky normované ireducibilné polynómy prvého stupňa, je $x^p - x = x(x-1).\cdots.(x-(p-1))$.

Nech f je bez štvorcov. Zaveďme nasledovné označenia: označme $f_1 = f$. Nech $u_1 = \gcd(f_1, x^{p^1} - x)$, $f_2 = \frac{f_1}{u_1}$, $u_2 = \gcd(f_2, x^{p^2} - x)$ a indukciou ďalej nech $f_{i+1} = \frac{f_i}{u_i}$ a $u_{i+1} = \gcd(f_{i+1}, x^{p^{i+1}} - x)$.

Dôsledok 15 *Polynóm u_1 je súčin všetkých lineárnych deliteľov f. Všeobecnejšie, polynóm u_i je práve súčin všetkých ireducibilných deliteľov polynómu f stupňa práve i.*

Dôsledok je vlastne algoritmus na získanie netriviálnych deliteľov u_1, u_2, ... polynómu $f(x)$. V prípade, že polynóm obsahuje pre každý stupeň najviac jeden ireducibilný faktor tohto stupňa, získame úplný rozklad (odtiaľ názov distinct degree). Zo stupňa polynómu u_i vieme hneď určiť počet ireducibilných faktorov stupňa i daného polynómu bez štvorcov.

6.3 Vylepšenia Berlekampovho algoritmu

Ešte si ukážeme možné urýchlenia bodu 4) Berlekampovho algoritmu.

Nech teda $f(x) \in Z_p[x]$ je polynóm bez štvorcov a $u(x) \in Z_p[x]$ je riešením kongruencie $u(x)^p \equiv u(x) \mod f$. Základná myšlienka je nájsť kandidátov typu $u(x) - s$, $s \in Z_p$, pre ktoré je $\gcd(u(x) - s, f(x))$ netriviálny. Úvahy v tejto časti pochádzajú od H. Zassenhausa. Pre dané $u(x)$ položme

$$\begin{aligned} S &= \{s \in Z_p;\ \gcd(u(x) - s, f(x)) \neq 1\} \\ m_u(x) &= \prod\{(x - s);\ s \in S\}. \end{aligned}$$

Platí tvrdenie

Veta 16 *Polynóm $m_u(x)$ je minimálny polynóm polynómu $u(x)$, t.j. $m_u(x)$ je normovaný polynóm najnižšieho stupňa s koeficientami zo Z_p, pre ktorý platí $m_u(u(x)) \equiv 0$ mod $f(x)$.*

Dôkaz. Predpokladajme, že existuje polynóm $m(x) \in Z_p[x]$ nižšieho stupňa, ako je $\mathrm{st}(m_u(x))$. Musí existovať $s \in S$ také, že $m(s) \neq 0$ a teda existujú $q(x) \in Z_p[x]$ a $0 \neq r \in Z_p$ také, že

$$m(x) = q(x)(x - s) + r. \tag{8}$$

Keďže $s \in S$, jeden z ireducibilných faktorov f_i polynómu $f(x)$ delí $u(x) - s$. Ale zároveň $m(u(x)) \equiv 0 \mod f(x)$ a preto f_i delí $m(u(x))$. Ak v rovnici (8) urobíme substitúciu $u(x)$ za x, zistíme, že f_i by malo deliť nenulovú konštantu r — spor. Fakt, že $m_u(u(x)) \equiv 0 \mod f(x)$ hneď vidieť. $\square$

Štandartná metóda na nájdenie minimálneho polynómu $m_u(x)$ je určenie $u^2(x) \mod f(x), u(x)^3 \mod f(x), \ldots, u(x)^{\mathrm{st}(f)} \mod f(x)$ ($1 \mod f(x) = 1$, $u(x) \mod f(x) = u(x)$) a hľadanie najmenšieho r, pre ktoré existujú koeficienty $m_0, \ldots, m_{r-1} \in Z_p$ také, že

$$m_{r-1}u(x)^{r-1} + \cdots + m_1 u(x) + m_0 \equiv 0 \mod f(x).$$

Toto prvé riešenie je hľadaný minimálny polynóm. Poznamenajme, že tento polynóm možno nájsť aj pomocou rezultantov.

Príklad. Nech $f(x) = x^6 + x^4 + x^2 + 1 \in Z_3[x]$. Tento polynóm je bez štvorcov. Matice Q a $Q - I$ z bodu 2) Berlekampovho algoritmu sú

$$Q = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \quad Q - I = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Napríklad $x^{3.5} = x^{15} \equiv 0 + (-1)x + 0x^2 + (-1)x^3 + 0x^4 + (-1)x^5 \mod f(x)$, teda prvky $0, -1, 0, -1, 0, -1$ sme napísali zhora dole do šiesteho stĺpca matice Q. Po úprave matice $Q - I$ na trojuholníkový tvar ostanú tri nenulové riadky

$$Q - I \sim \begin{pmatrix} 0 & 1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Bázou riešení tejto sústavy sú vektory $(1, 0, 0, 0, 0, 0)$, $(0, 1, 0, 1, 0, 0)$, $(0, 0, 0, 0, 1, 0)$. To znamená, že $f(x)$ má tri ireducibilné faktory. Druhý vektor je vlastne polynóm $u(x) = x + x^3$. Nájdime jeho minimálny polynóm. Keďže $u^2(x) \mod f(x) = x^4 - 1$, $\mathrm{st}(m_u(x)) = 3$ (lebo $1 \mod f(x) = 1$, $u(x) \mod f(x) = x + x^3$ a $u^2(x) \mod f(x) = x^4 - 1$ sú zrejme lineárne nezávislé $\mod f(x)$, majú totiž rôzne stupne). Preto $S = Z_p$ a polynóm $u(x)$ nám teda vygeneruje tri netriviálne nesúdeliteľné delitele, ktoré preto musia byť ireducibilné. Vypočítajme teda ireducibilné delitele polynómu $f(x)$. Sú to

$$\begin{aligned} f_1(x) &= \gcd(u, f) = x^2 + 1 \\ f_2(x) &= \gcd(u + 1, f) = x^2 + x - 1 \\ f_3(x) &= \gcd(u - 1, f) = x^2 - x - 1 \end{aligned}$$

Keby sme použili druhý netriviálny vektor $(0, 0, 0, 0, 1, 0)$, t.j. polynóm $v(x) = x^4$, zistili by sme, že $v^2(x) \mod f(x) = 1$ a teda $\mathrm{st}(m(x)) = 2$. Tento vektor by vygeneroval len dva delitele, jeden z nich by nebol ireducibilný. Vzhľadom na výsledok, ktorý nám vyšiel je vidieť, že distinct degree algoritmus by nám nedal žiadny netriviálny deliteľ polynómu $f(x)$.

Metóda hľadania množiny S je efektívna vtedy, keď je počet ireducibilných deliteľov malý v porovnaní s číslom p.

Iná metóda (Zassenhaus, Cantor), umožňujúca nájsť netriviálny deliteľ rýchlejšie, je založená na skutočnosti, že pre nepárne p platí:

$$x^p - x = x(x^{(p-1)/2} - 1)(x^{(p-1)/2} + 1)$$

a teda každé $u(x)$, ktoré je riešením kongruencie $u(x)^p \equiv u(x) \mod f(x)$ platí

$$u(x)^p - u(x) \equiv u(x)(u(x)^{(p-1)/2} - 1)(u(x)^{(p-1)/2} + 1) \mod f(x).$$

Zdá sa prirodzené očakávať, že takmer polovica netriviálnych spoločných deliteľov $u(x)$ a $f(x)$ sú už delitele $u(x)^{(p-1)/2} - 1$ a druhá polovica sú delitele

$u(x)^{(p-1)/2}+1$, lebo stupne týchto polynómov sú približne polovica stupňa polynómu $u(x)^p - u(x)$.

Skutočne, platí veta

Veta 17 *Nech $u(x) \in W$, $f(x) \in Z_p[x]$ je bez štvorcov a $f(x) = f_1(x) \cdot \ldots \cdot f_r(x)$ je jeho rozklad na ireducibilné delitele v $Z_p[x]$. Potom pravdepodobnosť, že $\gcd(u(x)^{(p-1)/2} - 1, f(x))$ je netriviálny, je*

$$1 - \left(\frac{p-1}{2p}\right)^r - \left(\frac{p+1}{2p}\right)^r, \tag{9}$$

teda je najmenej 4/9.

Dôkaz. Nech $\phi : W \cong W_1 \times \cdots \times W_r$ je izomorfizmus popísaný v poznámke ku Berlekampovmu algoritmu v prvej časti tohoto článku. Ak u je riešením kongruencie $u(x)^p \equiv u(x) \mod f(x)$, tak je to vektor z W a $u\phi = (s_1, \ldots, s_r)$ nazývame modulárnou reprezentáciou $u(x)$. Zvoľme teda nejaké $u \in W$. Nech $w(x) = \gcd(u(x)^{(p-1)/2} - 1, f(x))$. Potom $w(x)$ je netriviálny ak buď $w(x) \neq 1$ alebo $w(x) \neq f(x)$.

Nech f_i je deliteľ $w(x)$. Toto je ekvivalentné s tým, že $s_i^{(p-1)/2} = 1$. Táto rovnica ale má vo W_i (ktoré je ako pole izomorfné so Z_p) práve $\frac{p-1}{2}$ riešení a preto pravdepodobnosť toho, že f_i delí $w(x)$ je $\frac{p-1}{2p}$. Teda pravdepodobnosť toho, že $w(x) = f(x)$ je $(\frac{p-1}{2p})^r$. Podobne sa dokáže, že pravdepodobnosť toho, že $w(x) = 1$ je $(\frac{p+1}{2p})^r$. Preto celková hľadaná pravdepodobnosť toho, že $w(x)$ je netriviálny deliteľ $f(x)$ je určená rovnicou (9). Skutočnosť, že pre $r \geq 2$ a $p \geq 3$ platí nerovnosť

$$1 - \left(\frac{p-1}{2p}\right)^r - \left(\frac{p+1}{2p}\right)^r \geq 4/9$$

nie je ťažké overiť. □

Pre zložitosť pôvodného Berlekampovho algoritmu a Berlekampovho algoritmu s modifikáciu podľa poslednej vety (t.j. namiesto počítania jednotlivých $\gcd(u(x)-s, f(x))$, $s \in Z_p$ sa počíta $\gcd(u(x)^{(p-1)/2} - 1, f(x))$ pre istý počet náhodne vybratých $u(x) \in W$) platí

Veta 18 *1. Nech p je také "malé", aby sa zmestilo do jedného slova v počítači. Potom zložitosť pôvodného Berlekampovho algoritmu na výpočet ireducibilných deliteľov polynómu $f(x)$ stupňa n nad poľom Z_p je $O(r.p.n^2 + n^3)$ operácií v poli Z_p.*

2. Modifikovaný Berlekampov algoritmus na výpočet ireducibilných deliteľov polynómu $f(x)$ stupňa n nad poľom Z_p má zložitosť $O(r.n^2.log(p).log(r) + 2n^3)$ operácií nad poľom Z_p.

V oboch vzorcoch je r počet ireducibilných deliteľov polynómu $f(x)$, priemerne je to $\log(n)$. Zložitosť sčítania a násobenia nad poľom $GF(p^m)$ bude pre prípad, že sa p zmestí do jedného počítačového slova $O(m)$ respektíve $O(m^2)$, ak sa nezmestí, je to asi $O(m.\log(p))$ respektíve $O(m^2.\log(p)^2)$.

Poznamenajme, že všetky úvahy uvedené pre prípad faktorizácie nad poľom Z_p (Berlekampov algoritmus, modifikovaný Berlekampov algoritmus, distinct degree algoritmus) prejdú (viac-menej) bezo zmeny pre ľubovoľné konečné pole, okrem polí typu $GF(2^m)$, kde pri vete 17 treba namiesto rozkladu

$$x^p - x = x(x^{(p-1)/2} - 1)(x^{(p-1)/2} + 1)$$

použiť rozklad $x^{2^m} - x = Tr(x).(Tr(x)+1)$, kde

$$Tr(x) = x + x^2 + x^4 + \cdots + x^{2^{m-1}}.$$

Pravdepodobnosť bude $1 - (\frac{1}{2})^{r-1} \geq \frac{1}{2}$.

Všetky výpočty uvedené v oboch pokračovaniach tohoto článku boli realizované pomocou systému MuPAD ([9]) v operačnom systéme Linux.

7 Henselova lema

7.1 Vzťah medzi rozkladmi $\mod p$ a v celých číslach

V úvodnom článku sme uviedli efektívny Berlekampov algoritmus na faktorizáciu polynómov z okruhu $Z_p[x]$, p prvočíslo. Ako ho použiť na faktorizáciu polynómu s celočíselnými koeficientami? Samozrejme, ak je polynóm $f(x) = a_n x^n + \cdots + a_1 x + a_0$ ireducibilný $\mod p$ (t.j. polynóm $(a_n \mod p)x^n + \cdots + (a_0 \mod p)$ je ireducibilný v okruhu $Z_p[x]$, p pritom nesmie deliť vedúci koeficient), je $f(x)$ ireducibilný nad Z. Opačné tvrdenia ale neplatí, napr. polynóm x^2+1 je ireducibilný, ale nie je ireducibilný nad Z_p, akonáhle p má tvar $4k+1$, lebo pre takéto čísla je -1 vždy kvadratický zvyšok. Tento polynóm je ale ireducibilný $\mod p$ pre každé prvočíslo tvaru $4k+3$ a preto vieme dokázať jeho ireducibilitu aj nad Z.

Horšie je, že existujú (nad Z) ireducibilné polynómy, ktoré pre ľubovoľné prvočíslo p nie sú ireducibilné $\mod p$. Jednoduchý príklad je x^4+1, ktorý má vždy rozklad na dva polynómy stupňa 2 (a tie možno tiež nie sú ireducibilné). Polynómy s takouto vlastnosťou tvoria celé triedy. Navyše, polynómy tohoto typu sa vyskytujú pri manipulácii s algebraickými číslami.

Berlekampov algoritmus skombinovaný s Landau-Mignotteovou nerovnosťou umožňuje sformulovať nasledovný algoritmus na faktorizáciu $f(x) = a_n x^n + \cdots + a_1 x + a_0$ nad Z:

Algoritmus 4

1) Nájdime prvočíslo M dostatočne veľké, aby pre ľubovoľný možný deliteľ g polynómu $f(x)$ platilo $|g| < M/2$.

2) Rozložme $f \mod M$ (t.j. ako polynóm v $Z_M[x]$). Ak je f normovaný, vždy vyberme normované ireducibilné delitele. Napíšeme $f = f_1 \cdot \ldots \cdot f_k \mod M$.

3) Zoberme f_i ako polynóm s celočíslenými koeficientami, zistime, či delí f. Ak áno, je to ireducibilný deliteľ f, vynecháme ho zo zoznamu.

4) Ak po bode 3) ostali nejaké delitele $\mod M$, ktoré nie sú deliteľmi nad Z, urobme z týchto zvyšných polynómy tvaru $f_i f_j$ a zoberme ich ako polynómy s koeficientami z intervalu $\langle -(M-1)/2, (M-1)/2 \rangle \mod M$. Zistime, či takýto súčin delí f. Ak áno, je ireducibilný deliteľ f, f_i a f_j vynecháme zo zoznamu.

5) Ak ešte niečo ostalo, urobíme súčiny po troch, štyroch, ..., keď vzniknú nové ireducibilné faktory, príslušné irebucibilné faktory $f_{i_1}, \ldots, f_{i_l}$ vynecháme zo zoznamu (pokračujeme, kým nie je vyskúšaná každá kombinácia, alebo jej doplnok).

Problém je v tom, že často je M také veľké, že nemôžeme efektívne pracovať v $Z_M[x]$, pretože Berlekampov algoritmus je pre veľké prvočísla veľmi náročný. Naviac, ak máme podozrenie, že je polynóm ireducibilný, s malým p to zistíme dosť rýchlo.

V prípade, že M je príliš veľké, treba použiť iný postup, ktorý umožní z rozkladu modulo niekoľko prvočísiel zistiť delitele nad Z. Tu ale nie je vhodné používať čínsku vetu o zvyškoch. Asi hlavný dôvod je ten, že by sme museli použiť Berlekampov algoritmus pre príliš veľa polí a naviac, po rozklade vo viacerých poliach by sme nevedeli ktoré polynómy v jednotlivých rozkladoch zodpovedajú jednému deliteľu v $Z[x]$ a na zistenie tohoto by sme museli použiť CRT - Newtonovu interpoláciu pre príliš mnoho možností, podobne ako pri naivnom Kroneckerovm prístupe. Preto je vhodné použíť inú metódu, pri ktorej budeme používať len jedno prvočíslo a našťastie je taká metóda známa.

7.2 Henselova lema — lineárna verzia

Štartujúc z rozkladu získaného Berlekampovým algoritmom, budeme počítať rozklad $\mod p^k$. Pre jednoduchosť si postup naznačíme len pre normovaný polynóm f bez štvorcov nad Z, ktorý má rozklad $f = gh$, kde g, h sú nesúdeliteľné[2]. Nech teda $f = a_n x^n + \cdots + a_1 x + a_0$, položme $f_i = (a_n \mod p^i)x^n + \cdots + (a_0 \mod p^i)$, podobne zadefujeme g_i, h_i. (To je vlastne $f \mod p^i$, koeficienty berieme vždy z intervalu $\langle -(p^i-1)/2, (p^i-1)/2\rangle$.) Predpokladajme, že g_1, h_1 sú nesúdeliteľné v okruhu $Z_p[x]$.

Platí teda $f_i \equiv g_i h_i \mod p^i$, alebo $f_i = g_i h_i$ v okruhu $Z_{p^i}[x]$. Ako výsledok Berlekampovho algoritmu poznáme rozklad $f_1 = g_1 h_1$. Skúsme sa na toto pozrieť z opačnej strany. Vypočítajme g_i, h_i pre $i = 2, \ldots$

Zrejme $g_2 \equiv g_1 \mod p$ a preto $g_2 = g_1 + p\hat{g}_2$, kde $\hat{g}_2$ je polynóm s koeficientami v intervale $\langle -(p-1)/2, (p-1)/2\rangle$. Podobne pre h, f. Potom $f_2 \equiv g_2 h_2 \mod p^2$ sa stáva

$$f_1 + p\hat{f}_2 \equiv (g_1 + p\hat{g}_2)(h_1 + p\hat{h}_2) \qquad \mod p^2.$$

Keďže $f_1 \equiv g_1 h_1 \mod p^1$, môžeme poslednú kongruenciu napísať ako

$$\frac{f_1 - g_1 h_1}{p} + \hat{f}_2 \equiv \hat{g}_2 h_1 + \hat{h}_2 g_1 \qquad \mod p. \tag{10}$$

V tejto kongruencii poznáme ľavú stranu, pravá lineárne závisí na neznámych $\hat{g}_2$ a $\hat{h}_2$. Keďže g_1, h_1 sú nesúdeliteľné, existuje jediná dvojica $\hat{g}_2$, $\hat{h}_2$, ktorá je riešením poslednej kongruencie a má vlastnosť $\mathrm{st}(\hat{g}_2) < st(g_1)$ (nájdeme ju pomocou Euklidovho algoritmu aplikovaného na f_1 a g_1 v okruhu $Z_p[x]$). Keďže máme predpoklad, že polynóm f a teda aj všetky f_i, g_i, h_i sú normované, bude dokonca aj $\mathrm{st}(\hat{h}_2) < \mathrm{st}(h_1)$. Polynómy g_2, h_2 sú jednoznačne určené.

Ďalej, $g_3 \equiv g_2 \mod p^2$ a preto $g_3 = g_2 + p^2\hat{g}_3$, kde $\hat{g}_3$ je polynóm s koeficientami v intervale $\langle -(p-1)/2, (p-1)/2\rangle$. Podobne pre h, f. Potom $f_3 \equiv g_3 h_3 \mod p^3$ sa stáva

$$f_2 + p^2\hat{f}_3 \equiv (g_2 + p^2\hat{g}_3)(h_2 + p^2\hat{h}_3) \qquad \mod p^3.$$

[2] I keď to nie je nevyhnutné, pri rozklade v $Z_p[x]$ je vhodné použiť prvočíslo p také, aby $f(x) \mod p \in Z_p[x]$ bol bez štvorcov, aby sme mohli použiť priamo algoritmus na rozklad a nemuseli tento polynóm ešte upravovať, lebo Berlekampov algoritmus pracuje len vtedy, keď je $f(x) \mod p \in Z_p[x]$ bez štvorcov. To dosiahneme tak, že zoberieme p tak, aby nedelilo resultant $\mathrm{res}_x(f, Df)$ - pojem resultantu a jeho dôležité vlastnosti sú spomínané v časti 8. Z vlastností, ktoré sú tam spomínané sa dá nahliadnuť, že táto "nedeliteľnosť" je presne to je na "bezštvorcovosť" potrebné - ale nie je to celkom triviálne.

Tiež je treba voliť p tak, aby nedelilo vedúci koeficient polynómu $f(x)$, aby sa nám pri "prevode" do $Z_p[x]$ nezmenil stupeň. Toto pri normovanom polynóme nespôsobí problém, len chceme naznačiť možné "singularity".

Keďže $f_2 \equiv g_2 h_2 \mod p^2$, môžeme poslednú kongruenciu napísať ako

$$\frac{f_2 - g_2 h_2}{p^2} + \hat{f}_3 \equiv \hat{g}_3 h_2 + \hat{h}_3 g_2 \mod p.$$

Navyše, $g_2 \equiv g_1 \mod p$ a $h_2 \equiv h_1 \mod p$, a preto platí

$$\frac{f_2 - g_2 h_2}{p^2} + \hat{f}_3 \equiv \hat{g}_3 h_1 + \hat{h}_3 g_1 \mod p. \tag{11}$$

Podľa našich výpočtov musí v zlomku na ľavej strane kongruencií (10), (11) vychádzať celočíselné delenie bezo zvyšku. Opäť, v tejto kongruencii poznáme ľavú stranu, pravá lineárne závisí na neznámych $\hat{g}_3$ a $\hat{h}_3$. Keďže g_1, h_1 sú nesúdeliteľné, existuje jediná dvojica $\hat{g}_3$, $\hat{h}_3$, ktorá je riešením poslednej kongruencie a má vlastnosť $\mathrm{st}(\hat{g}_3) < st(g_1)$ (a tu tiež $\mathrm{st}(\hat{h}_3) < \mathrm{st}(h_1)$). Polynómy g_3, h_3 sú jednoznačne určené.

Tiež je dobré si všimnúť, že v oboch prípadoch (10) a (11) vlastne riešime kongruencie typu $xh_1 + yg_1 \equiv c \mod p$, pričom sa mení polynóm c na pravej strane takejto kongruencie. Ak sú h_1, g_1 nesúdeliteľné, takéto riešenie vždy existuje a dá sa ľahko vygenerovať z riešenia kongruencie $xh_1 + yg_1 \equiv 1 \mod p$.

Takýmto spôsobom pokračujeme, kým nie je p^k dosť veľké, aby $|g| < p^k/2$ a $|h| < p^k/2$ — použitie Landau-Mignotteovej nerovnosti, pozri diskusiu nižšie.

Predošlý postup pomerne dobre odôvodňuje nasledujúcu vetu:

Veta 19 (Henselova lema) *Nech p je prvočíslo, $f(x) \in Z[x]$ je normovaný polynóm. Nech g_1, $h_1 \in Z_p[x]$ sú nesúdeliteľné normované polynómy také, že $f(x) \equiv g_1 h_1 \mod p$ (v tejto a podobných kongruenciách sa budeme na všetky polynómy dívať ako na polynómy nad Z, koeficienty z príslušného intervalu typu $\langle -(p^i - 1)/2, (p^i - 1)/2 \rangle$). Potom existujú jednoznačne určené polynómy g_i, h_i, $i = 2, 3, \ldots$ také, že*

$$f(x) \equiv g_i h_i \mod p^i, \quad g_{i+1} \equiv g_i \mod p^i, \quad h_{i+1} \equiv h_i \mod p^i$$

Ako zistíme, či nesúdeliteľné delitele g_1, h_1 nájdené Berlekampovým algoritmom pre f_1 sú "obrazmi"nejakých deliteľov polynómu f? Predpokladajme, že $\mathrm{st}(g_1) \leq \mathrm{st}(h_1)$. Vzhľadom na uvedenú jednoznačnosť sa v momente, keď máme dosť veľké p^i — určené pomocou Landau-Mignotteovej nerovnosti pre polynóm g, ktorý má byť deliteľom polynómu $f(x)$ stupňa $\mathrm{st}(g_1)$, čiže pri stanovovaní veľkosti p^i využijeme stupeň $\mathrm{st}(g_1)$ (všetky polynómy g_i majú rovnaký stupeň) — stačí presvedčiť, či g_i pre takto určené i naozaj delí polynóm $f(x)$ nad Z. Ak áno, je to v poriadku, ak nie, uvedené g_1, h_1 negenerujú (nie sú obrazmi) deliteľov $f(x)$. Inak by sme totiž mali dve postupnosti uvedených vlastností začínajúce tou istou dvojicou g_1, h_1.

Na nájdenie ireducibilných deliteľov (normovaného) polynómu $f(x)$ bez štvorcov nad Z použijeme podobný postup, ako v algoritme 1. Bod 2) urobíme s nejakým prvočíslom p (nestaráme sa o ohraničenie koeficientov). V bodoch 3)-5) namiesto toho, aby sme zistili, či príslušný kandidát (jeden z f_i, súčin tvaru $f_i f_j$, ...) je deliteľom, pomocou Henselovej lemy zistíme, či je obrazom (či generuje) deliteľa polynómu $f(x)$ nad Z.

Pri faktorizácii nenormovaného polynómu (vedúci člen $a_n \neq 1$) použijeme postup analogický ako pri algoritme 2 v kapitole 5. Budeme pracovať len s polynómom $f(x) \in Z[x]$, ktorý je bez štvorcov, je primitívny, t.j. content polynómu

$f(x)$ je 1 a a_n je jeho koeficient pri najvyššej mocnine, p nedelí a_n. Položme $u(x) = a_n f(x)$, nech v je "normovaná" verzia $u \mod p$ v $Z_p[x]$ a $v = g_1 h_1$ je bezštvorcový rozklad v $Z_p[x]$ na normované polynómy (o voľbe p, aby to "vyšlo" sme hovorili vyššie). Potom $a_n v \equiv f \mod p$, $(a_n)^2 v \equiv u \mod p$ a vedúce koeficienty sú správne (nie len kongruentné $\mod p$). Tiež očividne platí $u \mod p \equiv (a_n g_1)(a_n h_1)$, u má vedúci koeficient a_n^2, $a_n g_1$ aj $a_n h_1$ majú vedúce koeficienty a_n ako polynómy nad $Z[x]$.

Henselovu lemu teraz použijeme tak, že ju budeme robiť pre $u(x)$ namiesto $f(x)$, vo vyššie použitých vzorcoch budeme stále udržiavať a_n ako vedúci koeficient vo všetkých g_i a h_i. Takto pomocou Henselovej lemy po vhodne veľa iteráciách buď zistíme, že $a_n g_1$ je obraz ($\mod p$) nejakého deliteľa $g(x)$ polynómu $u(x)$ a potom keď urobíme primitívnu časť $\overline{g}$, tak to bude deliteľ $f(x)$ (ktorého obraz $\mod p$ je (po "znormovaní") $g_1(x)$) — vtedy sme OK — alebo to nie je obraz deliteľa a vtedy túto dvojicu deliteľov g_1, h_1 normovaných deliteľov $v(x)$ v $Z_p[x]$ "zahodíme".

Príklad:

Poďme faktorizovať polynóm

$$g(x) = 4x^4 + 55x^3 - 207x^2 + 319x - 18$$

Tento polynóm je primitívny (jeho koeficienty sú nesúdeliteľné), ale nie je normovaný. Preto budeme namiesto neho faktorizovať polynóm

$$f(x) = 4g(x) = 16x^4 + 220x^3 - 828x^2 + 1276x - 72$$

```
> g:=4*x^4+55*x^3-207*x^2+319*x-18;
```

$$g := 4\,x^4 + 55\,x^3 - 207\,x^2 + 319\,x - 18$$

```
> f:=4*g;
> f_1:=mods(f,3);f_2:=mods(f,9); f_3:=mods(f,27);
> f_4:=mods(f,81);f_5:=mods(f,243);f_6:=mod s(f,3^6);
> f_7:=mods(f,3^7);f_8:=mods(f,3^8);
```

$$f := 16\,x^4 + 220\,x^3 - 828\,x^2 + 1276\,x - 72$$

$$f\_1 := x^4 + x^3 + x$$

$$f\_2 := -2\,x^4 + 4\,x^3 - 2\,x$$

$$f\_3 := -11\,x^4 + 4\,x^3 + 9\,x^2 + 7\,x + 9$$

$$f\_4 := 16\,x^4 - 23\,x^3 - 18\,x^2 - 20\,x + 9$$

$$f\_5 := 16\,x^4 - 23\,x^3 - 99\,x^2 + 61\,x - 72$$

$$f\_6 := 16\,x^4 + 220\,x^3 - 99\,x^2 - 182\,x - 72$$

$$f\_7 := 16\,x^4 + 220\,x^3 - 828\,x^2 - 911\,x - 72$$

$$f\_8 := 16\,x^4 + 220\,x^3 - 828\,x^2 + 1276\,x - 72$$

Pre účely našej demonštrácie musíme opraviť vedúce koeficienty polynómov f_2 a f_3, (pre f_1 to nie je potrebné) aby to bolo číslo 16, napr. pričítaním $f\_2 := f\_2 + 18 * x^4$ a $f\_3 := f\_3 + 27 * x^4$:

```
> f_2:=f_2+18*x^4; f_3:=f_3+27*x^4;
```

$$f\_2 := 16\,x^4 + 4\,x^3 - 2\,x$$
$$f\_3 := 16\,x^4 + 4\,x^3 + 9\,x^2 + 7\,x + 9$$

Pozrime sa na vhodné prvočísla, ktoré môžeme použiť pre Berlekampov algoritmus, sú to tie, ktoré nedelia resultant $f(x)$ a jeho derivácie (pre také prvočísla bude polynóm f modulo p bez štvorcov)

```
> r:=resultant(f,diff(f,x),x); ifactor(r);
```

$$r := -1340093793839546368$$
$$-\ (2)^{16}\ \ (7)\ \ (17)\ \ (61)^2\ \ (293)\ \ (397)^2$$

Čiže môžeme použiť napr. prvočíslo 3, vyfaktorizujme teda v $Z_3[x]$ — použitie Berlekampovho algoritmu.

```
> Factor(f) mod 3;
```

$$(x^2 + 2\,x + 2)\,x\,(x + 2)$$

Naozaj vidíme, že $f \mod 3$ je bez štvorcov. Predpokladajme (alebo) overme, že polynóm $f \in Z[x]$ nemá racionálne korene (t.j. f nemá deliteľa stupňa 1 nad $Z[x]$) a to znamená, že jediné čo má zmysel, je skúsiť, či sa nedá rozložiť na polynómy stupňa 2. Keďže v rozklade $f \mod 3$ máme jeden ireducibilný polynóm stupňa 2, jediná dvojica deliteľov, ktorú má zmysel "zdvíhať" do $Z[x]$ je

```
> g_1:=mods(x^2+2*x+2,3);h_1:=mods(x^2+2*x,3);
```

$$g\_1 := x^2 - x - 1$$
$$h\_1 := x^2 - x$$

Opäť opravme tieto polynómy tak, aby mali vedúce koeficienty 4 (4 je vedúci koeficient v pôvodnom polynóme g):

```
> g_1:=g_1+3*x^2;h_1:=h_1+3*x^2;
```

$$g\_1 := 4x^2 - x - 1$$
$$h\_1 := 4x^2 - x$$

V skutočnosti sme mali tieto polynómy vynásobiť číslom 4 a všetky koeficienty okrem vedúceho zmeniť pomocou symetrického modulu číslom 3, ale keďže $4 \equiv 1 \mod 3$, spôsob, ktorý sme pre náš prípad použili je OK.

Teraz budeme počítať $\frac{f\_1-g\_1*h\_1}{3} + \hat{f}_2$ modulo 3 v symetrickej reprezentácii (formula (10)), čo je ale to isté ako počítať $f\_2 - g\_1 * h\_1$ v symetrickej reprezentácii modulo 9 ($9 = 3^2$) a vydelené číslom 3:

```
> it2:=mods(simplify(expand(f_2-g_1*h_1)),9)/3;
```

$$it2 := x^3 - x + x^2$$

Čiže riešime rovnicu (neznáme sú $\hat{g}_2$ a $\hat{h}_2$)

$$x^3 - x + x^2 = g_1\hat{h}_2 + h_1\hat{g}_2$$

v symetrickej reprezentácii $Z_3[x]$, pričom $\hat{g}_2$ a $\hat{h}_2$ musia mať stupeň menší ako 2.

Všimnime si, že $1 = -g_1 + h_1$, preto

$$x^3 + x^2 - x = (-x^3 - x^2 + x)g_1 + (x^3 + x^2 - x)h_1$$

Stupeň $x^3 + x^2 - x$ je ale 3 a preto musíme napr. pri h_1 zobrať zvyšok pri delení polynómom g_1:

```
> q:=mods(quo(it2,g_1,x,'r'),3);mods(r,3);
```

$$q := x - 1$$

$$-x - 1$$

čiže

$$x^3 + x^2 - x = (-x^3 - x^2 + x)g_1 + (qg_1 + (-x - 1))h_1$$

a teda

$$x^3 + x^2 - x = (\underbrace{-x^3 - x^2 + x + qh_1}_{-x})g_1 + (-x - 1)h_1$$

čo je vidieť z výpočtu:

```
> quo(-it2+q*h_1,g_1,x,'r1');mods(r1,3);
```

$$-3$$

$$-x$$

Takže sme získali $\hat{h}_2 = -x$ a $\hat{g}_2 = -x - 1$ a "druhá iterácia" je

```
> g_2:=mods(g_1+3*(-x-1),9)+3*x^2;h_2:=mods(h_1+3*(-x),9)+3*x^2;
```

$$g\_2 := 4\,x^2 - 4\,x - 4$$

$$h\_2 := 4\,x^2 - 4\,x$$

Na získanie ďalšej iterácie najprv vypočítajme $f_3 - g_2 h_2$ v symetrickej reprezentácii mod 3^3, t.j. 27 vydelené 3^2 (formula 11):

```
> it3:=mods(simplify(expand(f_3-g_2*h_2)),3^3)/3^2;
```

$$it3 := x^3 + x^2 - x + 1$$

Čiže riešime rovnicu (neznáme sú $\hat{g}_3$ a $\hat{h}_3$)

$$x^3 + x^2 - x + 1 = g_2\hat{h}_3 + h_2\hat{g}_3$$

v $Z_3[x]$, resp.

$$x^3 + x^2 - x + 1 = g_1\hat{h_3} + h_1\hat{g_3}$$

v $Z_3[x]$. Vieme, že

$$x^3 + x^2 - x + 1 = (-x^3 - x^2 + x - 1)g_1 + (x^3 + x^2 - x + 1)h_1$$

Potrebujeme znížiť stupne, na získanie $\hat{g_3}$ vydeľme

```
> q:=mods(quo(it3,g_1,x,'r'),3);mods(r,3);
```

$$q := x - 1$$
$$-x$$

a na získanie $\hat{h_3}$ vydeľme

```
> quo(-it3+q*h_1,g_1,x,'r1');mods(r1,3);
```

$$-3$$
$$-1 - x$$

a získali sme $\hat{g_3} = -x$ a $\hat{h_3} = -x - 1$. Preto

```
> g_3:=mods(g_2+3^2*(-x),3^3);h_3:=mods(h_2+3^2*(-x-1),3^3);
```

$$g\_3 := 4\,x^2 - 13\,x - 4$$
$$h\_3 := 4\,x^2 - 13\,x - 9$$

Ďalšia iterácia

```
> it4:=mods(simplify(expand(f_4-g_3*h_3)),3^4)/3^3;
```

$$it4 := x^2 - x - 1$$

```
> q:=mods(quo(it4,g_1,x,'r'),3);mods(r,3);
```

$$q := 1$$
$$0$$

```
> quo(-it4+q*h_1,g_1,x,'r1');mods(r1,3);
```

$$0$$

$$1$$

a preto

```
> g_4:=mods(g_3+3^3*(0),3^4);h_4:=mods(h_3+3^3*(1),3^4);
```

$$g\_4 := 4\,x^2 - 13\,x - 4$$

$$h\_4 := 4\,x^2 - 13\,x + 18$$

Ďalej

```
> it5:=mods(simplify(expand(f_5-g_4*h_4)),3^5)/3^4;
```

$$it5 := x^3 - x^2$$

```
> q:=mods(quo(it5,g_1,x,'r'),3);mods(r,3);
```

$$q := x$$

$$x$$

```
> quo(-it5+q*h_1,g_1,x,'r1');mods(r1,3);
```

$$0$$

$$0$$

preto

```
> g_5:=mods(g_4+3^4*(x),3^5);h_5:=mods(h_4+3^4*(0),3^5);
```

$$g\_5 := 4\,x^2 + 68\,x - 4$$

$$h\_5 := 4\,x^2 - 13\,x + 18$$

```
> it6:=mods(simplify(expand(f_6-g_5*h_5)),3^6)/3^5;
```

$$it6 := 0$$

Keď je výsledok 0, budú aj opravy nulové a preto

```
> g_6:=mods(g_5+3^5*(0),3^6);h_6:=mods(h_5+3^5*(0),3^6);
```

$$g\_6 := 4\,x^2 + 68\,x - 4$$

$$h\_6 := 4\,x^2 - 13\,x + 18$$

```
> it7:=mods(simplify(expand(f_7-g_6*h_6)),3^7)/3^6;
```

$$it7 := 0$$

aj tu máme 0 a preto sú opravy znova nulové, t.j. aj

```
> g_7:=mods(g_6+3^6*(0),3^7);h_7:=mods(h_6+3^6*(0),3^7);
```

$$g\_7 := 4\,x^2 + 68\,x - 4$$
$$h\_7 := 4\,x^2 - 13\,x + 18$$

```
> it8:=mods(simplify(expand(f_8-g_7*h_7)),3^9)/3^8;
```

$$it8 := 0$$

a rovnako dopadla aj ďalšia iterácia, znovu nerobíme žiadnu opravu, $g\_7 = g\_8$ a tiež $h\_7 = h\_8$. Už sme "vyčerpali" všetky iterácie $f_1, \ldots, f_8$, skúsme, či sme už nedostali deliteľov

```
> simplify(expand(f_8-g_7*h_7));
```

$$0$$

Ako vidíme, mali sme štastie a našli sme deliteľov, $f = f\_8 = g\_7h\_7$, ale dokonca už aj $f = f\_8 = g\_5h\_5$, lebo od piatej iterácie sme nerobili žiadne opravy. Tu si ale treba uvedomiť, že síce $f\_8 = g\_5h\_5$ platí v $Z[x]$, ale v $Z[x]$ je $f\_5 \neq g\_5h\_5$, $f\_6 \neq g\_6h\_6$ a $f\_7 \neq g\_7h\_7$ (príslušné rovnice platia v $i-$tej iterácii vždy $\mod 3^i$).

Ešte je dobre si uvedomiť, že sme počítanie iterácií skončili asi predčasne, v skutočnosti sme ich mali robiť až po takú iteráciu, kedy $\binom{2}{1}||f|| < \frac{3^i}{2}$, lebo hľadáme delitele stupňa 2. A túto nerovnosť sme neoverovali.

Primitívna časť polynómu g_8 je $x^2 + 17x - 1$, h_8 je primitívny a preto primitívna časť polynómu h_8 je $4x^2 - 13x + 18$ a teda pôvodný polynóm g vieme rozložiť na súčin týchto primitívnych polynómov, t.j.

$$g(x) = (x^2 + 17x - 1)(4x^2 - 13x + 18)$$

je rozklad na ireducibilné delitele na $Z[x]$.

Pozorný čitateľ si možno všimol použitie Taylorovho rozvoja pre funkciu $F(g, h) = f - g \cdot h$ a skutočnoť, že sa vlastne snažíme hľadať postupné aproximácie (v duchu p-adickej aproximácie) koreňov g, h rovnice $F(g, h) = 0$ metódou veľmi podobnou Newtonovej interpolačnej metóde. Len v skratke,

$$F(g_k+p^k\hat{g}_{k+1}, h_k+p^k\hat{h}_{k+1}) = F(g_k, h_k)+F_g(g_k, h_k)p^k\hat{g}_{k+1}+F_h(g_k, h_k)p^k\hat{h}_{k+1}+E,$$

E je nula $\mod\ p^{k+1}$ (Taylorov rozvoj). Potom

$$-F(g_k, h_k) \equiv F_g(g_k, h_k)(p^k\hat{g}_{k+1}) + F_h(g_k, h_k)(p^k\hat{h}_{k+1}) \mod\ p^{k+1}.$$

Keď uvážime, že $F_g(g_k, h_k) \equiv F_g(g_1, h_1) \mod p$ a $F_h(g_k, h_k) \equiv F_h(g_1, h_1) \mod p$, vidíme, že špeciálnym prípadom tejto kongruencie sú (10), (11). Neznáme sú $\hat{g}_{k+1}$, $\hat{h}_{k+1}$, vďaka nesúdeliteľnosti koeficientov ich vypočítame Euklidovým algoritmom (iterácia).

Uvedenú všeobecnú metódu (prípadne jej jednoargumentovú verziu) možno použiť na riešenie viacerých úloh pri "zdvihu" riešení zo Z_p do Z.

Faktorizácia polynómov nad Z sa dá robiť aj pomocou tzv. LLL algoritmu (čo je algoritmus na hľadanie generujúcich množín s "krátkymi" vektormi v podgrupách grúp typu Z^n - také podgrupy sa niekedy nazývajú mriežky podľa ich "geometrickej interpretácie" v R^n). LLL = Lenstra, Lenstra, Lovasz

8 Faktorizácia polynómov nad poliami algebraických čísiel

V predošlej časti sme si ukázali, ako rozkladať polynómy nad $Z[x]$ a teda aj $Q[x]$. I keď toto je najbežnejší okruh, v ktorom bežne potrebujeme faktorizovať, algoritmy na symbolické integrovanie predpokladajú, že vieme faktorizovať polynómy nad poliami typu $Q[\alpha_1, \ldots, \alpha_n]$, kde $\alpha_1, \ldots, \alpha_n$ sú algebraické čísla nad Q. Algoritmov riešiacich túto úlohu je viacero, popíšeme len jeden z nich, tzv. Tragerov algoritmus.

Nech F je pole typu $Q[\alpha_1, \ldots, \alpha_n]$, kde $\alpha_1, \ldots, \alpha_n$ sú algebraické čísla nad Q. Nech $m(x)$ je ireducibilný polynóm z $F[x]$, potom faktorový okruh $F[x]/(m(x))$ je pole a ak je α koreň $m(x)$ (α uvažujeme z nejakého nadpoľa poľa F), potom $F[x]/(m(x)) \cong F(\alpha)$. Toto pole môžeme tiež stotožniť (pri zadefinovaní násobenia $\mod m(x)$) s množinou $\{f_{n-1}\alpha^{n-1} + \cdots + f_1\alpha + f_0;\ f_0, \ldots, f_{n-1} \in F \,\&\, n = \mathrm{st}(m(x))\}$.

Nech $m(x)$ je normovaný. Potom je to minimálny polynóm prvku α nad F, označme ostatné korene tohoto polynómu $\alpha_2, \alpha_3, \ldots, \alpha_n$. Tieto prvky nazývame konjugáty prvku α nad F. (Napr. $-\sqrt{2}$ je konjugát $\sqrt{2}$ nad Q, lebo oba sú korene nad Q ireducibilného polynómu $x^2 - 2$.) Nech $\beta \in F(\alpha)$ je $\beta = f_{n-1}\alpha^{n-1} + \cdots + f_1\alpha + f_0$, potom konjugáty ku β sú $\beta_2, \beta_3, \ldots, \beta_n$ určené rovnicami $\beta_i = f_{n-1}\alpha_i^{n-1} + \cdots + f_1\alpha_i + f_0$. Poznamenajme, že konjugácia indukuje izomorfizmy σ_i: $F(\alpha) \to F(\alpha_i)$, kde $\sigma_i(\beta) = \beta_i$.

Nech $m(x)$ je ireducibilný nad F, a $\alpha_1, \ldots, \alpha_n$ sú jeho korene (v nejakom nadpoli). Z teórie symetrických funkcií vyplýva, že prvok $\beta \in F(\alpha_1, \ldots, \alpha_n)$ možno vyjadriť pomocou symetrických funkcií v premenných $\alpha_1, \ldots, \alpha_n$. Tieto sú podľa Vietových vzorcov až na znamienko koeficienty minimálneho polynómu $m(x)$ a teda z poľa F. Platí teda nasledujúca veta, veľmi dôležitá pre počítanie s algebraickými číslami:

Veta 20 *Prvok $\beta \in F(\alpha_1, \ldots, \alpha_n)$ je z F práve vtedy, keď je invariantný vzhľadom na ľubovoľnú permutáciu prvkov $\alpha_1, \ldots, \alpha_n$.*

Nasledujúci postup je závislý na funkcii Norm definovanej vzorcom

$$\mathrm{Norm}(\beta) = \beta \cdot \beta_2 \cdot \ldots \cdot \beta_n,$$

t.j. súčin β a všetkých konjugátov. Je vidieť, že $\mathrm{Norm}(\beta)$ je invariantné vzhľadom na permutácie koreňov polynómu $m(x)$ a teda Norm: $F(\alpha) \to F$.

Toto je jediné miesto, kde nevystačíme s vedomosťami, získanými v štandartných kurzoch základnej algebry. Aby sme sa na funkciu Norm mohli pozrieť z trochu všeobecnejšieho hľadiska, musíme použiť pár poznatkov z teórie rezultantov (pozri napr. [10], hlava V). Rezultant polynómov p, q v premennej x označíme $\mathrm{res}_x(p(x), q(x))$. Ak $q(x)$ je normovaný, tak (až na znamienko) platí

$$\mathrm{res}_x(p, q) = \prod\{p(x);\ q(x) = 0\}$$

a teda ak máme β vyjadrené polynómom $b(\alpha)$, tak môžeme napísať (až na znamienko) $\mathrm{Norm}\,(\beta) = \mathrm{res}_x(b(x), m(x))$, b, m uvažujeme ako polynómy v premennej x. Totiž

$$\mathrm{res}_x(b, m) = \prod\{b(x);\ m(x) = 0\} = b(\alpha)b(\alpha_2)\ldots b(\alpha_n),$$

lebo $\alpha, \alpha_2, \ldots, \alpha_n$ sú práve všetky prvky x, pre ktoré je $m(x) = 0$. Použime tieto vzorce na rozšírenie definície funkcie Norm tak, aby zahŕňala polynómy s koeficientami z $F(\alpha)$: nech $p \in F(\alpha)[z]$. p uvažujme ako polynóm o dvoch premenných α a z, t.j. $p(\alpha, z)$. Nahradením x za α tiež môžeme používať $p(x, z)$. Položíme

$$\mathrm{Norm}\,(p) = \mathrm{res}_x(p(x, z), m(x)),$$

výsledkom je polynóm v $F[z]$. Podobne sa dá Norm rozšíriť pre polynómy o viacerých premenných nad $F(\alpha)$.

Fundamentálna vlastnosť funkcie Norm je $\mathrm{Norm}\,(a \cdot b) = \mathrm{Norm}\,(a) \cdot \mathrm{Norm}\,(b)$, tiež platí, že $a(z)$ delí $\mathrm{Norm}\,(a)$ v $F(\alpha)[z]$. Teda rozklad polynómu $a(z) \in F(\alpha)[z]$ má za dôsledok rozklad $\mathrm{Norm}\,(a)$ v $F[z]$. Tragerov algoritmus vlastne obracia tento postup, rozklad $\mathrm{Norm}\,(a)$ "zdvíha" do rozkladu $a(z)$ v $F(\alpha)[z]$. Tiež je vidieť, že ak $f(z) \in F[z]$, tak $\mathrm{Norm}\,(f) = \prod\{f(z);\ m(x) = 0\} = f(z)^n$, lebo $f(z)$ nezávisí od α a teda sa v tomto súčine pre každý konjugát berie stále polynóm $f(z)$.

Prvá úloha je rozoznať, či je $a(z)$ ireducibilný. Na to slúži nasledujúca veta.

Veta 21 *Nech $a(z) \in F(\alpha)[z]$ je ireducibilný nad $F(\alpha)$. Potom* $\mathrm{Norm}\,(a)$ *je mocnina polynómu ireducibilného nad F.*

Dôkaz. Nech $\mathrm{Norm}\,(a)$ nie je mocnina ireducibilného polynómu, nech $f, g \in F[z]$ sú nesúdeliteľné polynómy také, že $\mathrm{Norm}\,(a) = f(z).g(z)$ Vieme, že $a(z)|\mathrm{Norm}\,(a)$, vďaka ireducibilite $a(z)$ musí $a(z)|f(z)$ alebo $a(z)|g(z)$. Bez újmy na všeobecnosti, nech platí prvá možnosť. Potom $\mathrm{Norm}\,(a(z))|\mathrm{Norm}\,(f(z)) = f(z)^n$, takže sme dostali, že pre nesúdeliteľné polynómy f, g platí, že $f(z)g(z)|f(z)^n$, čo je nemožné. □

Z uvedenej vety plynie, že ak $a(z) \in F(\alpha)[z]$ má vlastnosť, že $\mathrm{Norm}\,(a)$ je bez štvorcov, tak $a(z)$ je ireducibilný práve vtedy, keď je $\mathrm{Norm}\,(a)$ ireducibilný. Podobne, ak $a(z)$ má v $F(\alpha)[z]$ rozklad na ireducibilné delitele

$$a(z) = a_1(z) \cdot \ldots \cdot a_k(z), \text{ tak} \tag{12}$$

$$\mathrm{Norm}\,(a) = \mathrm{Norm}\,(a_1) \cdot \ldots \cdot \mathrm{Norm}\,(a_k) \tag{13}$$

je rozklad na ireducibilné delitele $\mathrm{Norm}\,(a)$ nad F. Ak je $\mathrm{Norm}\,(a)$ bez štvorcov, tak nemôže byť pre $i \neq j$ $\mathrm{Norm}\,(a_i) = \mathrm{Norm}\,(a_j)$. Špeciálne sme získali jednoznačnú korešpondenciu medzi deliteľmi $a(z)$ nad $F(\alpha)$ a $\mathrm{Norm}\,(a)$ nad F. Opak popíšeme vo vete

Veta 22 *Nech $a(z) \in F(\alpha)[z]$ má vlastnosť, že* $\mathrm{Norm}(a)$ *je bez štvorcov. Nech $p_1(z), \ldots, p_k(z)$ je rozklad na ireducibilné delitele* $\mathrm{Norm}(a)$ *v $F[z]$. Potom*

$$a(z) \quad = \quad \prod\{\gcd(a(z), p_i(z));\ i = 1, \ldots, k\} \tag{14}$$

je rozklad na ireducibilné delitele v $F(\alpha)[z]$.

Dôkaz. Nech (12) je rozklad na $a(z)$ ireducibilné delitele v $F(\alpha)[z]$ a teda (13) je rozklad $\mathrm{Norm}(a)$. Pre každé i preto existuje j pre ktoré platí vzťah

$$p_i(z) \quad = \quad \mathrm{Norm}(a_j) \tag{15}$$

$\mathrm{Norm}(a)$ je bez štvorcov a preto sa nemôže stať, že $\mathrm{Norm}(a_j) = \mathrm{Norm}(a_l)$ pre $j \neq l$.

Dokážeme, že ak sú p_i a a_j zviazané vzťahom (15), tak

$$a_j(z) \quad = \quad \gcd(a(z), p_i(z)), \tag{16}$$

kde gcd sa počíta v $F(\alpha)[z]$. Vďaka (12) a (15) delí a_j aj $a(z)$ aj $p_i(z)$ v $F(\alpha)[z]$. Existencia "väčšieho" deliteľa je ekvivalentná existencii $l \neq j$ takého, že a_l v $F(\alpha)[z]$ delí aj $a(z)$ aj $p_i(z)$. Ale keď a_l delí p_i, tak $\mathrm{Norm}(a_l)$ delí $\mathrm{Norm}(p_i)$. Keďže $p_i(z) \in F[z]$, je

$$\mathrm{Norm}(p_i) = p_i(z)^n.$$

Keďže $\mathrm{Norm}(a_l)$ je ireducibilný v $F[z]$, porovnaním posledných dvoch faktov dostaneme rovnosti $\mathrm{Norm}(a_l) = p_i(z) = \mathrm{Norm}(a_i)$, čo je spor. □

Výsledky tejto časti závisia na skutočnosti, či pre daný $a(z) \in F(\alpha)[z]$ bez štvorcov je aj $\mathrm{Norm}(a)$ bez štvorcov. To samozrejme nemusí byť pravda. Pomôžeme si nasledujúcim trikom: nájdeme $s \in F$ také, že pre $b(z) = a(z + s.\alpha)$ je $\mathrm{Norm}(b)$ bez štvorcov. Použitím predošlého postupu nájdeme rozklad $b(z) = b_1(z) \cdot \ldots \cdot b_k(z)$ a ak položíme $a_i = b_i(z - s.\alpha)$, tak $a(z) = a_1(z) \cdot \ldots \cdot a_k(z)$ je rozklad $a(z)$.

Existencia takého s vyplýva z vety

Veta 23 *Nech $a(z) \in F(\alpha)[z]$ je bez štvorcov. Potom až na konečne veľa $s \in F$ je* $\mathrm{Norm}(a(z - s.\alpha))$ *bez štvorcov.*

Dôkaz. Nech je $\mathrm{Norm}(a(z)) = p(z) \in F[z]$, nech $p(z) = p_1^{k_1}(z) \cdot \ldots \cdot p_m^{k_m}(z)$ je rozklad v $F[z]$. Zoberme polynóm $c(z) = p_1 \cdot \ldots \cdot p_m$ ($c(z)$ vieme nájsť pomocou algoritmu, ale to nie je pre potreby dôkazu dôležité).

Vieme, že $a(z) | \mathrm{Norm}(a(z)) = p(z)$ a $a(z)$ je bez štvorcov a preto $a(z)$ musí deliť aj $c(z)$. Teda

$$a(z - s\alpha) | c(z - s\alpha)) \text{ a preto aj } \mathrm{Norm}(a(z - s\alpha)) | \mathrm{Norm}(c(z - s\alpha))$$

Keď teraz dokážeme, že až na konečný počet $s \in F$ je $\mathrm{Norm}(c(z - s\alpha))$ bez štvorcov, bude to znamenať, že aj $\mathrm{Norm}(a(z - s\alpha))$ je až na konečný počet $s \in F$ bez štvorcov.

Nech $c(z) = (z - \beta_1).\ldots.(z - \beta_k)$, pre $i \neq j$ je $\beta_i \neq \beta_j$, lebo $c(z)$ je bez štvorcov. Vieme. že

$$c(z - s\alpha) = (z - (s\alpha + \beta_1)).\ldots.(z - (s\alpha + \beta_k))$$

Ak $\alpha_1 = \alpha, \alpha_2, \ldots, \alpha_n$ sú všetky konjugáty algebraického prvku α nad F. Potom

$$\mathrm{Norm}\,(c(z - s\alpha)) = \prod_{i=1}^{n}(z - (s\alpha_i + \beta_1)).\ldots.(z - (s\alpha_i + \beta_k))$$

a $\mathrm{Norm}\,(c(z - s\alpha))$ má teda viacnásobné korene práve vtedy, keď pre dané s existujú také i, j, l, m, že $(s\alpha_i + \beta_j) = (s\alpha_l + \beta_m)$. Keďže sú $\beta_1, \ldots, \beta_k$ po dvoch rôzne, môžeme možné s vyjadriť v tvare $s = \frac{\beta_m - \beta_j}{\alpha_i - \alpha_l}$ a na to máme len konečne veľa možností. □

Veta sa v algoritme používa tak, že ak samotné $\mathrm{Norm}\,(a(x))$ nie je bez štvorcov, tak hľadáme najmenšie $n \in N$ také, že $\mathrm{Norm}\,(b(z))$ pre $b(z) = a(z + n.\alpha)$ je bez štvorcov.

Vypočítajme si jeden klasický príklad. Nech $a(z) = z^4 + 1 \in Q(\sqrt{2})[z]$. Žiaľ, $\mathrm{Norm}\,(a)$ nie je bez štvorcov. Zoberme napr. $b(z) = a(z + \sqrt{2})$. Tu už je $\mathrm{Norm}\,(b)$ bez štvorcov a

$$\begin{aligned}\mathrm{Norm}\,((z+x)^4 + 1) &= \mathrm{res}_x((z+x)^4 + 1, x^2 - 2) = \\ &= z^8 - 8z^6 + 26z^4 - 8z^2 + 25 = \\ &= (z^4 + 1)(z^4 - 8z^2 + 25).\end{aligned}$$

Ďalej

$$\begin{aligned}\gcd((z+\sqrt{2})^4 + 1, z^4 + 1) &= z^2 + \sqrt{2}z + 1, \\ \gcd((z+\sqrt{2})^4 + 1, z^4 - 8z^2 + 25) &= z^2 + 3\sqrt{2}z + 5.\end{aligned}$$

Čiže ak položíme $b_1(z) = z^2 + \sqrt{2}z + 1$, $b_2(z) = z^2 + 3\sqrt{2}z + 5$, tak $b(z) = b_1(z)b_2(z)$. Položme teraz $a_1(z) = b_1(z - \sqrt{2}) = z^2 - \sqrt{2}z + 1$ a $a_2(z) = b_2(z - \sqrt{2}) = z^2 + \sqrt{2}z + 1$. Zistili sme, že

$$z^4 + 1 = (z^2 - \sqrt{2}z + 1)(z^2 + \sqrt{2}z + 1)$$

je rozklad $a(z)$ na ireducibilné delitele nad $Q(\sqrt{2})$.

Ak chceme použiť Tragerov algoritmus pre prípad rozkladu polynómu nad viacnásobným rozšírením, je nutné použiť rekurziu. Uvedený postup totiž vždy "odbúra" jednu premennú, t.j. znižuje "zložitosť" poľa, v ktorom sa skutočne robí faktorizácia. Alebo môžeme použiť vetu, podľa ktorej je každé konečné rozšírenie $Q[\alpha_1, \ldots, \alpha_n]$ nad Q jednoduchým rozšírením $Q[\beta]$ pre vhodné β. Otázka však je, ako toto β efektívne nájsť. Nevýhodou Tragerovho algoritmu je, že i keď sa zjednoduší okruh, v ktorom sa faktorizuje, zvýši sa stupeň polynómu, ktorý treba faktorizovať. Hľadanie efektívnych algoritmov na faktorizáciu polynómov nad poľami algebraických čísiel je stále oblasť aktívneho výskumu. Dá sa použiť napríklad metóda založená na gröbnerových bázach.

Literatúra

[1] J. Davenport, Y. Siret, and E. Tournier, *Computer Algebra*, Academic Press, London San Diego New York Sydney Tokyo Toronto, 1988, 1993.

[2] A. Fröhlich and J. Shepherdson, *Effective procedures in field theory*, Philos. Trans. roy. Soc. London, 248 (1955), pp. 407–432.

[3] T. Katriňák and kol., *Algebra a teoretická aritmetika 1*, Univerzita Komenského, Bratislava, 1985, 1995.

[4] E. Landau, *Sur quelques théorèmes de M. Petrovich relatifs aux zéros des fonctions analytiques*, Bull. Soc. Math. France, 33 (1905), pp. 251–261.

[5] D. Lazard, *Commutative algebra and computer algebra*, in Computer Algebra, J. Calmet, ed., LNCS 144, Springer-Verlag, Berlin Heidelberg New York, 1982, pp. 40–48.

[6] ——, *On polynomial factorization*, in Computer Algebra, J. Calmet, ed., LNCS 144, Springer-Verlag, Berlin Heidelberg New York, 1982, pp. 126–134.

[7] M. Mignotte, *An inequality about factoring polynomials*, Math. Comp., 28 (1974), pp. 1153–1157.

[8] ——, *Some useful Bounds*, Computing Supplementum 4, Springer-Verlag, Wien New York, 1982, pp. 259–263.

[9] B. Fuchssteiner, *MuPAD, Multi Processing Algebra Data Tool*, Kluwer Academic Publishers; Boston Dordrecht London, 1992.

[10] B. L. van der Waerden, *Algebra I, II*, Springer-Verlag, Berlin Heilderberg New York, 1971, 1967. ruský preklad - Nauka, Moskva, 1979.