Nagy elemszámú véges testek keresése Conway-polinomok segítségével

Műszaki Alkalmazott Matematika Msc
Diplomamunka

Szerző:
Hasznosi Tóth Csongor

Témavezető:
Dr. Nagy Gábor Péter

Szegedi Tudományegyetem Bolyai Intézet

2020

Tartalomjegyzék

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Dr. Nagy Gábor Péternek, az érdekes témaajánlásért, észrevételeit, tanácsadását, türelmét, megértését és végül, de nem utolsó sorban a dolgozatom elkészítéséhez szükséges szakirodalom rendelkezésemre bocsátását.

Köszönet a családomnak, hogy az évek során mindig mellettem álltak és támogattak céljaim elérésében. Barátaimnak, akik motiváltak, bíztattak és segítettek ahol tudtak. Évfolyamtársaimnak, hogy barátként és szaktársként is támogattak illetve segítettek. Továbbá, az általános iskolás tanáromnak és gimnáziumi oktatóimnak, hogy erre az útra tereltek és bíztattak. Páromnak a türelmét, támogatását, bíztatását, és hogy motivált mikor szükség volt rá.

Valamint, szeretném megköszönni a Szegedi Tudományegyetem Természettudomány és Informatikai Karának Bolyai Intézetében dolgozó valamennyi oktatónak a szakmai és lelkiismeretes munkáját, amivel a hallgatók képzését, tanulását és versenyképes tudás megszerzését támogatják.

Bevezetés

A legtöbb kriptográfiai protokol a véges testekre alapul, vagyis megfelelő véges testek használata elengedhetetlen a szükséges biztonság biztosításához. De vajon mennyire kell "végesnek" lennie egy ilyen testnek?

Ma az internet világában, mikor mindenkinek szinte minden pár kattintásnyira van, mi sem lehet fontosabb, mint a személyes adataink illetve a vagyonunk megőrzése az illetéktelen kezektől, így természetes, hogy a legnagyobb biztonságot szeretnénk elérni. Ezt pedig, olyan emberi elképzelést meghaladó, méretű, azaz elemszámú, véges testekkel tudjuk elérni, hogy az ebben a testben kódolt biztonsági kód feltörésére szánt idő, ne lehessen emberi években megszámlálható és egyenesen közelítse a végtelent. Szerencsére segítségünkre vannak a számítógépek, melyek ezen véges testek keresését elősegítik, de még a legerősebb számítógépnek is hónapokba, ha nem évekbe telik találnia egy ilyen megfelelő véges testet.

A dolgozatom ezen véges testek Conway polinomok általi keresésére hívatott algoritmusok működésébe, és működésének hátterébe nyújt betekintést.

Mivel magyar nyelvű forrás ezzel kapcsolatban alig található, ezért angol nyelvű forrásokat, melyek közül erős gerincét képezi dolgozatomnak *Lenwood S. Heath és Nicholas A. Loehr: "New algorithms for generating Conway polynomials over finite fields"* tanulmánya, magyar nyelvre való lefordítását és összegyűjtését, saját példákkal kibővítve, igyekeztem egy színvonalas mester szakos diplomamunkává gyúrni.

A dolgozat első felében a szükséges alapfogalmak vannak összegyűjtve a Conway polinomok bevezetéséhez. A Conway polinomok *John Horton Conway* angol matematikus nevéhez fűződnek 1980-ból. (John ez év áprilisában hunyt el 82 éves korában.) A Conway polinomok egy speciális irreducibilis polinomosztályba tartoznak, melyek megfelelő eszközöket biztosítanak véges testek kereséséhez.

A dolgozatom második felében Heath és Loehr Conway polinom kereső algoritmusát mutatom be, és hasonlítom össze a már korábban is létező, viszont időigényesebb bruth force algoritmussal. Megfigyelhetjük, hogy mikor

és miért sokkal gazdaságosabb ez az új algoritmus, még ha egyes esetekben alul is marad a bruth force algoritmussal.

Majd végül ismertetem a véges testek kódolásban betöltött szerepét és, nem részletbemenően, csak hogy legyen valamilyen viszonyítási alapunk, szót ejtünk a biztonsági kódolásról.

1. Alapfogalmak

Minden $\mathbb{F}$ véges testet két paraméter határoz meg, a test p prím tényezője és az n dimenziója Z_p , modulo p, egészek felett. Az $\mathbb{F}$ testnek p^n eleme van, és izomorf az összes p^n elemű testtel. Az $\mathbb{F}$ testet $GF(p,n)$-nel vagy csak egyszerűen $GF(p^n)$-nel jelöljük, GF a *Galois test*et (angolul:Galois field) jelöli. $\mathbb{F}$ multiplikatív csoportját jelölje $\mathbb{F}^*$ ami ciklikus. $\mathbb{F}^*$ egy elemét *primitív elem*nek nevezzük, ha az generálja a multiplikatív csoportot. Vagyis, ha $\alpha \in \mathbb{F}^*$ egy primitív elem, akkor

$$1 = \alpha^0, \alpha, \alpha^2, \ldots, \alpha^{p^n-2} \tag{1}$$

$\mathbb{F}^*$ elemei. Szükségünk lesz a multiplikatív csoport *számosságára* ami $p^n - 1$, jelölje $M_{p,n} = p^n - 1$.

Legyen $\mathbb{Z}_p[x]$ egyváltozós gyűrű $\mathbb{Z}_p$ felett. Egy $\mathbb{Z}_p[x]$-beli f polinom *irreducibilis* ha $f = gh$ esetén vagy g vagy h konstans. Egy n-ed fokú f irreducibilis polinom *primitív*, ha valahány (vagy az összes) gyöke f-nek primitív eleme $GF(p^n)^*$-nak. $GF(p^n)$-nek megfelel a $\mathbb{Z}_p[x]/(f)$ maradékosztálygyűrű, ahol f egy n-ed fokú irreducibilis polinom. Továbbá, ha f primitív, $GF(p^n)^*$ elemeinek megadása f egy α gyökének segítségével is lehetséges. Egy $\gamma \in GF(p^n)^*$ elemhez, legyen egy $i \geq 0$ legkisebb egész értékű *indexe* γ-nak amire igaz, hogy $\alpha^i = \gamma$. Vagyis, egyértelműen minden $\beta \in GF(p^n)$ megadható mint α legfeljebb $n-1$-ed fokú polinomja:

$$\beta = \sum_{i=0}^{n-1} b_i \alpha^i. \tag{2}$$

Az indexek segítségével történő felírás a $GF(p^n)^*$-beli multiplikációt összeadásra cseréli, modulo $M_{p,n}$, mivel a polinomalak egyszerűbbé teszi $GF(p^n)$-ben az összeadást. Mint ahogy azt a *[Lidl, Niederreiter]* jegyzet 2.52-es példájában is láthattuk, egy index táblázat, ami megadja az indexalak és polinomalak közti összefüggést, lesz a kulcs kiegészítő adatstruktúrája $GF(p^n)$ általános aritmetikájának.

Nagyobb kihívás viszont az, ha már nem csak $\mathbb{Z}_p$ és $GF(p^n)$ testeket szeretnénk felírni. Figyeljük testek egy sorozatát $\mathbb{Z}_p \subset GF(p^{n_1}) \subset GF(p^{n_2})$, ahol $1 < n_1 < n_2$. Ekkor n_1 osztja n_2-t. Tegyük fel, hogy α_1 és α_2 primitív elemek $GF(p^{n_1})$-ben és $GF(p^{n_2})$-ben, ebben a sorrendben. Ebben az esetben $GF(p^{n_1})^*$ ciklikus csoport részcsoportja $GF(p^{n_2})^*$ ciklikus csoportnak, és α_2 legkisebb hatványa amely $GF(p^{n_1})$ γ generátorát adja az

$$\gamma = \alpha_2^{M_{p,n_2}/M_{p,n_1}} \tag{3}$$

Testek ezen sorozatának aritmetikája, különösképpen a szorzás, akkor a legkényelmesebb, ha $\gamma = \alpha_1$. Ha f_1 és f_2 a minimálpolinomai α_1-nek és α_2-nek, akkor egyszerűen adódik, hogy $\alpha_1 = \alpha_2^{M_{p,n_2}/M_{p,n_1}}$ esetén

$$f_2(x) | f_1(x^{M_{p,n_2}/M_{p,n_1}}). \tag{4}$$

Ezeket a megfigyeléseket a következőképpen tudjuk általánosítani. Tegyük fel, hogy minden $GF(p^{n'})$ résztestnek, amely részteste a $GF(p^n)$ véges testnek választunk egy primitív, irreducibilis, n'-ed fokú $f_{p,n'} \in \mathbb{Z}_p[x]$ polinomot. Ekkor, ha $n_1 | n_2$ és $n_2 | n$ esetén

$$f_{p,n_2}(x) | f_{p,n_1}(x^{M_{p,n_2}/M_{p,n_1}}) \tag{5}$$

kapjuk, akkor azt mondjuk, hogy a választott polinom *összeegyeztethető*.

Most már definiálhatjuk ezen összeegyeztethető polinomok egy adott gyűjteményét, az úgynevezett *Conway polinomok*at. A $GF(p^n)$-t adó Conway polinomokat $C_{p,n}$-nel jelöljük. Conway polinomok definiálásához be kell vezetnünk a *lexikografikus rendezését* ($<_{lex}$) a d-ed fokú $\mathbb{Z}_p[x]$-beli polinomoknak:

$$a_d x^d + a_{d-1} x^{d-1} + \cdots + a_1 x + a_0 <_{lex} b_d x^d + b_{d-1} x^{d-1} + \cdots + b_1 x + b_0 \tag{6}$$

ahol, van olyan i, ami $d \geq i > 0$, hogy

$$a_j = b_j \ \ \forall j > i \tag{7}$$

és

$$(-1)^{d-i} a_i < (-1)^{d-i} b_i, \tag{8}$$

ahol az elemek rendje $\mathbb{Z}_p$-ben

$$0 < 1 < \cdots < p-1. \tag{9}$$

Conway polinomok definíciójának alap esete $C_{p,1}(x) = x - \gamma$, ahol γ Z_p, az elemek rendje alapján, legkisebb primitív eleme. Általános esetekre nézve, válasszuk úgy $C_{p,n}$-t, hogy lexikografikusan a legkisebb egyváltozós, irreducibilis, primitív, n-ed fokú polinom legyen, amire igaz az, hogy minden $n' < n$ esetén $n'|n$, ekkor

$$C_{p,n}(x) | C_{p,n'}(x^{M_{p,n}/M_{p,n'}}) \text{is teljesül.} \tag{10}$$

Ezen definíció alapján kapjuk a táblázatban szereplő Conway polinomokat, ahol $p = 3$.

$C_{3,1}$	$x + 1$
$C_{3,2}$	$x^2 - x - 1$
$C_{3,3}$	$x^3 - x - 1$
$C_{3,4}$	$x^4 - x^3 - 1$
$C_{3,5}$	$x^5 - x + 1$
$C_{3,6}$	$x^6 - x^4 + x^2 - x - 1$

Meglehetősen természetes elvárás az, hogy a $C_{p,n}$ Conway polinom primitív legyen és összeegyeztethető, n megfelelő d osztója esetén, $C_{p,d}$ Conway polinommal, mivel ez az összeférhetőség teszi lehetővé az egyszerű áttérést

$GF(p^d)$-beli elemek és ezen elemek $GF(p^n)$-beli megadásai között. Habár, semmi *algebrai* szükségesség nem kívánja meg a Conway polinomtól, hogy minimálpolinom legyen, ennek csak az a kellemes következménye, hogy így p és n egyértelműen meghatározzák a hozzájuk tartozó $C_{p,n}$ Conway polinomot, és leegyszerűsíti ezen polinomok keresésére használt brute force algoritmusokat. Ezen algoritmusok alapja, az n-ed fokú, egyváltozós $Z_p[x]$-beli polinomok lexikografikus sorozatait végig pásztázva, mindegyiket leellenőrizni, hogy irreducibilis-e, primitív-e és összeegyeztethető-e "kisebb" Conway polinomokkal. Az első polinom, amit az algoritmussal kapunk lesz a $C_{p,n}$. Bizonyos p-re és n-re, sok összeegyeztethető polinom létezik, a brute force algoritmus viszonylag gyorsan lefut, más értékekre viszont igen csak időigényes lehet. Következésképpen, a már ismert Conway polinomok listájának bővítése, csak nagy erőfeszítések árán történhet. [Heath, Loehr]

2. Összeegyeztethető rendszerek ciklikus csoportokban

A Conway polinomok definíciója alapján, még az sem világos, hogy mindig létezik ilyen $C_{p,n}$. Ebben a fejezetben, felelevenítünk egy-néhány alap csoportelméleti fogalmat, amelyek segítségével bizonyítjuk a Conway polinomok és az összeegyeztethető polinomok általános létezését.

Megfigyelhető, hogy az összeegyeztethető polinomok létezése csupán attól függ, hogy a véges test multiplikatív csoportja ciklikus-e. Következésképpen, először egybegyűjtjük az összeegyeztethető elemek rendszerét véges ciklikus csoportokban és ezek részcsoportjaiban. A Conway polinomok létezése ezen elméletek speciális következménye lesz.

Vegyünk egy k pozitív egészet. Legyen C_k a k-ad rendű ciklikus csoport, multiplikatív felírásban. Bármely $\alpha \in C_k$ elemre, α elem rendjét jelölje $o(\alpha)$.

1. Lemma. *$\alpha \in C_k$ és i egész esetén, α^i rendje*

$$\frac{o(\alpha)}{lnko(i, o(\alpha))}. \tag{11}$$

2. Lemma. Legyen j és k egészekre igaz, hogy $j|k$. Legyen $f : C_k \to C_{k/j}$ függvény, $f(x) = x^j$.

Ekkor f egy szürjektív csoport homomorfizmus. Továbbá,ha minden $y \in C_{k/j}$, akkor pontosan j elem van, hogy $x \in C_k$ esetén igaz, hogy $f(x) = x^j = y$.

Most már precízen tudjuk megfogalmazni a ciklikus csoport elemeinek összeegyeztethetőségét. Legyen $div(k)$ k osztóinak halmaza. *C_k összeegyeztethető generátorainak rendszerét* egy parciális függvény adja

$$\Gamma : div(k) \to C_k, \tag{12}$$

$def(\Gamma) \subset div(k)$-n értelmezve, ahol $def(\Gamma)$ a Γ függvény értelmezési tartománya, amire a következő tulajdonságok teljesülnek:

1. a függvény értelmezve van az 1-ben, azaz $1 \in def(\Gamma)$;
2. ha $i \in def(\Gamma), \text{akkor } o(\Gamma(i)) = i;$ és
3. ha $i \in def(\Gamma)$ és $j|i,$ akkor $j \in def(\Gamma)$ és $\Gamma(i)^{i/j} = \Gamma(j)$.

A Γ' összeegyeztethető generátorok rendszere Γ egy kiterjesztése, ha $def(\Gamma) \subset def(\Gamma')$, és ha $\Gamma'(i) = \Gamma(i)$, mindannyiszor ahányszor $i \in def(\Gamma)$. Ha $div(k) = def(\Gamma)$, akkor Γ *összeegyeztethető generátorok teljes rendszere.*

A kulcs eredményeként az összeegyeztethető generátorok rendszerének kapjuk, hogy minden parciális rendszer kiterjeszthető teljes rendszerré.

1. Tétel. *Tegyük fel, hogy Γ összeegyeztethető generátorok egy rendszere C_k-n. Akkor létezik egy Γ' összeegyeztethető generátorok teljes rendszere C_k-n, ami kiterjeszti Γ-t.* [Heath, Loehr]

//Nézzünk ezekre pár pédát:

1. Példa. Legyen $k = 20$. Ekkor:

$$div(20) = \{1, 2, 4, 5, 10, 20\}$$

C_k , multiplikatív jelöléssel, pedig

$$C_{20} = \{1, g, g^2, \ldots, g^{19}\}. \tag{13}$$

Keressük C_{20} generátor elemeit.

Nézzük is!

$\Gamma(1) = 1 \;\Rightarrow\; o(1) = 1$, ez egyértelműen következik a definícióból. De most nézzük az $i = 4$-re az $o(g^j) = i$-t. Benne lesznek C_{20} negyedrendű elemei $def(\Gamma)$-ban?

Vagyis keressük a 4-ed rendű elemeit C_{20}-nak.

$$\begin{aligned} &(g^j)^i = 1 \text{ alakban kerressük } j\text{-t } (j = 0, 1, \ldots, 19) \\ &(g^j)^4 = 1 = g^{20} \\ \Rightarrow &j = 5 \text{ és } j = 15 \end{aligned}$$

Tehát, azt kaptuk, hogy $o(g^5) = 4 = o(g^{15})$, ezek nem generálják C_{20}-t.

Táblázatba szedve C_{20} elemeit az elemek rendje alapján ezt kapjuk:

g^j	o( )	g^j	o( )	g^j	o( )	g^j	o( )
g^0	1	g^5	4	g^{10}	2	g^{15}	4
g^1	20	g^6	10	g^{11}	20	g^{16}	5
g^2	10	g^7	20	g^{12}	5	g^{17}	20
g^3	20	g^8	5	g^{13}	20	g^{18}	10
g^4	5	g^9	20	g^{14}	10	g^{19}	20

A táblázatból könnyen leolvasható, hogy melyek lesznek C_{20} generátor elemei:

$$gen(C_{20}) = \{g^i \in C_{20} |\; 2 \nmid i,\; 5 \nmid i \;\wedge\; lnko(i, 20) = 1\}.$$

Azaz $i = \{1, 3, 7, 9, 11, 13, 17, 19\}$, esetén lesz g^i generátor elem.

Hasonlóan $k = 28$-ra és $k = 42$-re:

2. Példa.

$$\begin{aligned} &k = 28 \\ &div(28) = \{1, 2, 4, 7, 14, 28\} \\ &C_{28} = \{1, g, g^2, \ldots, g^27\} \end{aligned} \tag{14}$$

g^j	o( )	g^j	o( )	g^j	o( )	g^j	o( )
g^0	1	g^7	4	g^{14}	2	g^{21}	4
g^1	28	g^8	7	g^{15}	28	g^{22}	14
g^2	14	g^9	28	g^{16}	7	g^{23}	28
g^3	28	g^{10}	14	g^{17}	28	g^{24}	7
g^4	7	g^{11}	28	g^{18}	14	g^{25}	28
g^5	28	g^{12}	7	g^{19}	28	g^{26}	14
g^6	14	g^{13}	28	g^{20}	7	g^{27}	28

Táblázat alapján:

$$gen(C_{28}) = \{g^i \in C_{28} |\ 2 \nmid i,\ 7 \nmid i\ \wedge\ lnko(i, 28) = 1\}.$$

Azaz $i = \{1, 3, 5, 9, 11, 13, 15, 17, 19, 23, 25\}$, esetén lesz g^i generátor elem.

3. Példa.

$$\begin{aligned} &k = 42 \\ &div(42) = \{1, 2, 3, 6, 7, 14, 21, 42\} \\ &C_{42} = \{1, g, g^2, \ldots, g^41\} \end{aligned} \tag{15}$$

g^j	o( )	g^j	o( )	g^j	o( )	g^j	o( )	g^j	o( )	g^j	o( )
g^0	1	g^7	6	g^{14}	3	g^{21}	2	g^{28}	3	g^{35}	6
g^1	42	g^8	21	g^{15}	14	g^{22}	21	g^{29}	42	g^{36}	7
g^2	21	g^9	14	g^{16}	21	g^{23}	42	g^{30}	7	g^{37}	42
g^3	14	g^{10}	21	g^{17}	42	g^{24}	7	g^{31}	42	g^{38}	21
g^4	21	g^{11}	42	g^{18}	7	g^{25}	42	g^{32}	21	g^{39}	14
g^5	42	g^{12}	7	g^{19}	42	g^{26}	21	g^{33}	14	g^{40}	21
g^6	7	g^{13}	42	g^{20}	21	g^{27}	14	g^{34}	21	g^{41}	42

Táblázat alapján:

$$gen(C_{42}) = \{g^i \in C_{42} |\ 2 \nmid i,\ 3 \nmid i,\ 7 \nmid i\ \wedge\ lnko(i, 42) = 1\}.$$

Azaz $i = \{1, 5, 11, 13, 15, 17, 19, 23, 25, 29, 31, 37, 41\}$, esetén lesz g^i generátor elem. //

Továbbá szeretnénk összeszámolni Γ összeegyeztethető generátorainak teljes rendszeré való kiterjesztéseit. Egy p prím és n egész esetén, legyen $\nu_p(n)$ p legnagyobb hatványa ami még osztja n-t, vagyis, $\nu_p(n) = p^e$, ahol $p^e|n$ és $p^{e+1} \nmid n$. Adott egy p prím és m , n egész számpár, úgyhogy $m|n$, definiáljuk m n-hez való *p-hozzájárulását*

$$\tau(m, n) = \begin{cases} \Phi(\nu_p(n)) & \text{ha } p \nmid m; \\ \nu_p(n)/\nu_p(m) & \text{ha } p|m. \end{cases} \tag{16}$$

Itt Φ az Euler Φ függvény, azaz $\Phi(p^e) = p^{e-1}(p-1)$ p prímre és $e \geq 1$.

Ha M n osztóinak halmaza, akkor M n-hez való p-hozzájárulása

$$\tau_p(M, n) = \min_{m \in M} \tau_p(m, n), \tag{17}$$

és definiáljuk M n-hez való *hozzájárulását*

$$\tau(M,n) = \prod_{p|n} \tau_p(M,n), \tag{18}$$

ahol p befutja n prímosztóit.

2. Tétel. *Ha* Γ *összeegyeztethető generátorok rendszere* C_k*-n, akkor* Γ *összeegyeztethető generátorok teljes rendszerré egészítéseinek száma* $\tau(def(\Gamma), k)$.

Alkalmazzuk az eddigieket a véges testekre. Rögzítsünk egy p prímet és egy n egészet. $GF(p^n)$ *primitív gyökeinek rendszerét* egy parciális függvénnyel kapjuk

$$\Psi : div(n) \to GF(p^n), \tag{19}$$

$def(\Psi) \subset div(n)$-n értelmezve, kielégítve a következőt: ha $i \in def(\Psi)$, akkor $o(\Psi(i)) = M_{p,i}$, vagyis, $\Psi(i)$ egy primitív eleme $GF(p^i)$-nek. Ψ' gyökeinek rendszere Ψ egy kiterjesztése, ha $def(\Psi) \subset def(\Psi')$, és ha $\Psi'(i) = \Psi(i)$, mindannyiszor $i \in def(\Psi)$.

Ha Ψ gyökök egy rendszere, akkor két gyök, $\Psi(i)$ és $\Psi(j)$, *összeegyeztethető*, ha az alábbiak közül valamelyik teljesül:

$$\begin{aligned} &1.i \text{ és } j \text{ nem osztják egymást,vagyis } i \nmid j \wedge j \nmid i; \\ &2. \text{ ha } i|j, \text{akkor} \Psi(j)^{M_{p,j}/M_{p,i}} = \Psi(i); \text{vagy} \\ &3. \text{ ha } j|i, \text{akkor } \Psi(i)^{M_{p,i}/M_{p,j}} = \Psi(j). \end{aligned} \tag{20}$$

Ha $\Psi(i)$ és $\Psi(j)$ összeegyeztethető minden i és j párra, akkor Ψ *összeegyeztethető gyökök rendszere.* Ha $div(n) = def(\Psi)$, akkor Ψ összeegyeztethető gyökök teljes rendszere.

3. Tétel. *Vegyünk egy* p *prímet és egy* n *egészet. Ekkor létezik* $GF(p^n)$*-nek összeegyeztethető gyökök teljes rendszere, ami* Ψ.

4. Tétel. *Vegyünk egy* p *prímet és egy* $n \geq 2$ *egészet. Tegyük fel, hogy* Ψ *összeegyeztethető gyökök rendszere* $GF(p^n)$ *összes résztestére. Le-*

gyen $M = M_{p,n} : 1 \leq i < n$ *és* $i|n$. *Ekkor* $GF(p^n)$ *primitív elemének, ami összeegyeztethető* Ψ*-vel, a száma* $\tau(M, M_{p,n})$.

5. Tétel. *A* $C_{p,n}$ *Conway polinom létezik minden* p *prímre és minden* n *egészre.* [Heath, Loehr]

3. Generáló algoritmusok

Az előző fejezet eredményeire alapozva bemutatok két Conway polinom generáló algoritmust. Viszonyítási alapnak elevenítsük fel a brute force algoritmust.

A következő jelölések végig fogják kísérni a fejezetet. Rögzítsünk egy p prímet és legyen n pozitív egész. Legyen n egy prímtényezős felírása $n = q_1^{e_1} \dots q_s^{e_s}$. Legyen minden $1 \leq i \leq s$ esetén $d_i = n/q_i$ és $m_i = M_{p,n}/M_{p,d_i}$, és végül legyen $g = lnko_{1 \leq i \leq s}\{m_i\}$ és $n_i = m_i/g$. [Heath, Loehr]

3.1. Kimerítő/Költségigényes keresés alapú algoritmus

Először tekintsük meg a bruth force algoritmust.

A legegyszerűbb bruth force algoritmus amit $C_{p,n}$ keresésére használunk azzal kezdődik, hogy keresni kezdjük $C_{p,d}$ polinomokat n minden d osztójára. Majd az algoritmus lexikografikusan sorba állítja Z_p összes n-ed fokú egyváltozós polinomját. Minden polinomra megvizsgáljuk a primitívségét és összeegyeztethetőségét $C_{p,d}$ polinomokkal. Az első polinom ami megfelel mind a kettő feltételnek, lesz a $C_{p,n}$

Jegyezzük meg, hogy az algoritmus keresés nagysága p^n, annak tekintetében, hogy p lehetséges együttható jut x minden fokár, 0 és $n-1$ között. Egy tételt mondunk ki a fejezet végén, miszerint az n-ed fokú egyváltozós polinomok száma amelyek összeegyeztethetők az alacsonyabb rendű Conway polinomokkal, pontosan g, sőt a 4.Tétel alapján, a lehetséges primitívek száma g összeegyeztethetők közül $\tau(M, M_{p,n})$. Semmi esetre sincs g-nél több pri-

mitív, összeegyeztethető polinom a keresésben. Ennélfogva, feltételezhetjük, hogy ezen polinomok véletlenszerűen (egyenletesen) oszlanak el a lexikografikus felsorolásban, azt várva, hogy a brute force algoritmus nagyjából p^n/g polinomot tesztel, mielőtt megtalálná az első feltételeknek megfelelőt. Ha n összetett és viszonylagosan nagy, mondjuk $n \geq 40$, akkor a $g \ll p^n$ és ekkor a brute force algoritmus egyáltalán nem praktikus.

A naiv algoritmust módosítsuk csupán azzal, hogy $C_{p,n}$ állandó tagját $(-1)^n\gamma$ alakban adjuk meg, ahol γ a $GF(p)$ legkisebb primitív eleme. Ez egyszerű következménye annak, hogy $C_{p,n}$ összeegyeztethető kell, hogy legyen $C_{p,1}$-gyel. Mivel, ha $C_{p,n}$ egy gyöke α, akkor az összeegyeztethetőség miatt $\alpha^{(p^{n-1})/(p-1)} = \gamma$. $C_{p,n}$ gyökei $\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{n-1}}$. $C_{p,n}$ állandó tagja $(-1)^n$-szerese a gyökök szorzatának, és ez a szorzat $\alpha^{1+p+\cdots+p^{n-1}} = \alpha^{(p^{n-1})/(p-1)} = \gamma$. Ugyanis így a keresés nagyságát egy p faktorral csökkentettük. Sajnos $C_{p,n}$-ben nincs ismert eljárás a magasabb rendű együtthatók megadására.

Egy másik javítása az algoritmusnak az alacsonyabb rendű polinomok összeegyeztethetőségi vizsgálatával történik. Emlékezzünk, $d_1, \ldots, d_s$ jelölte n osztóit. Tegyük fel, hogy $r(x)$ egy speciális polinom, ami összeegyeztethető minden $C_{p,d_i}(x)$ polinommal, vagyis

$$r(x)|C_{p,d_i}(x^{M_{p,n}/M_{p,d_i}}). \tag{21}$$

Legyen d bármelyik n osztói közül. Ekkor d osztója pár d_i-nek. Definíció alapján

$$C_{p,d_i}(x)|C_{p,d}(x^{M_{p,d_i}/M_{p,d}}),$$

és így

$$C_{p,d_i}(x^{M_{p,n}/M_{p,d_i}})|C_{p,d}([x^{M_{p,n}/M_{p,d_i}}]^{M_{p,d_i}/M_{p,d}}) = C_{p,d}(x^{M_{p,n}/M_{p,d}}).$$

Így ha (21) igaz n minden d_i maximális osztójára, akkor

$$r(x)|C_{p,d}(x^{M_{p,n}/M_{p,d_i}}) \tag{22}$$

teljesül n minden d osztójára is. Ennek köszönhetően polinomonként lecsökkentettük az összeegyeztethetőségi vizsgálatok mennyiségét. [Heath, Loehr]

3.2. Elemekre épülő algoritmus

A második fejezet eredményei szolgálnak ezen algoritmus alapjául. Ahhoz, hogy $C_{p,n}$ Conway polinomot megtaláljuk, először induktív módon ismernünk kell C_{p,d_i} Conway polinomokat, minden $1 \leq i \leq s$ esetén. Jegyezzük meg $GF(p^n)$ maximális alterén vett multiplikatív csoportok számossága $f_i = M_{p,d_i}$. Minden C_{p,d_i}-hez választunk egy x_i gyököt. Tudjuk, hogy $o(x_i) = f_i$. A $GF(p^n)^*$ multiplikatív csoportban tudunk olyan $x_{1,2}$ elemet találni aminek a rendje $lkkt\{f_1, f_2\}$ és összeegyeztethető x_1 és x_2-vel. A következő lépésben, tudunk olyan $x_{1,2,3}$ elemet találni aminek a rendje $lkkt\{f_1, f_2, f_3\}$ és összeegyeztethető x_1-gyel, x_2-vel és x_3-mal. Tovább iterálva, tudunk egy olyan $x_{1,2,\ldots,s}$ elemet találni aminek a rendje

$$f = lkkt\{f_1, f_2, \ldots, f_s\}$$

és összeegyeztethető $x_1, x_2, \ldots, x_s$-sel. Legyen $g = M_{p,n}/f$. Végül, minden g-edik gyöke $x_{1,2,\ldots,s}$-nek, ami primitív eleme $GF(p^n)$-nek, gyöke lehet $C_{p,n}$ Conway polinomnak.

Ennek az algoritmusnak a futási ideje szinte lineáris, attól függően, hogy mekkora g. [Heath, Loehr]

3.3. Polinomokra épülő algoritmus

Definíció alapján a Conway polinomokra teljesülnie kell

$$C_{p,n}(x) | C_{p,d}(x^{M_{p,n}/M_{p,d}}) \tag{23}$$

összeegyeztethetőségi feltételnek, n minden d osztójára. A (22) tranzitivitási egyenlet mellett, elegendő a (23) feltétel teljesülését ellenőrizni n maximális osztóira, azaz $d_1, \ldots, d_s$-re. Tisztán látszik, hogy az összeegyeztethetőségi feltétel teljesül ezekre az osztókra, akkor és csakis akkor, ha

$$C_{p,n}(x) |\ lnko_{1 \leq i \leq 1}\{C_{p,d_i}(x^{m_i})\}. \tag{24}$$

Így, ha ismerjük $C_{p,d_i}(x)$-t minden i-re, akkor megadhatjuk $C_{p,n}(x)$-t úgy, hogy egyszerűen kiszámoljuk a $C_{p,d_i}(x^{m_i})$ polinomok legnagyobb közös osz-

tóját, faktorizálva a kapott polinomot, és kiválasztjuk a lexikografikusan legkisebb primitív, irreducibilis, n-ed fokú faktort. Sajnos, a polinom foka

$$f(x) = lnko_{1\leq i\leq 1}\{C_{p,d_i}(x^{m_i})\} \tag{25}$$

legtöbbször n-hez viszonyítva nagyon nagy, így megnehezítve f faktorizálását.

Annak érdekében, hogy hatékony legyen az algoritmus, bevezetünk egy új ismeretlent $z = x^g$-t, ahol $g = lnko_{1\leq i\leq 1}\{m_i\}$. Vegyük észre, hogy minden $C_{p,d_i}(x^{m_i})$ polinom átírható $C_{p,d_i}(z^{m_i/g})$ alakba. Ennélfogva $f(x) = lnko_{1\leq i\leq 1}\{C_{p,d_i}(x^{m_i})\}$-t átírhatjuk $r(z)$ polinommá. Az így kapott polinom rendelkezi néhány hasznos tulajdonsággal, amit a következő tétel foglal magába.

A tétel bizonyításához szükségünk lesz a véges elemek elméletéből a következő lemmára:

3. Lemma. Ha f egy d-ed fokú irreducibilis polinom $GF(p)$ felett, akkor f minden gyöke $GF(p^n)$-beli. Ha α egy $GF(p^n)$-beli gyök, akkor f minden gyöke $\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{d-1}}$ és mind különbözőek.

Általánosan, ha $\alpha \in GF(p^d)$ nem szükségszerűen primitív, akkor az α^{p^i} elemeket, ahol $0 \leq i \leq d-1$, $GF(p^d)$-ben α *konjugáltjainak* nevezzük.

6. Tétel. *Legyen $n > 1$. A korábbi jelölést használva az*

$$r(z) = lnko_{1\leq i\leq s} C_{p,d_i}(z^{m_i/g})$$

polinom egy egyváltozós, irreducibilis, n-ed fokú polinom, továbbá s, n különböző prím osztóinak száma, legalább 2. Ha $s = 1$, akkor $r(z) = C_{p,n/q_i}(z)$.

Továbbá, ha $z_0 \in GF(p^n)$ valamelyik gyöke $r(z)$-nek, akkor z_0-nak pontosan g különböző g-edik gyöke van, $x_1, \ldots, x_g \in GF(p^n)$ és ezek a gyökök kielégítik a

$$C_{p,d_i}(x_j^{m_i}) = 0 \quad \textit{minden } i,j\textit{-re } (1 \leq i \leq s\ ;\ 1 \leq j \leq g)$$

összeegyeztethetőségi tulajdonságot. Ezen gyökök közül, az az egy amelyik primitív és minimálpolinoma lexikografikusan a legkisebb, az lesz a $C_{p,n}$ Conway polinom minimálpolinoma.

Ezen tétel bizonyítása alapozza meg a későbbi példáink megoldásának menetét.

Bizonyítás. Ha $s = 1$, egyértelmű, hogy $r(z) = C_{p,n/q_1}(z)$. Ezért tegyük fel, hogy $s > 1$, $r(z)$ pedig egyváltozós. Írjuk fel $r(z)$-t mint t darab $\mathbb{Z}_p$ feletti egyváltozós, nem-konstans, irreducibilis polinom szorzata

$$r(z) = f_1(z) \cdots f_t(z)$$

A tételből fakadóan tudjuk, hogy a $C_{p,n}(x)$ Conway polinomot $r(z) = r(x^g)$-nek osztania kell. Így, $r(z)$ nem konstans és $t > 0$.

Továbbá, mivel minden $C_{p,d_i}(x)$ irreducibilis tudjuk, hogy $lnko(C_{p,d_i}(x), \frac{d}{dx}C_{p,d_i}(x)) = 1$ minden i-re. $u = z^{m_i/g} = z^{n_i}$ választásával kapjuk, hogy

$$lnko\Big(C_{p,d_i}(z^{n_i}), \frac{d}{dz}C_{p,d_i}(z^{n_i})\Big) = lnko\Big(C_{p,d_i}(u), n_i z^{n_i-1}\frac{d}{du}C_{p,d_i}(u)\Big) = 1$$

Így $C_{p,d_i}(z^{n_i})$-nek nincs ismétlődő gyöke, bármely i-re, maga után vonva, hogy $r(z)$ sem tartalmazhat ismétlődő gyököket. Más szóval, $r(z)$ $f_i(z)$ faktorai páronként különbözőek.

Rögzítsük j-t úgy, hogy $1 \leq j \leq t$ és legyen $f_j(z)$ foka d. Legyen $p_j(z)$ egy gyöke z_j, $\mathbb{Z}_p$ kiterjesztett $GF(p^d)$ testében. Mivel $r_j(z) = 0$, ezért $C_{p,d_i}(z_j^{n_i})$=0. Így, z_j valamely hatványa, pontosabban $z_j^{n_i}$, $GF(p^{d_i})$-beli, ami alapján $GF(p^{d_i}) \subset GF(p^d)$ és ennélfogva $d_i \mid d$. Mivel $s \geq 2$, $n = lkkt(d_i) \mid d$. Ráadásul, mivel

$$z_j^{M_{p,n}} = (z_j^{m_i/g})^{gM_{p,d_i}} = 1^g = 1,$$

$z_j \in GF(p^n)$ és így $d|n$. Ennélfogva $d = n$, és $r(z)$ minden faktora n-ed fokú lesz.

Végül megmutatjuk, hogy $r(z)$ maga is irreducibilis, azaz $t = 1$. Mivel $r(z)$-nek nincsenek ismétlődő gyökei, elegendő csak azt megmutatni, hogy $r(z)$ bármely két gyöke egymás konjugáltja $GF(p^n)$-ben. Legyen z_0 és z_1 két tetszőleges gyöke $r(z)$-nek. C_{p,d_i} a $z_0^{n_i}$ és $z_1^{n_i}$ elemek minimálpolinoma, amiből következik, hogy $z_0^{n_i}$ konjugáltja $z_1^{n_i}$, $GF(p^{d_i})$-ben, azaz $z_1^{n_i} = (z_0^{n_i})^{p^{c_i}}$ ahol

$0 \leq c_i < d_i$. Azt mondjuk, hogy létezik c egész, amire $c \equiv c_i \pmod{d_i}$ minden i esetén. Ez a kiterjesztett *Kínai maradék tétel* következménye, feltéve, hogy $c \equiv c_i \pmod{n/q_iq_j}$, $1 \leq i < j \leq s$. Ahhoz, hogy igazoljuk ezen kongruenciákat, rögzítsük i-t és j-t, hogy teljesüljön $1 \leq i < j \leq s$, és definiáljuk $e = n/(q_iq_j)$-t. Továbbá definiáljuk $b_i = M_{p,d_i}/M_{p,e}$-t és $b_j = M_{p,d_j}/M_{p,e}$-t.

$$\begin{aligned} n_ib_i &= \frac{M_{p,n}}{gM_{p,d_i}}\frac{M_{p,d_i}}{M_{p,e}} \\ &= \frac{M_{p,n}}{gM_{p,d_j}}\frac{M_{p,d_j}}{M_{p,e}} \\ &= n_jb_j \end{aligned}$$

kiszámolva kapjuk, hogy

$$z_0^{n_ib_i} = z_0^{n_jb_j}$$

primitív eleme $GF(p^e)$-nek, mivel gyöke $C_{p,e}$-nek, akár csak

$$z_1^{n_ib_i} = z_1^{n_jb_j}.$$

Tovább számolva

$$\begin{aligned} (z_0^{n_ib_i})^{p^{c_i}} &= (z_0^{n_ip^{c_i}})^{b_i} \\ &= z_1^{n_ib_i} \\ &= z_1^{n_jb_j} \\ &= (z_0^{n_{ji}p^{c_j}})^{b_j} \\ &= (z_0^{n_jb_j})^{p^{c_j}} \\ &= (z_0^{n_ib_i})^{p^{c_j}}. \end{aligned}$$

Ennélfogva $c_i \equiv c_j \pmod{e}$-t igazoltuk.

Már van egy c egészünk amire $z_1^{n_i} = (z_0^{n_i})^{p^c}$ minden i esetén. Mivel $lnko\{n_1, \ldots, n_s\} = 1$, van a_i egész amire $\sum_{i=1}^{s} a_in_i = 1$. Ebből pedig kapjuk, hogy

$$\begin{aligned} \prod_{i=1}^{s}(z_1^{n_i})^{a^i} &= \prod_{i=1}^{s}(z_0^{n_ip^c})^{a^i} \\ z_1^{\sum_{i=1}^{s} a_in_i} &= z_0^{p^c\sum_{i=1}^{s} a_in_i}, \end{aligned}$$

következésképpen

$$z_1 = z_o^{p^c}.$$

Ennélfogva, z_0 és z_1 egymás konjugáltjai $GF(p^n)$-ben, vagyis $t = 1$. Ezzel kész a tétel első részének bizonyítása.

A bizonyítás hátralévő részében, tegyük fel, hogy $s \geq 1$- Legyen z_0 egy rögzített gyöke $r(z)$-nek $GF(p^n)$-ben. Mivel

$$z_0^{M_{p,n}/g} = (z_0^{m_1/g})^{M_{p,d_i}} = 1,$$

z_0 rendje $M_{p,n}/g$ osztója kell, hogy legyen. A második Lemma alapján, z_0-nak g különböző g-ed gyöke van $GF(p^n)$-ben. Ha x_j egy ilyen gyök, akkor minden i-re,

$$C_{p,d_i}(x_j^{m_i}) = C_{p,di}(z_0^{m_i/g}) = 0,$$

így a gyökök összeegyeztethetőek a korábban választott Conway polinommal. Megfordítva,$r(z)$ definíciója szerint, bármely összeegyeztethető testbeli x_0 elemnek van $m(x)$ minimálpolinoma amely osztója $r(z) = r(x^g)$-nek, ennélfogva, x_0 egyik g-edik gyök lesz z_0-nak. Ebből adódóan következik, hogy az egyik x_j gyöknek $C_{p,n}(x)$ a minimálpolinoma. A helyes gyöknek primitívnek kell lennie, amely minimálpolinoma lexikografikusan a legkisebb, ezen gyök létezését az 5. Tétel biztosítja. □

A tételből fakadóan, azonnal adódik a következő Conway polinom kereső algoritmus. n $q_1^{e_1} \cdots q_s^{e_s}$ prímtényezős felbontásával kezdjük. Az algoritmus három esetre bontható, n prímszorzatában lévő s különböző prímtényező alapján.

- ***I. Eset -*** $s \geq 2$

    1. Keressük ki (vagy számoljuk ki rekurzívan) a $C_{p,d_i}(x)$ Conway polinomot minden $d_i = n/q_i$ maximális valódi osztó esetén.
    2. Legyen $m_i = M_{p,n}/M_{p,d_i}$. Keressük meg $g = lnko_{1 \leq i \leq s}\{m_i\}$-t. z-t x^g-nek választva számoljuk ki

$$r(z) = lnko_{1 \leq i \leq s}\{C_{p,d_i}(z^{m_i/g})\}$$

bármely standard polinom legnagyobb közös osztóját kereső algoritmussal.

3. Legyen $r(z)$ egy gyöke z_0 $GF(p^n)$-ben. Keressük meg z_0 bármely $GF(p^n)$-beli g-edik α gyökét, és legyen ζ egy primitív g-edik gyöke $GF(p^n)$ egységének. Mivel $r(z)$ egy n-ed fokú irreducibilis polinom $\mathbb{Z}_p[z]$ felett (a hatos Tétel alapján), kényelmesen választás a test reprezentációs alakját felhasználni a gyökkereső algoritmusunkhoz

$$GF(p^n) = \mathbb{Z}_p[z]/(r(z)).$$

4. Vegyük észre, hogy a $GF(p^n)$-beli z_0 minden g-edik gyöke $\alpha\zeta^k$ alakú, $0 \leq k < g$. Vegyük sorra ezen gyököket. Minden primitív gyöknek adjuk meg a minimálpolinomát, és válasszuk ki közülük a lexikografikus értelemben legkisebbet. A négyes Tétel alapján, pontosan $\tau(M, M_{p,n})$ primitív g-edik gyöke van z_0-nak, és tudjuk, hogy $\tau(M, M_{p,n}) > 0$. Még általánosabban, bármely p értékre megkaphatjuk polinomok egy összeegyeztethető rendszerét úgy, hogy találunk z_0 g-edik gyökei között olyan primitív elemet, amely összeegyeztethető egy, már korábban kiválasztott, primitív elemmel. Ezt a későbbiekben még részletezem.

- ***II. Eset -*** $s = 1$

  Ez az eset egy degenerált formája az előzőnek. Vegyük $n = q_0^{e_1}$, $g = M_{p,n}/M_{p,n/q_1}$, és $r(Z) = C_{p,n/q_1}(z)$. Mint ahogy korábban is, z_0 legyen q bármely gyöke, amely $GF(p^n) \supset GF(p^{n/q_1})$. Így, pont mint korábban, végigvesszük mind a g gyököt, és kikeressük a lexikografikusan legkisebb n-ed fokú minimálpolinomát a megfelelő primitív gyöknek.

- ***III. Eset -*** $s = 0$

  Legyen $n = 1$. Ebben az esetben, $C_{p,1}(x)$-t keressük, a $\mathbb{Z}_p$ prímtest feletti Conway polinomot. Egyszerűen csak sorban leteszteljük $\mathbb{Z}_p$ minden elemének primitívségét, amíg nem találjuk γ-t, az első primitív

elemet. Ez polinom időben elvégezhető. Majd, a definíció alapján $C_{p,1}(x) = x - \gamma$.

A 6. Tétel bizonyítja az I. és II. esetben az algoritmus pontosságát, és egyértelműen megfelel a III. esetre is.

4. Példa. Tegyük fel, hogy meg szeretnénk keresni $C_{2,6}(x)$-t. Ekkor, $p = 2$, $n = 6$, $q_1 = 2$, $q_2 = 3$, $s = 2$, $d_1 = 3$, $d_2 = 2$, $m_1 = (2^6-1)/(2^3-1) = 9$, $m_1 = (2^6-1)/(2^2-1) = 21$, és $g = lnko(9, 21) = 3$. Kikeressük $C_{2,3}(x) = x^3 + x + 1$ és $C_{2,2}(x) = x^2 + x + 1$ polinomokat. $z = x^3$ választásával

$$C_{2,3}(x^9) = x^{27} + x^9 + 1 = z^9 + z^3 + 1$$
$$C_{2,2}(x^{21}) = x^{42} + x^{21} + 1 = z^{14} + z^7 + 1.$$

A két polinom legnagyobb közös osztója

$$f(x) = x^{18} + x^{15} + x^{12} + x^6 + 1 = z^6 + z^5 + z^4 + z^2 + 1 = r(z)$$

Ez egy z ismeretlenes irreducibilis polinom, viszont x-ben a következő faktorokra bontható

$$f(x) = (x^6 + x^4 + x^3 + x + 1)(x^6 + x^5 + 1)(x^6 + x^5 + x^2 + x + 1).$$

Legyen $r(z)$ egy gyöke z_0 $GF(2^6)$-ban, könnyen leellenőrizhető, hogy a három irreducibilis faktora $f(x)$-nek lesz z_0 három köbgyökének minimálpolinoma $GF(2^6)$-ban. Az is megfigyelhető, hogy ezen példában, mindhárom primitív. Ezért közülük a lexikografikus rendezés alapján $x^6+x^4+x^3+x+1$ a legkisebb, és ez lesz $C_{2,6}(x)$.

Ebben az esetben, mikor n kicsi, a lehetséges polinomokat, már maga $f(x)$ faktorizációjában megkapjuk. A gyakorlatban, valójában z_0 α köbgyökében egyenként kapjuk ezeket a polinomokat, majd minden primitív köbgyök minimálpolinomát meg kell vizsgálni.

5. Példa.

Keressük meg $C_{3,6}(x)$-t. Ekkor $p = 3,\ n = 6,\ q_1 = 2,\ q_2 = 3,\ s = 2,\ d_1 = 3,\ d_2 = 2,\ m_1 = (3^6 - 1)/(3^3 - 1) = 28,\ m_2 = (3^6 - 1)/(3^2 - 1) = 91$, és

$g = lnko(28, 91) = 7$. Kikeressük $C_{3,3}(x) = x^3 - x + 1$ és $C_{3,2}(x) = x2 - x - 1$ polinomokat. $z = x^7$ választásával kapjuk

$$C_{3,3}(x^{28}) = x^{84} - x^{28} + 1 = z^{12} - z^4 + 1$$
$$C_{3,2}(x^{91}) = x^{182} - x^{91} - 1 = z^{26} - z^{13} - 1.$$

A két polinom legnagyobb közös osztója

$$f(x) = x^{42} - x^{35} - x^{28} + x^{21} - 1 = z^6 - z^5 - z^4 + z^3 - 1 = r(z)$$

Ez egy z ismeretlenes irreducibilis polinom, viszont x-ben a következő faktorokra bontható

$$f_1(x) = x^6 - x^4 + x^2 - x - 1$$
$$f_2(x) = x^6 + x^5 - 1$$
$$f_3(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$
$$f_4(x) = x^6 + x^5 - x^4 + x^3 + x - 1$$
$$f_5(x) = x^6 - x^5 - x^3 - 1$$
$$f_6(x) = x^6 - x^5 - x^4 - x^2 + x - 1$$
$$f_3(x) = x^6 - x^5 - x^4 + x^3 + x^2 + x - 1.$$

Ebben az esetben, nem mindegyik faktor primitív. Vegyük például a negyedik faktort, $o(f_4(x)) = 8$ $GF(3^6)$-ban, nem pedig $3^6 - 1$. Viszont tudjuk, hogy van primitív faktor. Ezen primitív faktorok közül, lexikografikusan $f_1(x)$ a legkisebb, ezért $C_{3,6}(x) = f_1(x)$.

Feltéve, hogy $n > 1$, az algoritmus gyökkereső folyamat futási ideje lineárisan nő g-vel. Valójában, $s < 3$ esetén g nagysága meghatározza, hogy mely p és n értékekre érdemes ezt az algoritmust használni $C_{p,n}$ keresésére. Sajnos, bármely rögzített p esetén, g nagyon szabálytalanul változik n növelésével. Ha n maga is egy prím hatvány, akkor g különösen nagy, és az algoritmus nagyon lelassul. Viszont, ezekre az esetekre jól használható a brute force algoritmus, mivel a keresés több primitív, összeegyeztethető elemre van kiterjesztve, így az gyorsabban fogja megtalálni az első lehetséges polinomot. Ezt a következő részben bővebben kifejtem. [Heath, Loehr]

// **6. Példa.** Keressük meg $C_{3,10}(x)$-t. Ekkor $p = 3,\ n = 10,\ q_1 = 2,\ q_2 = 5,\ s = 2,\ d_1 = 5,\ d_2 = 2,\ m_1 = (3^10 - 1)/(3^5 - 1) = 244,\ m_2 = (3^10 - 1)/(3^2 - 1) = 7381$, és $g = lnko(244, 7381) = 61$. Kikeressük $C_{3,5}(x) = x^5 - x + 1$ és $C_{3,2}(x) = x2 - x - 1$ polinomokat. $z = x^61$ választásával kapjuk

$$C_{3,5}(x^{244}) = x^{1220} - x^{244} + 1 = z^{20} - z^4 + 1$$
$$C_{3,2}(x^{7381}) = x^{14762} - x^{7381} - 1 = z^{242} - z^{121} - 1.$$

A két polinom legnagyobb közös osztója

$$f(x) = x^{610} + x^{549} - x^{488} + x^{427} - x^{366} - x^{305} + x^{183} + x^{122} + x^{61} - 1 =$$
$$= z^{10} + z^9 - z^8 + z^7 - z^6 - z^5 + z^3 + z^2 + z - 1 = r(z)$$

Ez egy z ismeretlenes irreducibilis polinom, viszont x-ben a 61 különböző 10-ed fokú faktorra bontható. Ezek közül, a következő lesz a lexikografikus rendezés szerint a legkisebb

$$f(x) = x^{10} - x^6 - x^5x^4 + x - 1.$$

Ez ráadásul primitív is, ezért $C_{3,10}(x) = f(x)$.

//

Az algoritmus helyességét igazoló tétel bizonyításában szó volt arról, hogy egy táblázatból kikeressük a keresett Conway polinom fokának osztóira létező kisebb Conway polinomokat, én ehhez *[Frank Luebeck]* honlapját használtam, itt is ellenőriztem le az utolsó példára kapott megoldásom.

3.4. Régi és új algoritmusok összehasonlítása

Mint azt már tudjuk, az új $C_{p,n}$ Conway polinom keresőalgoritmusunk futási ideje nagyban függ attól, hogy mekkora

$$g = lnko_{1 \le i \le s} \left\{ \frac{M_{p,n}}{M_{p,n/q_i}} \right\}.$$

Ha g viszonylag kicsi, vagyis 8 vagy annál kevesebb számjegy hosszú, akkor $C_{p,n}$ véges időn belül kiszámolható. Például, a $C_{2,42}$ megkeresése 59 másodpercbe telt, itt $g = 5419$, és az igénybe vett idő lényeges részét az $lnko$ kiszámítása tette ki. Összehasonlításképpen, a brute force algoritmusnak $C_{2,42}$

megkeresése többnapnyi futtatás után sem adott eredményt. A következő táblázat tartalmazza a különböző Conway polinomok kereséséhez szükséges időtartamot.

p	n	g	CPU futási idő (mp)
2	40	61681	125
2	42	5419	59
2	44	838861	1947
2	45	14709241	34396
2	46	2796203	16336
2	48	65281	203
2	50	1016801	2844
2	52	13421773	38880
2	54	261633	1000
2	56	15790321	51350
2	60	80581	318
3	26	398581	1332
3	28	478297	1361
3	34	32285041	197852
3	36	530713	2577
5	20	375601	830
5	22	8138021	26164
5	24	39001	1172
7	20	5649505	12217
7	24	5762401	24007
11	14	1623931	2745
11	18	1170231	6201
13	14	4482037	6737

Nehéz összehasonlítani a javított algoritmust a brute force algoritmussal, ugyanis nagy n és p értékek esetét legtöbbször csak az egyik fut le elfogadható időn belül. Azonban, egyértelmű a javított algoritmus futási idő növekedésének g általi linearitása. Míg a brute force p^n lehetséges polinom közül keresi

ki a lexikografikus értelemben első összeegyeztethető primitív polinomot. Az összeegyeztethető, primitív polinomok száma $c = \tau(M, M_{p,n})$, és g méretét kell összehasonlítanunk. Ez azt jelenti, hogy várhatóan p^n/c idő telik el mire talál egy ilyen polinomot, ez p^n/g megközelítőleg. Ez adja meg a magyarázatot arra, hogy míg az egyik algoritmusunk viszonylag gyorsan és precízen lefut, addig a másik algoritmus miért olyan lassú.

Például, keressük a $C_{2,n}$ Conway polinomot, ahol $40 \leq n < 70$. Táblázatba szedjük (1.ábra) minden n értékhez tartozó g, c, és $h = p^n/c$ értékeket. A brute force algoritmus gyors, ha h kicsi, pont mint ahogy a javított algoritmus kis g esetén.

Tanulságos, továbbá, különböző n értékekre megfigyelni g és h értékének relatív méretét. Ha n prím, akkor $g = M_{p,n}/(p-1)$ h pedig elég kicsi. Ebben az esetben, a brute force algoritmus meglehetősen jól működik, mivel jó pár primitív, összeegyeztethető polinom van ekkor a keresési elemek közt, így az elsőt hamar megtalálhatjuk. Viszont a javított algoritmus, ekkor egyáltalán nem hatékony, ugyanis az összes g lehetőséget végig kell vennie ahhoz, hogy a lexikografikusan legkisebbet megtalálja.

Másrészt, összetett n esetén, g értéke kicsinek mutatkozik, szinte mindig sokkal kisebb p^n/c-nél. Példának okáért figyeljük $n \in \{40, 42, 44, 48, 50, 52, 60, 66\}$ értékeket az előző táblázatból, és vegyük észre azon méretbeli különbségeket, melyek a javított algoritmus mellett szólnak. Ekkor a brute force algoritmus szinte mindig elbukott. Csak, hogy legyen egy kis viszonyítási alapunk, a *Sun Ultra Sparc 30* számítógépek, a kétezres évek elején, minden Conway polinomot $g < 10^8$-ra három napon belül ki tudtak számolni. Természetesen, 9 és 10 számjegyű g-re is működik az algoritmus, csak a munkamenet arányos növekedésével, hetekről vagy hónapokról van ekkor szó. [Heath, Loehr]

n	g	$c = \tau(M, \mathrm{M}_{p,n})$	$h = p''/c$
40	61 681	61 680	17 826 064
41	2 199 023 255 551	2 198 858 730 832	1
42	5 419	5418	811 747 233
43	8 796 093 022 207	8 774 777 333 880	1
44	838 861	836 352	21 034 428
45	14 709 241	14 685 300	2 395 890
46	2 796 203	2 796 202	25 165 830
47	140 737 488 355 327	140 646 443 289 600	1
48	65 281	64 512	4 363 141 380
49	4 432 676 798 593	4 432 676 798 592	127
50	1 016 801	1 012 500	1 111 999 907
51	2 454 285 751	2 429 105 112	927 007
52	13 421 773	13 076 544	344 402 896
53	9 007 199 254 740 991	9 005 653 101 120 000	1
54	261 633	261 630	68 854 483 467
55	567 767 102 431	566 942 112 000	63 549
56	15 790 321	15 790 320	4 563 403 024
57	39 268 347 319	39 267 102 096	3 670 125
58	178 956 971	175 923 744	1 638 382 458
59	576 460 752 303 423 487	576 457 548 871 463 200	1
60	80 581	79 200	14 557 089 704 631
61	2 305 843 009 213 693 951	2 305 843 009 213 693 950	1
62	715 827 883	715 827 882	6442 450 950
63	60 247 241 209	60 246 498 816	153 093 909
64	4 294 967 297	4 288 266 240	4 301 678 823
65	145 295 143 558 111	145 295 143 558 110	253 921
66	1 397 419	1 376 496	53 604 933 319 703
67	147 573 952 589 676 412 927	147 573 951 827 644 447 920	1
68	3 435 973 837	3 407 185 152	86 625 144 220
69	10 052 678 938 039	10 052 678 938 038	58 720 249

1. ábra. A futási időt befolyásoló paraméterek a két algoritmusban [Heath, Loehr]

4. Véges testek kódoláselméletben betöltött szerepe

A véges testek elmélete sokféle alkalmazási lehetősége révén az elmúlt évszázadban különösen fontossá vált. Ilyen alkalmazás például a kódelmélet és a titkosírás, de különféle kombinatorikai problémák megoldása során is felmerülnek véges testek. A véges testek jelentősége a 20. század közepén nőtt meg, az információelmélet megalapozásával (Shannon, 1948).

Az egyik legfontosabb alkalmazása a véges testeknek a kódoláselméletben van. A hibajelző és -javító, illetve ciklikus kódok, azon belül is a BCH-kódok és a Reed-Solomon kódok. Ezek a kódok többszörös hibákat is képesek javítani, és dekódolásukra hatékony algoritmusok vannak. Alkalmazhatóságukat és hatékonyságukat jól mutatja az, hogy BCH, illetve Reed-Solomon kódot használnak több területen is, mint például a műholdas kommunikáció során, CD és DVD lemezek hibajavításánál és kétdimenziós vonalkódoknál, például a QR-kódok hibajavítására is. De a diszkrét logaritmuson alapuló titkosírásban is nélkülözhetetlen. Ezen eredmények és még néhány említésre méltó kódolás elméleti eredményt láthatunk a következőkben.

Elsőnek nézzük a titkosírást:

- **Diffie-Hellman-kulcsváltás**

  Tegyük fel, hogy az A és B személyek akik esetleg korábban soha nem kommunikáltak nyílt csatornát használva egy közös kulcsban kívánnak megállapodni. Először is választanak egy (nagy) q prímhatványt. Legyen $F_q^* = GF(q)^* = \langle hGF(q)\ 0; \cdot \rangle$ az $F_q = GF(q)$ egy véges test multiplikatív csoportja. Ebben a ciklikus csoportban választanak egy g generátorelemet. Eddig minden nyilvános. Ezt követően A választ egy véletlen a elemet, B pedig egy véletlen b elemet Z_{q-1}-ből. Majd A elküldi g^a-t B-nek, B pedig elküldi g^b-t A-nak. Ezt követően a közös titkos kulcs $g^a b$. Mármost B könnyen megkapja ezt a kulcsot: megkapta g^a-t, és ezt a csak általa ismert b kitevőre emeli. Hasonlóan kapja meg A is a közös kulcsot: g^b-t a csak általa ismert a kitevőre emeli.

Jelen tudásunk szerint az ellenség tehetetlen: hiába ismeri a két hatványt, g^a-t és g^b-t, és g-t, a kitevőket innen nem tudja meg, ehhez a diszkrét logaritmust kellene kiszámolnia.

- **Massey–Omura-rejtjelrendszer** és **ElGamal rejtjelrendszer**

  Ezen rendszerek alapja a nyilvános g generátorelem, és a saját véletlen választott, bizonyos egyedi feltételeket teljesítő, hatványok ismeretében a beszélgető partner segítségével közösen történő kulcs megkeresése. Mindezekhez elengedhetetlen az elektronikus aláírás, a biztonságos kommunikációhoz szükséges a másik fél kilétéről megbizonyosodni.

- **Silver-Pohlig-Hellman algoritmus**

  Ezen algoritmus célja a diszkrét logaritmus kiszámolása $GF(q)$-ban, mikor $q-1$ minden prímtényezője kicsi. Ez két generátorelem ismeretében végezhető el.

Hibajelző és -javító algoritmusok

Ezen algoritmusok ismertetése több időt venne igénybe komplexitásuk miatt, ezért csak felsorolok párat.

- Hamming-kódok
- Elsőrendű Reed-Müller-kódok
- Kiterjesztett bináris Golay-kódok
- BCH-kódok
- Reed-Solomon-kódok

[Lidl, Niederreiter]

Továbbá van ezen kódoknak egy külön csoportja ahol a Conway polinomok által meghatározott véges testeket, és azon belül is a nagy elemszámúakat használjuk. Bankok biztonsági rendszerében van rájuk szükség, de a zsebünkben lapuló elektronikus személyigazolványokhoz is elengedhetetlen. A következő részben egy kis betekintést teszünk a biztonsági kódolás világába.

4.1. Biztonsági kódolás (Safety and security)

Ennek a témának a matematikai része külön fejezetet érdemelne, illetve külön lehetne róla írni egy publikációt. Ezért csak ismertetés céljából foglaltam dolgozatomba.

A *Digital Signature Standards*, röviden DSS, magyarul *Digitális Aláírási Szabvány*, egy olyan nyilvános, mindenki számára elérhető publikáció ami magában tartalmazza azon szabványokat,elvárásokat, amik alapján, például egy bank, teljes biztonságban tudja lebonyolítani átutalásait.

Ez a szabvány meghatározza azokat az algoritmuskészleteket, amelyek felhasználhatók digitális aláírás generálására. A digitális aláírást az adatok jogosulatlan módosításának észlelésére és az aláíró személyazonosságának elektronikus hitelesítésre használják. Ezen felül az aláírt adatok címzettje a digitális aláírást arra is használhatja, hogy bizonyíthassa egy harmadik fél számára, hogy a tényleges küldőtől érkeztek az adatok. Ezt úgy nevezzük, hogy letagadhatatlanság, mivel az aláíró nem tudja könnyen letagadni az aláírást későbbi időpontban.

A témához leginkább vágó része a cikknek a prím testek feletti elliptikus görbék:

Minden p prím esetén létezik egy pseudo-random görbe, hogy

$$E: \ y^2 \equiv x^3 - 3x + b(mod\ p)$$

ahol n prím hatvány.

Ennél jobban nem is szeretnék belebonyolódni a témába.

Még csak annyit szeretnék megmutatni, hogy mekkora számokról van itt szó.

- **P-256 görbe**

  p =115792089210356248762697446949407573530086143
  415290314195533631308867097853951

  n =115792089210356248762697446949407573529996955224135703424
  22259061068512044369

- **P-521 görbe**

p =6864797660130609714981900799081393217269435300143305493
94463459185543183397655394245057746333217197532963996371
36332111386476861244038034037280889270700544 9

n =6864797660130609714981900799081393217269435300143305493
94463459185543183397655394245057746333217197532963996371
36332111386476861244038034037280889270700544 9

Innen is látható, hogy milyen méretekről is beszéltünk. Emlékezzünk vissza, az általunk ismertetett algoritmus 8 számjegy hosszú g-re futott nagyjából 3 napig, akkor P-521-nél ahol a nagyságrend 10^{60}, meddig tart megtalálni a megfelelő Conway polinomot. Ne felejtsük el, ezek nem csak biztonsági céllal léteznek, hanem helyreállító szerepük is, az adatvesztés megelőzése okán. [Digital Signature Standards]

Hivatkozások

[Heath, Loehr] Lenwood S. Heath, Nicholas A. Loehr: *New algorthms for generating Conway polynomials over finite fields*. Journal of Symbolic Computation 38 (2004) 1003-1024, Elsevier, 2004

[Lidl, Niederreiter] Lidl, R., Niederreiter, H.:*Introduction to Finite Fields and Their Applications*. Cambridge University Press, Cambridge, 1994

[Frank Luebeck] http://www.math.rwth-aachen.de/ Frank.Luebeck/data/ConwayPol/index.htm utolsó elérés:2020.05.16. 23:30

[Digital Signature Standards] FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION,Digital Signature Standard (DSS),Information Technology Laboratory, National, Institute of Standards and Technology, Gaithersburg, MD 20899-8900, Issued July 2013

Nyilatkozat

Alulírott Hasznosi Tóth Csongor, matematika mesterszakos hallgató, kijelentem, hogy a szakdolgozatban foglaltak a saját munkám eredményei, és csak a hivatkozott forrásokat használtam fel. A szakdolgozat a 2018-1.2.1-NKP-2018-00004 számú "IoT rendszerek biztonságát növelő technológiák (SETIT)" projekt keretében, a Nemzeti Kutatási és Innovációs Alapból biztosított támogatással, a "Nemzeti Kiválósági Program: 2018-1.2.1-NKP" pályázati program finanszírozásában valósult meg. Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem könyvtárában a kölcsönözhető könyvek között helyezik el, és az interneten is nyilvánosságra hozhatják.

2020. május 16.

Hasznosi Tóth Csongor