УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Јована Павловић

ВЕЛИКИ ЛОВ НА ТРИНОМЕ

мастер рад

Београд, 2023.

Ментор:

др Миодраг Живковић
Универзитет у Београду, Математички факултет

Чланови комисије:

др Весна Маринковић, доцент
Универзитет у Београду, Математички факултет

др Стефан Мишковић, доцент
Универзитет у Београду, Математички факултет

Датум одбране: 2023.

Породици

Наслов мастер рада: Велики лов на триноме

Резиме: У овом раду разматра се концепт несводљивих и примитивних тринома над пољем GF(2) и њихова улога у криптографији и генерисању случајних бројева. Указује се на везу између примитивних тринома и Мерсенових простих бројева. Описује се укратко пројекат GIMPS и његова улога у проналажењу великих Мерсенових простих бројева. Такође се разматра алгоритам за тестирање несводљивости тринома над GF(2), као и процес просијавања који се користи за смањење броја кандидата за тестирање. Представљају се и објашњавају методе првог и другог нивоа блокирања које имају за циљ смањење сложености операције факторисања тринома.

Кључне речи: несводљиви и примитивни триноми, GF(2), Мерсенови прости бројеви, GIMPS, просијавање, блокирање

Садржај

1 Увод **1**

2 Потрага за триномима **4**

2.1 Теоријске основе 4

2.2 Свонова теорема 7

2.3 Основне операције потраге 8

2.4 Тест несводљивости 10

2.5 Просијавање и сертификати 12

2.6 Напреднији приступ потрази 15

2.7 Преглед до сада откривених тринома 17

3 Програмска реализација алгоритама **19**

3.1 Реализација алгоритма просијавања 20

3.2 Реализација алгоритма за први ниво блокирања 21

3.3 Реализација алгоритма за други ниво блокирања 23

3.4 Резултати реализација алгоритама 26

4 Закључак **29**

Библиографија **31**

Глава 1

Увод

У данашње вријеме све већи акценат се ставља на сигурност и поузданост комуникације. Криптографија, наука која се бави осигуравањем тајности и интегритета података, постаје све важнија у свијету дигиталних комуникација. Један од кључних елемената криптографских система је генерисање случајних бројева. У потрази за ефикасним методама генерисања таквих бројева, посебна пажња посвећује се триномима над пољем GF(2).

Овај рад фокусира се на триноме облика $x^r + x^s + 1$, $r > s > 0$, над пољем GF(2), гдје је r Мерсенов експонент. Триноми над GF(2) имају значајну улогу у криптографији и генерисању случајних бројева због својих својстава. Посебно су интересантни примитивни триноми, који имају кључну улогу у генерисању псеудослучајних бројева. Примитивни триноми $x^r + x^s + 1$ се користе као основа за конструкцију коначних поља GF(2^r) са 2^r елемената, гдје r представља степен тринома. Да би се разумјело о каквим триномима је ријеч, морамо се упознати са Мерсеновим простим бројевима и пројектом GIMPS.

Мерсенов број је позитиван природан број облика $M_n = 2^n - 1$, гдје је n природан број. Мерсенови прости бројеви су прости бројеви облика $2^p - 1$, при чему је p такође прост број. Име су добили по француском математичару М. Мерсену (Marin Mersenne, 1588–1648), који их је проучавао почетком 17. вијека. Мерсенов експонент је експонент p који одговара Мерсеновом простом броју $2^p - 1$. Важно је напоменути да број $2^p - 1$ може бити прост само ако је p прост, али не важи да је за сваки прост број p и $2^p - 1$ прост. Најмањи сложен Мерсенов број са простим експонентом је $2^{11} - 1 = 23 * 89$. Ако је p сложен,

онда је и $2^p - 1$ сложен. Није познато да ли Мерсенових простих бројева има коначно или бесконачно много. [1] Мерсен је у својој књизи Cogitata physico mathematica 1644. године изнио тврдњу да је $2^p - 1$ прост број за $p = 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257$, те да је $2^p - 1$ сложен број за све остале просте бројеве $p < 257$. Неколико стотина година нико није успио потврдити, ни оповргнути ту тврдњу. Каснија истраживања су показала да је направио 5 грешака – укључио је у списак бројеве $2^{67} - 1$ и $2^{257} - 1$, који су сложени, и изоставио бројеве $2^{61} - 1$, $2^{89} - 1$ и $2^{107} - 1$, који су прости. Амерички математичар Кол (Frank Nelson Cole) је 1903. године доказао сложеност Мерсеновог броја $2^{67} - 1$ ($2^{67} - 1 = 193707721 \cdot 761838257287$).

Стари Грци су били упознати са прва четири Мерсенова проста броја: $M_1 = 3$, $M_2 = 7$, $M_3 = 31$, $M_4 = 127$. Пети Мерсенов прост број, $M_5 = 8191$, откривен је 1461. године и није познато ко га је открио. Каталди (Pietro Antonio Cataldi) је 1588. године открио следећа два броја, $M_6 = 2^{17} - 1$ и $M_7 = 2^{19} - 1$. Након скоро два вијека, 1722. године, Ојлер (Leonhard Euler) је утврдио да је $2^{31} - 1$ прост Мерсенов број. Иако број $2^{127} - 1$ није био следећи по величини који је откривен, Лукас (François Édouard Anatole Lucas) га је идентификовао 1876. године. Затим, 1883. године, Первушин (Ivan Mikheevich Pervushin) је открио број $2^{61} - 1$. На почетку 20. вијека Пауерс (Ralph Ernest Powers) је открио још два броја, $2^{89} - 1$ 1911. године и $2^{107} - 1$ 1914. године.

Појавом електронских рачунара, потрага за Мерсеновим простим бројевима доживјела је значајан напредак. Тјуринг (Alan Mathison Turing) је 1949. године користио рачунар Manchester Mark 1 у потрази за овим бројевима. Након 38 година од последњег проналаска, следећи Мерсенов прост број, $2^{521} - 1$, је успешно откривен 1952. године на рачунару SWAC на Институту нумеричке анализе Универзитета у Калифорнији, под вођством Лемера (Derrick Henry Lehmer). Програм је написао Робинсон (Raphael Mitchel Robinson). Непуна два сата касније откривен је број $2^{607} - 1$. У наредним мјесецима, истим програмом су идентификовани бројеви $2^{1279} - 1$, $2^{2203} - 1$ и $2^{2281} - 1$. Број $2^{4423} - 1$ је био први откривени прост број са више од 1000 цифара, $2^{44497} - 1$ је био први број са више од 10.000 цифара, док је $2^{6972593} - 1$ био први број са преко 1.000.000 цифара. За сада је највећи 51. откривени Мерсенов прост број,

[1] Дискусија о томе се може пронаћи на https://t5k.org/mersenne/heuristic.html

$2^{82.589.933} - 1$. Открио га је Ларош (Patrick Laroche) 7. децембра 2018. године, у оквиру пројекта GIMPS. То је уједно и највећи познати прост број, и он има 24.862.048 цифара. [2]

GIMPS (Great Internet Mersenne Prime Search) је пројекат који се бави проналажењем и провјером Мерсенових простих бројева. Покренуо га је Волтман (George Woltman) 1996. године, после чега је постао један од највећих дистрибуираних рачунарских пројеката. Удружење броји више од 100.000 чланова који трагају за што већим Мерсеновим простим бројевима. До октобра 2022. године пројекат је открио 17 Мерсенових простих бројева, од којих су 15 били највећи познати прости бројеви у вријеме када су откривени. Мерсенови прости бројеви се обично нумеришу у растућем редоследу величина: $M_1 = 2^2 - 1 = 3$, $M_2 = 2^3 - 1 = 7$, $M_3 = 2^5 - 1 = 31$, $M_4 = 2^7 - 1 = 127$, итд. GIMPS не проналази увијек Мерсенове просте бројеве по реду, и зато може постојати извјесна несигурност у њиховом нумерисању. Да би се то избјегло, користи се ознака M'_n за n–ти по реду откривени Мерсенов прост број. Постоје празнине у потрази изнад броја $M_{39} = 2^{13.466.917} - 1$, па може да се деси да је $M'_n > M'_{n+1}$ за n>39. На примјер, број $M'_{45} = 2^{43.112.609} - 1$ је пронађен прије бројева $M'_{46} = 2^{37.156.667} - 1$ и $M'_{47} = 2^{42.643.801} - 1$. У наставку рада ћемо са r_n обиљежавати експонент броја M_n, док ћемо са r'_n обиљежавати експонент броја M'_n.

[2] Доступно на https://www.mersenne.org/primes/press/M82589933.html

Глава 2

Потрага за триномима

Анализа и класификација тринома облика $x^r + x^s + 1$ је од суштинског значаја за разумијевање њихове примјене и потенцијала. У овој глави представљају се математичке основе, теоретски концепти и технике које стоје иза ових тринома. Упућује се на основне појмове о полиномима и пољу GF(2). Да би се разумјели потенцијални кандидати за испитивање, представља се теорема која говори о парности несводљивих фактора тринома. Разматра се сложеност основних операција кључних за алгоритамске приступе, као и критеријуми за несводљивост тринома. С обзиром на важност ефикасности у пракси, истражује се техника просијавања тринома, као и иновативне методе убрзања и оптимизације технике. На крају, представљају се резултати истраживања кроз табелу пронађених тринома.

2.1 Теоријске основе

Трином је полином над једном промјенљивом са три не–нула члана [5], на примјер $P(x) = x^5 + x^2 - 8$. Другим ријечима, трином представља суму три монома. Ако коефицијенти тринома припадају неком прстену или пољу F, тада кажемо да је то трином дефинисан над F, у ознаци $P \in \mathrm{F}[x]$. Као што је наведено у уводном дијелу рада, посматрамо триноме облика $x^r + x^s + 1$, гдје су r и s позитивни цијели бројеви, $r > s > 0$, над коначним пољем GF(2). То је поље са два елемента, 0 и 1, гдје се операције сабирања и множења изводе по модулу 2:

+	0	1
0	0	1
1	1	0

*	0	1
0	0	0
1	0	1

Као последицу овог својства имамо да је $(P + M)^2 = P^2 + M^2$, ако су $P, M \in \mathrm{GF}(2)$, јер двоструки члан $2PM$ нестаје. Друге ознаке за ово поље су $\mathbb{Z}/2\mathbb{Z}$, која се често користи у алгебри и теорији бројева, и F_2, која се уобичајено користи у теорији поља и рачунарству. Када се ради над GF(2), увијек можемо писати + умјесто −, пошто је 1+1=0, тј. 1 = –1.

Дефиниција 2.1.1. *Нека је* F *поље. Полином* $f(x) \in \mathrm{F}[x]$ *је сводљив над* F *ако и само ако постоји полином* $g(x) \in \mathrm{F}[x]$*, такав да* $g(x)$ *дијели* $f(x)$ *и* $0 < deg(g) < deg(f)$*. Ако такав полином постоји, називамо га фактором полинома* $f(x)$*. Полином је несводљив ако и само ако није сводљив, тј. не може се растaвити у производ два или више неконстантних полинома са коефицијентима у* F*.*

На примјер, трином x^4+x+1 је несводљив, док је трином x^5+x+1 сводљив, јер се може представити у облику производа $x^5+x+1 = (x^2+x+1)(x^3+x^2+1)$.

Дефиниција 2.1.2. *Несводљиви полином* P *степена* $r>0$ *је примитиван ако је* $P(x) \neq x$ *и остаци* $x^k \mod P$, $0 \leq k < 2^r - 1$*, су различити.*

На примјер, трином $x^3 + x + 1$ је примитиван, јер је $x^3 + x + 1 \neq x$ и остаци $x^k \mod (x^3 + x + 1)$, $0 \leq k < 7$, су различити:

$$\begin{aligned}
x^0 &\mod (x^3 + x + 1) = 1,\\
x^1 &\mod (x^3 + x + 1) = x,\\
x^2 &\mod (x^3 + x + 1) = x^2,\\
x^3 &\mod (x^3 + x + 1) = x + 1,\\
x^4 &\mod (x^3 + x + 1) = x^2 + x,\\
x^5 &\mod (x^3 + x + 1) = x^2 + x + 1,\\
x^6 &\mod (x^3 + x + 1) = x^2 + 1
\end{aligned}$$

Несводљиви полином $P(x)$ степена $r > 1$ може се користити за представљање коначног поља $\mathrm{GF}(2^r)$ са 2^r елемената. Елементи овог поља су бинарни полиноми [1] степена мањег од r, са коефицијентима из поља GF(2). Сабирање

[1] Бинарни полином је полином чији су коефицијенти ограничени на поље GF(2), што значи да могу узети само вредности 0 или 1. Назива се бинарним полиномом јер укључује само два могућа коефицијента.

бинарних полинома се врши сабирањем коефицијената по модулу 2, док се множење врши коришћењем правила множења заједно са свођењем резултата по модулу несводљивог полинома $P(x)$.

На примјер, за дефинисање коначног поља $\mathrm{GF}(2^3)$ потребан је несводљиви полином степена $r = 3$. Један такав полином је $x^3 + x + 1$. Поље $\mathrm{GF}(2^3)$ има 8 елемената и то су $\{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$. Сабирање у овом пољу је једноставно сабирање полинома са коефицијентима у $\mathrm{GF}(2)$:

$$(x^2 + x) + (x^2 + 1) = x + 1.$$

Множење се врши множењем полинома, а затим узимањем остатка при дијељењу са полиномом $x^3 + x + 1$:

$$(x^2 + x) * (x^2 + 1) = x^4 + x^3 + x^2 + x,$$

$$(x^4 + x^3 + x^2 + x) \mod (x^3 + x + 1) = x + 1.$$

Стога, поље $\mathrm{GF}(2^r)$ је изоморфно пољу $\mathrm{GF}(2)[x]/P(x)$, гдје је $P(x)$ несводљиви полином степена r [9]. Полином x је генератор мултипликативне групе коначног поља $\mathrm{GF}(2)[x]/P(x)$ ако и само ако је полином $P(x)$ примитиван. Приликом утврђивања да ли је полином $P(x)$ примитиван, провјера за све вриједности k у интервалу $0 \leq k < 2^r - 1$ може бити неефикасна, посебно за велике бројеве r. Ако x генерише мултипликативну групу коначног поља $\mathrm{GF}(2)[x]/P(x)$, онда је $x^k \neq 1 \mod P$ за оне вриједности k који су дјелиоци броја $2^r - 1$, осим броја $k = 2^r - 1$. Обрнуто, ако је $x^k = 1 \mod P$ за неки број k који је дјелилац броја $2^r - 1$, онда x не може генерисати цијелу мултипликативну групу, што значи да полином $P(x)$ не може бити примитиван. Дакле, да би се провјерила примитивност несводљивог полинома $P(x)$, потребно је само провјерити да је $x^k \neq 1 \mod P$ за оне бројеве k који су нетривијални дјелиоци броја $2^r - 1$.

На примјер, посматрајмо несводљиви полином $x^4 + x + 1$. Да бисмо провјерили да ли је он примитиван, довољно је испитати остатак $x^k \mod (x^4 + x + 1)$ за оне вриједности k које су нетривијални дјелиоци броја $2^4 - 1 = 15$, а то су 3 и 5.

$$x^3 \mod (x^4 + x + 1) = x^3,$$
$$x^5 \mod (x^4 + x + 1) = x^2 + x$$

С обзиром на то да је $x^3 \neq 1$ и $x^2 + x \neq 1$, полином $x^4 + x + 1$ је примитиван.

Ако је r велико и $2^r - 1$ није прост број, може бити изазовно тестирати примитивност полинома степена r, јер је потребно знати просте чиниоце броја $2^r - 1$. Са друге стране, ако је број $2^r - 1$ прост, како прости бројеви немају нетривијалне факторе, сви несводљиви полиноми степена r су примитивни. Из тог разлога ограничавамо нашу пажњу на степене r који су Мерсенови експоненти.

Пошто је полином $x^r + x^s + 1$ несводљив ако и само ако је реципрочни полином $x^r + x^{r-s} + 1$ несводљив, приликом тражења несводљивих тринома степена r можемо претпоставити да је $s \leq \frac{r}{2}$.

2.2 Свонова теорема

Свонова (Swan) теорема [5] даје једноставан начин да се одреди парност броја несводљивих фактора бинарног тринома. Ово је кључно, јер парност броја несводљивих фактора одређује да ли је трином сводљив или не. Ако трином има паран број несводљивих фактора, он је сводљив. С друге стране, ако трином има непаран број несводљивих фактора, постоји могућност да је несводљив: непаран број фактора може укључивати само један фактор, тј. сам трином. Стога, Свонова теорема, коју наводимо без доказа, даје неопходан услов (непаран паритет) да трином буде несводљив. [2]

Теорема 2.2.1. *Нека је $r > s > 0$ и претпоставимо да је $r + s$ непаран број. Тада трином $T_{r,s}(x) = x^r + x^s + 1$ има паран број несводљивих фактора над* GF(2) *у следећим случајевима:*

(а) број r је паран, $r \neq 2s$, $\frac{rs}{2} \mod 4 \in \{0, 1\}$.

(б) број r је непаран, $s \nmid 2r$, $r \mod 8 = \pm 3$.

(в) број r је непаран, $s \mid 2r$, $r \mod 8 = \pm 1$.

У свим осталим случајевима трином $T_{r,s}(x)$ има непаран број несводљивих фактора.

Ако су оба броја r и s парни, онда је $T_{r,s}(x)$ квадрат, па самим тим није несводљив. Ако су и r и s непарни, можемо примјенити теорему на реципрочни

[2] Доказ за теорему се може пронаћи на [1]

трином $T_{r,r-s}(x) = x^r \cdot T\left(\frac{1}{x}\right) = x^r + x^{r-s} + 1$, пошто $T_{r,s}(x)$ и $T_{r,r-s}(x)$ имају исти број несводљивих фактора, при чему је експонент $r - s$ паран број. [7]

Случај када је r паран, а s непаран број се не разматра у овој потрази. Разлог за то је што је r Мерсенов експонент, што значи да не може бити паран број. Једини изузетак је тривијалан случај $r = 2$.

Ако је r непаран, а s паран број, разликују се два случаја. Број $2r$ има четири дјелиоца, а то су $\{1, 2, r, 2r\}$. Уколико је $r \equiv \pm 3 \pmod 8$ и s није дјелилац броја $2r$, тада s може узети $r - 3$, односно због парности $\frac{r-3}{2}$ различите вриједности за које трином $T_{r,s}(x)$ има паран број несводљивих фактора. За потрагу су заправо значајне преостале двије вриједности, тј. због парности s једна вриједност, будући да за њу важи да трином $T_{r,s}(x)$ има непаран број несводљивих фактора. Та вриједност је $s = 2$.
На примјер, из $r = 19$ слиједи да је $r \mod 8 = 3$. Дјелиоци броја $2r = 38$ су $\{1, 2, 19, 38\}$. За s које узима вриједности $\{4, 6, 8, 10, 12, 14, 16, 18\}$ трином $x^r + x^s + 1$ има паран број несводљивих фактора. Преостала парна вриједност за s, таква да је $0 < s < r$, је $s = 2$ и за њу трином $x^r + x^s + 1$ има непаран број несводљивих фактора.

Уколико је $r \equiv \pm 1 \pmod 8$ и s јесте дјелилац броја $2r$, тада за $s = 2$ трином $T_{r,s}(x)$ има паран број несводљивих фактора. У овом случају, за потрагу се разматра s које узима преостале $\frac{r-3}{2}$ различите вриједности, тј. због услова $s \leq \frac{r}{2}$ преостале $\frac{r-3}{4}$ различите вриједности.
На примјер, ако је $r = 17$, онда је $r \mod 8 = 1$. Дјелиоци броја $2r = 34$ су $\{1, 2, 17, 34\}$. За $s = 2$ трином $x^r + x^s + 1$ има паран број несводљивих фактора. Преостале парне вриједности за s, такве да је $0 < s < r$, су $\{4, 6, 8, 10, 12, 14, 16\}$ и за њих трином $x^r + x^s + 1$ има непаран број несводљивих фактора. Додатно, због услова $s \leq \frac{r}{2}$ можемо се сконцентрисати само на вриједности $\{4, 6, 8\}$ и за њих даље испитивати да ли је трином несводљив.

2.3 Основне операције потраге

Основне операције које су потребне у потрази за несводљивим триномима су квадрирање по модулу тринома $T(x) = x^r + x^s + 1$, множење по модулу тринома $T(x)$, као и израчунавање највећег заједничког дјелиоца (GCD) полинома степена мањег од r и тринома $T(x)$. Сложеност ових операција је број операција са битовима или ријечима приликом њиховог извршавања. Поли-

номе можемо представити у бинарном облику, гдје свака јединица означава присутност одговарајућег члана x^n, док нула представља његову одсутност. На примјер, полином $x^4 + x + 1$ може се представити као бинарни број 10011.

Због чињенице да у GF(2) важи једнакост $(x^i + x^j)^2 = x^{2i} + x^{2j}$, у контексту битских операција, квадрирање сваког члана се своди на помјерање бита на лијево за одговарајући број позиција. Када квадрирамо члан x^i резултат је x^{2i}, тј. бит који представља члан x^i се помјера са позиције i на позицију $2i$. Ова операција се може извести у константном времену на рачунару. Број операција које се извршавају једнак је броју чланова у полиному, тачније r, па је сложеност операције квадрирања полинома степена мањег од r једнака $O(r)$.

Када се врши свођење једног полинома по модулу другог полинома, први корак је да упоредимо њихове степене. Ако је степен првог полинома већи или једнак од степена другог полинома, одузимамо други полином од првог. У бинарном облику ово је еквивалентно операцији XOR и константне је сложености, ако су полиноми представљени једном ријечи. Затим помјерамо други полином надесно и понављамо поступак. Ово се наставља док степен првог полинома не постане мањи од степена другог полинома. Резултат је остатак дијељења првог полинома са другим. Свођење полинома степена мањег од $2r$ по модулу тринома T има сложеност $O(r)$ због ријеткости T. Због свега наведеног сложеност операције квадрирања по модулу тринома T једнака је $O(r)$.

Множење полинома је сложености $O(M(r))$, гдје $M(r)$ представља сложеност множења два полинома степена мањег од r над GF(2). „Класични" алгоритам множења полинома је сложености $O(r^2)$, док је сложеност Шенхагеовог (Schönhage) алгоритма једнака $O(r \cdot \log r \cdot \log\log r)$. Како цијена множења доминира над цијеном свођења по модулу тринома T, укупна сложеност операције множења по модулу је $O(M(r))$.

За израчунавање највећег заједничког дјелиоца (НЗД) полинома са степеном ограниченим на r, сложеност је $O(M(r) \cdot \log r)$. Ово се постиже коришћењем приступа „подјели и владај", тј. Еуклидовог алгоритма за полиноме, у комбинацији са Шенхагеовим брзим множењем полинома. [3]

Табела 2.1: Сложеност основних операција

Модуларно квадрирање	$O(r)$
Модуларно множење	$O(M(r))$
GCD	$O(M(r) \cdot \log r)$

2.4 Тест несводљивости

Лема 2.4.1. *Нека је* $f(x) \in \mathrm{F}[x]$, $\deg(f) = r$. *Ако је* $f(x)$ *сводљив, онда има фактор са степеном* $\leq \left\lfloor \frac{r}{2} \right\rfloor$.

Доказ. Претпоставимо да је $f(x)$ сводљив и да има само факторе са степеном $> \left\lfloor \frac{r}{2} \right\rfloor$. Пошто је $f(x)$ сводљив слиједи да је

$$f(x) = g(x)h(x),$$

гдје је $deg(g) > \left\lfloor \frac{r}{2} \right\rfloor$ и $deg(h) > \left\lfloor \frac{r}{2} \right\rfloor$.

Ако претпоставимо да је r паран, имамо да је $\left\lfloor \frac{r}{2} \right\rfloor = \frac{r}{2}$. Затим важи

$$deg(f) = deg(g) + deg(h) > \frac{r}{2} + \frac{r}{2} = r.$$

Ово је контрадикторно са $\deg(f) = r$.

Ако претпоставимо да је r непаран, имамо да је $\left\lfloor \frac{r}{2} \right\rfloor = \frac{r-1}{2}$. Онда је $deg(g) > \frac{r-1}{2}$ и $deg(h) > \frac{r-1}{2}$, или еквивалентно $deg(g) \geq \frac{r+1}{2}$ и $deg(h) \geq \frac{r+1}{2}$. Затим важи

$$deg(f) = deg(g) + deg(h) \geq \frac{r+1}{2} + \frac{r+1}{2} = r + 1.$$

Ово је контрадикторно са $\deg(f) = r$. □

Теорема 2.4.1. *Нека је* $I_{\mathrm{F}_q,d}$ *скуп несводљивих полинома над* F_q *степена* d *и нека је* $\Phi_r := \{f(x) \in I_{\mathrm{F}_q,d} : d \mid r\}$. *Тада је*

$$x^{q^r} - x = \prod_{f(x) \in \Phi_r} f(x).$$

[3]

[3] Доказ теореме није приказан у овом раду, али може се пронаћи у [8].

Ако ову теорему посматрамо над пољем $\mathrm{GF}(2)[x]$, добијамо полином $P_r(x) = x^{2^r} - x$, који представља производ свих несводљивих полинома степена d, гдје d узима вриједности дјелилаца броја r. [10] На примјер, за $r = 2$ и $d \in \{1, 2\}$, $P_2(x) = x^{2^2} - x = x(x+1)(x^2+x+1)$. Полиноми x и $x+1$ су несводљиви полиноми степена 1, док је x^2+x+1 несводљив полином степена 2.

Из теореме 2.4.1 закључујемо да ако полином $f(x)$ има несводљиви фактор $g(x)$ степена d, онда $g(x)$ дијели $\mathrm{GCD}(f(x), x^{q^d} - x)$.

Теорема 2.4.2. *Нека је $f(x) \in \mathrm{F}_q[x]$, $\deg(f) = r$. Полином f(x) је несводљив ако и само ако је $\mathrm{GCD}(f(x), x^{q^d} - x) = 1$ за $d \in \{1, 2, \ldots, \lfloor \frac{r}{2} \rfloor\}$.*

Доказ. $\Leftarrow$ Претпоставимо да је $\mathrm{GCD}(f(x), x^{q^d} - x) \neq 1$ за $d \in \{1, 2, \ldots, \lfloor \frac{r}{2} \rfloor\}$. Нека је $g(x) = \mathrm{GCD}(f(x), x^{q^d} - x)$.

Пошто $g(x)$ дијели $x^{q^d} - x$, онда, на основу теореме 2.4.1, добијамо да $g(x)$ дијели $\prod_{h(x) \in \Phi_d} h(x)$. Слиједи да је $g(x) = \prod_{h(x) \in \Delta} h(x)$, гдје је $\Delta \subseteq \Phi_d$.

Нека је $\hat{h}(x) \in \Delta$. Онда $\hat{h}(x)$ дијели $g(x)$ и $g(x)$ дијели $f(x)$, па $\hat{h}(x)$ дијели $f(x)$. Због тога што је $\hat{h}(x) \in \Delta$ и $\Delta \subseteq \Phi_d$, онда $\hat{h}(x) \in \Phi_d$ и слиједи да је $0 < \deg(\hat{h}) \leq d < \lfloor \frac{r}{2} \rfloor$. Закључујемо на основу дефиниције да је $f(x)$ сводљив.

$\Rightarrow$ Претпоставимо да је $f(x)$ је сводљив. На основу леме 2.4.1 добијамо да полином $f(x)$ има фактор $g(x)$ са степеном $n \leq \lfloor \frac{r}{2} \rfloor$. Из теореме 2.4.1 слиједи да $g(x)$ дијели $x^{q^n} - x$. Закључујемо да је $\mathrm{GCD}(f(x), x^{q^n} - x) \neq 1$. □

Теорема 2.4.3. *Нека је $f(x) \in \mathrm{F}_q[x]$, $\deg(f) = r$. Полином f(x) је несводљив ако и само ако је $x^{q^r} \equiv x \pmod{f(x)}$ и $\mathrm{GCD}(f(x), x^{q^{r/p}} - x) = 1$ за све p који су прости дјелиоци броја r.*

Доказ. $\Leftarrow$ Претпоставимо да $x^{q^r} \not\equiv x \pmod{f(x)}$. Онда $f(x)$ не дијели $x^{q^r} - x$. На основу теореме 2.4.1, $f(x)$ не дијели $\prod_{h(x) \in \Phi_r} h(x)$, па слиједи да је $f(x)$ сводљив.

Претпоставимо да је $\mathrm{GCD}(f(x), x^{q^{r/p}} - x) \neq 1$ за неке просте бројеве p који су дјелиоци броја r. Пошто је p прост број, онда је $p \geq 2$, па је на основу теореме 2.4.2 $f(x)$ сводљив.

$\Rightarrow$ Претпоставимо да је $f(x)$ сводљив и $x^{q^r} \equiv x \pmod{f(x)}$. Због тога што је $f(x)$ сводљив, онда има фактор $g(x)$ са степеном $0 < deg(g) < r$. Такође, с обзиром на то да $f(x)$ дијели $x^{q^r} - x$, на основу теореме 2.4.1, $f(x)$ дијели $\prod_{h(x) \in \Phi_r} h(x)$. Будући да $g(x)$ дијели $f(x)$, слиједи да $g(x)$ дијели $\prod_{h(x) \in \Phi_r} h(x)$. Према томе, $g(x) = \prod_{h(x) \in \Delta} h(x)$, гдје је $\Delta \subseteq \Phi_r$.

Нека је $\hat{h}(x) \in \Delta$. Пошто је $\hat{h}(x) \in \Delta$ и $\Delta \subseteq \Phi_r$, онда је $\hat{h}(x) \in \Phi_r$. Затим, $\deg(\hat{h})$ дијели r. С обзиром на то да је $\hat{h}(x) \in \Delta$ и $g(x) = \prod_{h(x)\in\Delta} h(x)$, слиједи да $\hat{h}(x)$ дијели $g(x)$. Шта више, $\deg(\hat{h}) \leq deg(g)$, па је $\deg(\hat{h}) < r$. Слиједи да постоји прост број p који дијели r, такав да $\deg(\hat{h})$ дијели $\frac{r}{p}$. Из теореме 2.4.1 закључујемо да $\mathrm{GCD}(f(x), x^{q^{r/p}} - x) \neq 1$. □

Дакле, ако је r прост број, полином $P(x) \in \mathrm{GF}(2)[x]$ степена r је несводљив ако и само ако је $x^{2^r} \equiv x \pmod{P(x)}$. Ово нам даје једноставан тест несводљивости полинома, или примитивности, уз додатан услов да је r Мерсенов експонент.

Алгоритам 2.4.1: Тест несводљивости

```
A(x) ← x;
for j ← 1 to r do
    A(x) ← A(x)^2  mod P(x)
end for
if A(x) = x then
    return "несводљив"
else
    return "сводљив"
end if
```

Унутрашња петља алгоритма, која се извршава r пута, састоји се од два корака: квадрирања $A(x) \leftarrow A(x)^2$ и свођења $A(x) \leftarrow A(x) \mod P(x)$. Ако је $P(x) = x^r + x^s + 1$, оба корака се могу имплементирати у $O(r)$ битских операција, па је укупна временска сложеност теста несводљивости $O(r^2)$.

2.5 Просијавање и сертификати

Да би се могло предвидјети очекивано понашање алгоритма, неопходно је знати очекивану расподјелу степена несводљивих фактора. Процјене сложености се заснивају на следећој хипотези [5]:

Претпоставка 2.5.1. *За све тринoме облика $x^r + x^s + 1$ степена r над* $\mathrm{GF}(2)$*, вјероватноћа π_d да тринoм нема нетривијалан фактор степена $\leq d$ је највише $\frac{c}{d}$, гдје је c апсолутна константа и $1 < d \leq \frac{r}{\ln(r)}$.*

Важно је напоменути да исправност алгоритма не зависи од ове претпоставке, она само утиче на процјену времена извршавања алгоритма.

Прије примјене теста несводљивости, можемо уштедјети вријеме тако што ћемо прво провјерити да ли полином има неки мали фактор, тј. да ли је дјељив са несводљивим полиномом малог степена. Пошто несводљиви полиноми степена d дијеле $P_d(x)$, можемо провјерити да ли трином T има фактор степена d или неког његовог дјелиоца, рачунањем $\mathrm{GCD}(T, P_d)$. По аналогији са процесом просијавања малих цјелобројних фактора, овај процес се зове просијавање, иако се сито изводи рачунањем највећег заједничког дјелиоца.

Нека је $T(x) = x^r + x^s + 1$ и $2^d < r$. Дефинишемо

$$d' = 2^d - 1, \quad r' = r \bmod d', \quad s' = s \bmod d'.$$

Примјеном датих дефиниција на P_d добијамо

$$P_d = x^{2^d} - x = x^{d'+1} - x = x^{d'} \cdot x - x = x(x^{d'} - 1).$$

Како је $r' = r \mod d'$, можемо изразити $r = q \cdot d' + r', \quad q \in \mathbb{Z}$. Одатле добијамо да је $x^r = x^{q \cdot d' + r'} = (x^{d'})^q \cdot x^{r'}$. Захваљујући својству $(x^{d'})^q \mod (x^{d'} - 1) = 1$, долазимо до закључка да је $x^r \mod (x^{d'} - 1) = 1 \cdot x^{r'} = x^{r'}$. Узимајући ово у обзир добијамо једнакост $(x^r + x^s + 1) \mod (x^{d'} - 1) = x^{r'} + x^{s'} + 1$. С тога,

$$T = x^{r'} + x^{s'} + 1 \mod (x^{d'} - 1).$$

На основу свега наведеног, рачунање $\mathrm{GCD}(x^r + x^s + 1, x^{2^d} - x)$ може да се сведе на рачунање $\mathrm{GCD}(x^{r'} + x^{s'} + 1, x^{d'} - 1)$. [4] [2]

Узмимо за примјер трином $T(x) = x^7 + x^3 + 1$. Прво, треба да нађемо све вриједности d за које важи да је $2^d < r = 7$. Те вриједности су $d = 1$ и $d = 2$. За $d = 1$:

$$\begin{aligned} &d' = 2^1 - 1 = 1, \\ &r' = 7 \mod 1 = 0, \\ &s' = 3 \mod 1 = 0, \\ &P_1(x) = x^{2^1} - x = x(x - 1), \\ &(x^7 + x^3 + 1) \mod (x - 1) = 1 \end{aligned}$$

[4] $\mathrm{GCD}(a + m \cdot b, b) = \mathrm{GCD}(a, b), \quad m \in \mathbb{Z}$

Израчунавање $\mathrm{GCD}(x^7 + x^3 + 1, x^{2^1} - x)$ се своди на израчунавање

$$\mathrm{GCD}(1, x - 1) = 1.$$

За $d = 2$:

$$\begin{aligned} &d' = 2^2 - 1 = 3, \\ &r' = 7 \mod 3 = 1, \\ &s' = 3 \mod 3 = 0, \\ &P_2(x) = x^{2^2} - x = x(x^3 - 1), \\ &(x^7 + x^3 + 1) \mod (x^3 - 1) = x \end{aligned}$$

Израчунавање $\mathrm{GCD}(x^7 + x^3 + 1, x^{2^2} - x)$ се своди на израчунавање

$$\mathrm{GCD}(x, x^3 - 1) = 1.$$

За обе вриједности d које задовољавају услов $2^d < 7$, добијамо да је $\mathrm{GCD}(x^7 + x^3 + 1, x^{2^d} - x) = 1$. То значи да трином $x^7 + x^3 + 1$ нема мали фактор и може се наставити са тестом несводљивости.

Узимајући у обзир да је $2^d < r$, долази се до закључка да је горња граница за степен d једнака $\log_2(r)$. Ако се триноми који имају факторе степена мањег од $\log_2(r)$ искључе процесом просијавања, према претпоставци 2.5.1, остаје $\frac{r}{\log(r)}$ тринома за тестирање. Трошкови процеса просијавања су занемарљиви у односу на укупне трошкове. Стога, укупна сложеност потраге износи $\frac{r}{\log(r)} \cdot O(r^2) = O(\frac{r^3}{\log(r)})$.

Овдје уводимо и појам сертификата о сводљивости за све триноме који су тестирани. Ако је просијавањем утврђено да трином T има мали фактор, онда је сертификат о несводљивости управо тај фактор. У случају да постоји неколико фактора са истим степеном, чува се онај који је најмањи у лексикографском поретку. На примјер, између тринома $x^3 + x + 1$ и $x^3 + x^2 + 1$ бира се први трином. Ако не знамо фактор, али трином није прошао тест несводљивости, онда можемо забиљежити остатак $R(x) = (x^{2^r} - x) \mod T$. Уколико је остатак велики, може се забиљежити дио њега, нпр. $R(x) \mod x^{32}$. Сертификати треба да омогуће потврду сводљивости за много краће вријеме од времена утрошеног за утврђивање сводљивости. У овом раду сертификати нису реализовани.

2.6 Напреднији приступ потрази

Како је најзахтјевнији дио прорачуна тестирање несводљивости путем једначине (2.4), проширивањем процеса просијавања до већег степена може доћи до значајног смањења броја потребних тестова несводљивости. Циљ је идентификовати фактор најмањег могућег степена d, гдје је $d \leq \frac{r}{2}$, како би се додатно унапредила потрага.

За велико d, када је $2^d \gg r$, прво рачунамо $x^{2^d} \bmod T(x)$, а затим

$$\mathrm{GCD}(T(x), (x^{2^d} \bmod T(x)) - x).$$

На овај начин радимо са полиномима степена мањег од $2r$. Пошто провјеравамо потенцијалне степене d у растућем редоследу, $x^{2^d} \bmod T(x)$ можемо израчунати на основу резултата из претходног корака, тј. када на $x^{2^{d-1}} \bmod T(x)$ примјенимо једно додатно модуларно квадрирање.

Да би се смањио трошак приликом факторисања тринома израчунавањем НЗД, примјењује се техника позната као блокирање, која омогућава замјену израчунавања НЗД модуларним множењем. Суштина блокирања је одабрати параметар $\ell > 0$ и умјесто израчунавања

$$\mathrm{GCD}(T, x^{2^d} - x) \quad \text{za } d \in [d', d' + \ell),$$

НЗД се израчунава само једном

$$\mathrm{GCD}(T, p_\ell(x^{2^{d'}}, x)),$$

гдје је $p_\ell(X, x)$ интервални полином дефинисан као

$$p_\ell(X, x) = \prod_{j=0}^{\ell-1} (X^{2^j} - x).$$

Овом методом замјењујемо ℓ НЗД израчунавања једним НЗД израчунавањем и $\ell - 1$ модуларних множења. [6] Ако НЗД има степен једнак λd, при чему је $\lambda > 1$, и постоји потреба да се производ подјели на λ фактора степена d, тада се примјењује метода факторизације једнаког степена EDF (equal degree factorization). У ситуацијама када је EDF потребан, његова примјена је обично јефтина, јер укупан степен λd обично има малу вриједност када је $\lambda > 1$.

У циљу постизања још веће ефикасности у потрази, примјењује се други ниво блокирања. Идеја је замјенити већину модуларних множења модуларним квадрирањима, што у $\mathrm{GF}(2)[x]/P(x)$ убрзава израчунавање претходно наведеног интервалног полинома. Интервал $[d', d' + \ell)$ се дијели на $k \geq 2$ мањих интервала дужине m, таквих да на сваком од њих рачунамо

$$p_m(X, x) = \prod_{j=0}^{m-1} (X^{2^j} - x) = \sum_{j=0}^{m} x^{m-j} \cdot s_{j,m}(X),$$

гдје је

$$s_{j,m}(X) = \sum_{0 \leq k < 2^m, w(k)=j} X^k.$$

У овом изразу $w(k)$ представља Хемингову тежину. [5] Како је $s_{j,m}(X^2) = s_{j,m}(X)^2$, ако знамо вриједност $s_{j,m}(x^{2^{d-m}})$ за $0 < j \leq m$, онда вриједност $s_{j,m}(x^{2^d})$ можемо израчунати поступком сложености $m^2 S(r)$, гдје је $S(r)$ цијена модуларног квадрирања. На овај начин замјењујемо $m - 1$ модуларних множења и m модуларних квадрирања са m^2 модуларних квадрирања. Избор вриједности $m \approx \sqrt{\frac{M(r)}{S(r)}}$ (око 20 када је $\frac{M(r)}{S(r)} \approx 400$) резултира убрзањем за фактор око $\frac{m}{2}$, што је око 10 пута брже од убрзања са једним нивоом блокирања.

На примјер, ако је $m = 3$, рачунамо

$$s_{0,3}(X) = 1,$$

$$s_{1,3}(X) = X^4 + X^2 + X,$$

$$s_{2,3}(X) = X^6 + X^5 + X^3,$$

$$s_{3,3} = X^7,$$

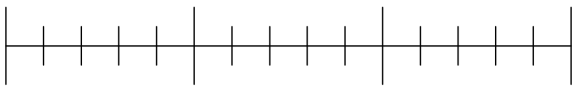
па добијамо да је

$$p_3(X, x) = x^3 + x^2(X^4 + X^2 + X) + x(X^6 + X^5 + X^3) + X^7.$$

Претпостављајући да знамо вриједности $s_{1,3}(x^{2^{d-3}})$, $s_{2,3}(x^{2^{d-3}})$ и $s_{3,3}(x^{2^{d-3}})$, сваку од њих квадрирамо $m = 3$ пута да бисмо добили вриједности $s_{1,3}(x^{2^d})$, $s_{2,3}(x^{2^d})$ и $s_{3,3}(x^{2^d})$, помоћу којих даље рачунамо $p_3(x^{2^d}, x)$.

Као илустрација стварних предности ових оптимизација, следећи примјер демонстрира колико се поступак убрзава од првобитне верзије до коначне.

[5] Хемингова тежина представља број јединица у бинарном запису броја. На примјер, бинарни запис броја 13 је 1101 и Хемингова тежина овог записа је 3, јер има три јединице.



Слика 2.1: $l = 15$, $m = 5$

Посматрајући подјелу интервала на слици 2.1 добијамо број корака потребних да се изврши сваки од три приступа:

- Број корака без блокирања: $15G + 15S$
- Број корака са првим нивоом блокирања: $G + 14M + 15S$
- Број корака са другим нивоом блокирања: $G + 2M + 75S$

Приликом примјене другог нивоа блокирања се најпре израчунава $p_m(X, x)$ за сваки подинтервал дужине m рачунањем m^2 модуларних квадрирања, а затим се производ из сваког поинтервала спаја у једну вриједност, чиме се штеде још два допунска множења.

2.7 Преглед до сада откривених тринома

Користећи технике и методе описане у претходним одјељцима, могуће је доћи до тражених примитивних тринома. У наставку је представљена табела [6] са примитивним триномима који су откривени примјеном наведених техника. У табели, цртица (-) у колони „године откривања" означава да година откривања за одређене вриједности r и s није позната. Занимљиво је напоменути да једина два случаја када је $r \equiv \pm 3 \pmod 8$ су за $r = 3$ и $r = 5$. Такође, једини трином који се појављује у табели са парним степеном r је за $r = 2$. Реципрочни триноми $x^r + x^{(r-s)} + 1$ нису приказани у табели.

[6] Доступно на https://maths-people.anu.edu.au/~brent/ftp/trinom/table.txt и [4]

Табела 2.2: Списак познатих примитивних тринома

r	s	године откривања
2	1	-
3	1	-
5	2	-
7	1,3	-
17	3, 5, 6	-
31	3, 6, 7, 13	-
89	38	-
127	1, 7, 15, 30, 63	-
521	32, 48, 158, 168	-
607	105, 147, 273	-
1279	216, 418	-
2281	715, 915, 1029	-
3217	67, 576	-
4423	271, 369, 370, 649, 1393, 1419, 2098	-
9689	84, 471, 1836, 2444, 4187	-
19937	881, 7083, 9842,	-
23209	1530, 6619, 9739	-
44497	8575, 21034	-
110503	25230, 53719	-
132049	7000, 33912, 41469, 52549, 54454	-
756839	215747, 267428, 279695	2000
859433	170340, 288477	2000
3021377	361604, 1010202	2000
6972593	3037958	2002
24036583	8412642, 8785528	2007
25964951	880890, 4627670, 4830131, 6383880	2007
30402457	2162059	2007
32582657	5110722, 5552421, 7545455	2008
42643801	55981, 3706066, 3896488, 12899278, 20150445	2009
43112609	3569337, 4463337, 17212521, 21078848	2008, 2009
74207281	9156813, 9999621, 30684570	2016

Глава 3

Програмска реализација алгоритама

Како би се постигла висока ефикасност у потрази за несводљивим триномима, коришћене су методе доступне у библиотеци NTL [1] (Number Theory Library), која пружа обимне ресурсе за рад са полиномима над пољем GF(2). Као тип података употребљена је класа GF2X, која имплементира полиномску аритметику по модулу 2. У имплементацији алгоритама примјењене су следеће методе:

- `GF2X()`: Иницијализује полином на 0.
- `void SetCoeff(GF2X& x, long i)`: Поставља коефицијент за члан X^i на 1. Грешка се јавља ако је $i < 0$.
- `void SetX(GF2X& x)`: Поставља x на моном X.
- `void GCD(GF2X& x, const GF2X& a, const GF2X& b)`: Рачуна НЗД за вриједности a и b и резултат поставља у x.
- `GF2X MulMod(const GF2X& a, const GF2X& b, const GF2X& f)`: Рачуна производ вриједности a и b по модулу полинома f.
- `GF2X SqrMod(const GF2X& a, const GF2X& f)`: Рачуна квадрат вриједности a по модулу полинома f.

[1] Доступна на https://libntl.org/download.html

- `GF2X PowerMod(const GF2X& a, long e, const GF2XModulus& f)`: Рачуна вриједност a^e по модулу полинома f. Обезбјеђена је аутоматска конверзија између типова GF2XModulus и GF2X помоћу оператора конверзије: `operator const GF2X& () const`.

Све имплементације су израђене у програмском језику C++. Коментари у коду обезбјеђују додатне информације о функционалностима и логици која стоји иза одређених дијелова.

3.1 Реализација алгоритма просијавања

У наставку је представљена реализација алгоритма просијавања, помоћу ког провјеравамо да ли трином $x^r + x^s + 1$ има мали фактор.

Код 4.1: Просијавање

```
#include <NTL/GF2X.h>

NTL::GF2X sieving(long r, long s) {
    //promjenljiva za smijestanje polinoma x^r+x^s+1
    NTL::GF2X T;
    //promjenljiva za smijestanje monoma x
    NTL::GF2X X;
    //promjenljiva za smijestanje polinoma x^(2^i) + x
    NTL::GF2X F;
    //promjenljiva za smijestanje izracunatog GCD
    NTL::GF2X G;

    SetCoeff(T, r, 1);
    SetCoeff(T, s, 1);
    SetCoeff(T, 0, 1);

    SetX(X);

    //petlja za racunanje GCD
    for(int i = 1; i<=r/2; i++) {
        //pri svakoj iteraciji se kvadrira x
```

```
        X = SqrMod(X, T);

        //smijestanje polinoma x^(2^i)+x
        F = X;
        SetCoeff(F, 1, 1);

        GCD(G, T, F);

        //na pronalazak malog faktora izlazimo iz petlje
        if (G != 1) break;
    }

    return G;
}
```

Приликом позива методе прослеђују јој се степени r и s за трином T. Ако прослеђени полином нема несводљиви фактор степена $i = 1, .., \frac{r}{2}$, као резултат се враћа вриједност 1, а у супротном се враћа најмањи пронађен фактор тринома.

На примјер, трином $x^4 + x + 1$ је несводљив, док трином $x^7 + x^2 + 1$ има фактор $x^2 + x + 1$:

```
sieving(4, 1) -> [1]
sieving(7, 2) -> [1,1,1]
```

3.2 Реализација алгоритма за први ниво блокирања

У овој секцији је представљена реализација алгоритма за први ниво блокирања, који нуди ефикаснији приступ у односу на алгоритам просијавања.

Код 4.2: Први ниво блокирања

```
#include <NTL/GF2X.h>

NTL::GF2X first_level_blocking(long r, long s) {
```

```
//promjenljiva za smijestanje polinoma x^r+x^s+1
NTL::GF2X T;
//promjenljiva za smijestanje monoma x
NTL::GF2X X;
//promjenljiva za smijestanje polinoma x^(2^i) + x
NTL::GF2X P_pom;
//promjenljiva za smijestanje proizvoda svih polinoma
//x^(2^i) + x
NTL::GF2X P;
//promjenljiva za smijestanje izracunatog GCD
NTL::GF2X G;

SetCoeff(T, r, 1);
SetCoeff(T, s, 1);
SetCoeff(T, 0, 1);

SetX(X);

//smijestamo prvi polinom x^(2^1)+x
P = SqrMod(X, T);
SetCoeff(P, 1, 1);

//petlja za racunanje proizvoda polinoma
for(int i = 1; i<=r/2 - 1; i++) {
    //pri svakoj iteraciji se kvadrira x
    X = SqrMod(X, T);

    //racunamo novi polinom x^(2^i)+x na osnovu
    //kvadrirane vrijednosti promjenljive X
    P_pom = X;
    SetCoeff(P_pom, 1, 1);

    //mnozimo prethodni rezultat sa novim x^(2^i)+x
    P = MulMod(P, P_pom, T);
}
```

```
    GCD(G, T, P);
    return G;
}
```

Слично као за претходни алгоритам, приликом позива методе прослеђују јој се степени r и s за трином T. Ако прослеђени полином нема несводљиви фактор степена $i = 1, .., \frac{r}{2}$, као резултат се враћа вриједност 1, а у супротном се враћа најмањи пронађен фактор тринома.

На примјер, трином $x^7 + x^3 + 1$ нема мале факторе, док трином $x^{13} + x + 1$ има фактор $x^5 + x^4 + x^3 + x + 1$:

```
sieving(7, 3) -> [1]
sieving(13, 1) -> [1,1,0,1,1,1]
```

3.3 Реализација алгоритма за други ниво блокирања

У наставку је представљена реализација алгоритма за други ниво блокирања, који даје додатни слој оптимизације у потрази за несводљивим триномима, смањујући вријеме извршавања.

Код 4.3: Други ниво блокирања

```
#include <NTL/GF2X.h>

int hammingWeight(int k) {
    int weight = 0;
    while (k>0) {
        weight += k & 1;
        k >>= 1;
    }

    return weight;
}
```

```
NTL::GF2X calculate_S_j_m(
    long j, long m, const NTL::GF2X& X_d,
    const NTL::GF2X& T) {

    NTL::GF2X new_S_j_m = NTL::GF2X();

    for(int k = 0; k < pow(2,m); k++) {
        if (hammingWeight(k) == j) {
            //racunamo X^k = (x^(2^1))^k za svako w(k) = j
            new_S_j_m += PowerMod(X_d, k, T);
        }
    }

    return new_S_j_m;
}

NTL::GF2X second_level_blocking(long r, long s, long m) {
    //promjenljiva za smijestanje polinoma x^r+x^s+1
    NTL::GF2X T;
    //promjenljiva za smijestanje monoma x
    NTL::GF2X X;
    //promjenljiva za smijestanje proizvoda rezultata iz
    //podintervala
    NTL::GF2X P;
    //promjenljiva za smijestanje sume proizvoda
    //x^(m-j) * S_{j,m}(x), za j = 0, .., m
    NTL::GF2X sum;
    //promjenljiva za smijestanje izracunatog GCD
    NTL::GF2X G;
    //niz za smijestanje izracunatih vrijednosti S_jm
    NTL::GF2X S_j_m[m+1];

    SetCoeff(T, r, 1);
    SetCoeff(T, s, 1);
    SetCoeff(T, 0, 1);
```

```
SetX(X);

//proizvod se postavlja na 1 na pocetku
SetCoeff(P, 0, 1);

//petlja za racunanje proizvoda vrijednosti iz
//pointervala
for (int i = 0; i < (r/2)/m; i++) {
    //suma se postavlja na nulu prije racunanja svakog
    //podinteravala
    sum = 0;

    //petlja za racunanje podintervala
    for (int j = 0; j <= m; j++) {
        //ako se racuna prvi podinterval
        if (i == 0) {
            if (j == 0)
                S_j_m[j] = 1;
            else
                S_j_m[j] = calculate_S_j_m(
                    j, m, SqrMod(X, T), T);
        }
        //ako se racunaju ostali podintervali na osnovu
        //vrijednosti izracunatih u prvom
        else {
            S_j_m[j] = PowerMod(
                S_j_m[j], pow(2, m), T);
        }

        sum += MulMod(
            PowerMod(X, m-j, T), S_j_m[j], T);
    }

    P = MulMod(P, sum, T);
```

```
    }

    GCD(G, T, P);
    return G;
}
```

Приликом позива методе прослеђују јој се степени r и s за трином T, као и вриједност r за дужину подинтервала. Ако прослеђени полином нема несводљиви фактор степена $i = 1, .., \frac{r}{2}$, као резултат се враћа вриједност 1, а у супротном се враћа најмањи пронађен фактор тринома.

На примјер, трином $x^7 + x + 1$ нема мале факторе, док трином $x^{13} + x^4 + 1$ има фактор $x^3 + x^2 + 1$:

```
sieving(7, 1) -> [1]
sieving(13, 4) -> [1,0,1,1]
```

3.4 Резултати реализација алгоритама

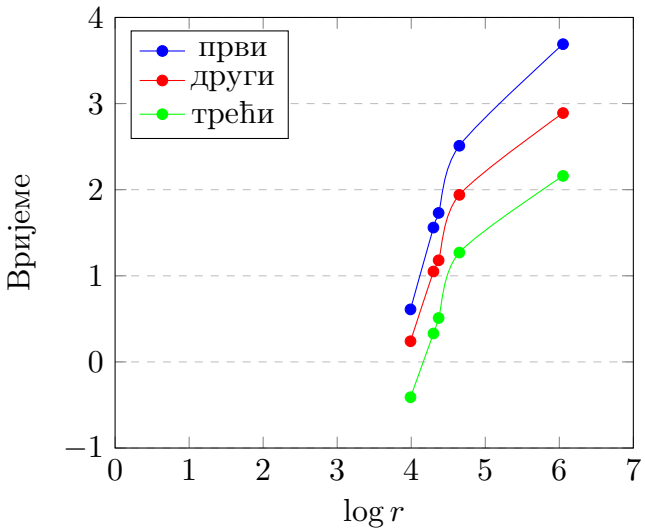
Ако погледамо реализацију првог алгоритма, видимо да се у њему врши $\frac{r}{2}$ GCD израчунавања и $\frac{r}{2}$ модуларних квадрирања, што алгоритму даје временску сложеност $O(\frac{r}{2} \cdot M(r) \log r + \frac{r}{2} \cdot r) = O(\frac{r}{2} \cdot (M(r) \log r + r))$. Други алгоритам се убрзава тако што се у њему замјењује $\frac{r}{2}$ GCD израчунавања са једним GCD израчунавањем и $\frac{r}{2} - 1$ модуларним множењем. То му даје временску сложеност $O(M(r) \log r + (\frac{r}{2} - 1) \cdot M(r) + \frac{r}{2} \cdot r) = O(M(r) \log r + \frac{r}{2} \cdot (M(r) + r))$. Трећи алгоритам још више добија на убрзању тако што дијели интервал на m подинтервала и на сваком од њих мијења $m-1$ модуларних множења и m модуларних квадрирања са m^2 модуларних квадрирања. Временска сложеност последњег алгоритма је $O(M(r) \log r + \frac{\frac{r}{2}}{m} \cdot m^2 \cdot r) = O(M(r) \log r + \frac{r^2}{2} \cdot m)$.

Узимајући у обзир GCD израчунавање, може се видјети да реализација првог алгоритма може бити најспорија, јер се у њој израчунава GCD у свакој итерацији петље. У реализацији другог и трећег алгоритма GCD се израчунава само једном, али трећи алгоритам може бити најбржи ако је вриједност m добро изабрана, тако да минимизује број операција.

Током извршавања програма, кориштена је метода clock() за прецизно мјерење временског интервала потребног алгоритмима. Као улазне вриједности алгоритама узимани су подаци из табеле 2.2. Резултати добијени овим путем дати су у табели 3.1, при чему се ознака t_1 односи на вријеме извршавања алгоритма просијавања, ознака t_2 на вријеме извршавања алгоритма са првим нивоом блокирања, а ознака t_3 на вријеме извршавања алгоритма са другим нивоом блокирања. У случајевима гдје је вријеме извршавања било изнимно брзо и резултирало бројевима са много децимала, за практичност је означено као $< 0.01s$. На слици 3.1 приказан је дијаграм зависности $\log_{10}$ времена извршавања у односу на $\log_{10} r$. У наставку, $\log_{10}$ ћемо означавати као log. За бољу прегледност и јаснођу, изабране су логаритамске вриједности последњих пет улазних величина из табеле 3.1 за r.

Табела 3.1: Вријеме извршавања алгоритама

r	s	t_1	t_2	t_3
2	1	<0.01s	<0.01s	<0.01s
3	1	<0.01s	<0.01s	<0.01s
5	2	<0.01s	<0.01s	<0.01s
7	1	<0.01s	<0.01s	<0.01s
17	3	<0.01s	<0.01s	<0.01s
31	3	<0.01s	<0.01s	<0.01s
89	38	<0.01s	<0.01s	<0.01s
127	1	<0.01s	<0.01s	<0.01s
521	32	<0.01s	<0.01s	<0.01s
1279	216	0.03s	<0.01s	<0.01s
2281	715	0.08s	0.04s	0.01s
3127	67	0.19s	0.09s	0.02s
4423	271	0.47s	0.22s	0.05s
9689	84	4.12s	1.73s	0.39s
19937	881	36.12s	11.21s	2.12s
23209	1530	53.54s	15.02s	3.27s
44497	8575	323.86s	86.98s	18.51s
110503	25230	4856.82s	777.20s	146.19s



Слика 3.1: Приказ логаритам времена извршавања сва три алгоритма у зависности од улаза $\log r$

Глава 4

Закључак

Примјеном описаних приступа изводљиво је пронаћи примитивне триноме веома великих степена. Основни принцип лежи у оптимизацијама које омогућавају ефикасну потрагу. Сумирајући идеје, најзначајније оптимизације које чине основу приступа су:

- Примјеном Свонове теореме концентришемо пажњу само на степене $r \equiv \pm 1 \pmod 8$. Ako je $r \equiv \pm 3 \pmod 8$, даље истраживање наставља се само за $s = 2$.
- Пошто је $x^r P(\frac{1}{x}) = x^r + x^{r-s} + 1$, разматрамо само $s \leq \frac{r}{2}$.
- Квадрирање полинома над GF(2) може бити урађено у линеарном времену, јер укрштени чланови у квадрату нестају: $(\sum_k a_k * x^k)^2 = \sum_k a_k * x^{2k}$.
- Свођење полинома степена мањег од $2r$ по модулу тринома $x^r + x^s + 1$ степена r такође може бити урађено у линеарном времену.
- Већина израчунавања НЗД може се замјенити модуларним множењима, користећи горе споменуту технику блокирања.
- Већина модуларних множења се може замјенити модуларним квадрирањима користећи други ниво блокирања.

Лов на триноме је резултирао побољшањем софтвера за операције на полиномима над GF(2). Са сталним напретком у области истраживања и континуираним напорима иницијатива као што је пројекат GIMPS у откривању

нових Мерсенових простих бројева, отвара се могућност даљег трагања за примитивним триномима. Тиме се уједно пружа прилика за тестирање и оптимизацију описаних алгоритама и приступа, као и за разматрање нових техника и стратегија.

Библиографија

[1] A. W. Bluher. A swan-like theorem. *Finite Fields and Their Applications*, 12(1):128–138, 2006.

[2] R. Brent, S. Larvala, and P. Zimmermann. A fast algorithm for testing reducibility of trinomials mod 2 and some new primitive trinomials of degree 3021377. *Mathematics of Computation*, 72(243):1443–1452, 2003.

[3] R. P. Brent, P. Gaudry, E. Thomé, and P. Zimmermann. Faster multiplication in gf (2)[x]. In *Algorithmic Number Theory: 8th International Symposium, ANTS-VIII Banff, Canada, May 17-22, 2008 Proceedings 8*, pages 153–166. Springer, 2008.

[4] R. P. Brent and P. Zimmermann. Twelve new primitive binary trinomials. *arXiv preprint arXiv:1605.09213*, 2016.

[5] R. P. Brent, P. Zimmermann, et al. The great trinomial hunt. *Notices of the AMS*, 58(2), 2010.

[6] R. P. Brent, P. Zimmermann, and LORIA Villers-les Nancy. A multi-level blocking distinct-degree factorization algorithm. *Contemporary mathematics*, 461:47–58, 2008.

[7] B. Hanson, D. Panario, and D. Thomson. Swan-like results for binomials and trinomials over finite fields of odd characteristic. *Designs, Codes and Cryptography*, 61(3):273–283, 2011.

[8] R. Lidl and H. Niederreiter. *Introduction to finite fields and their applications*. Cambridge university press, 1994.

[9] N. Lord. A first course in abstract algebra , by john b. fraleigh. pp 518.£ 17· 95. 1989. isbn 0-201-16847-2 (addison-wesley). *The Mathematical Gazette*, 73(466):349–351, 1989.

[10] J. Von zur Gathen and J. Gerhard. Arithmetic and factorization of polynomial over f2. In *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*, pages 1–9, 1996.

Биографија аутора

Јована Павловић је рођена 25.09.1996. у Братунцу. Основну школу је завршила 2011. године као вуковац. Исте године уписује средњу школу „Вук Караџић" у Љубовији, општи смијер. Гимназију завршава 2015. године као одличан ђак и исте године уписује Математички факултет Универзитета у Београду. Основне академске студије завршава 2020. године, а потом на истом факултету уписује и мастер студије, на смијеру Математика, модул Рачунарство и информатика.