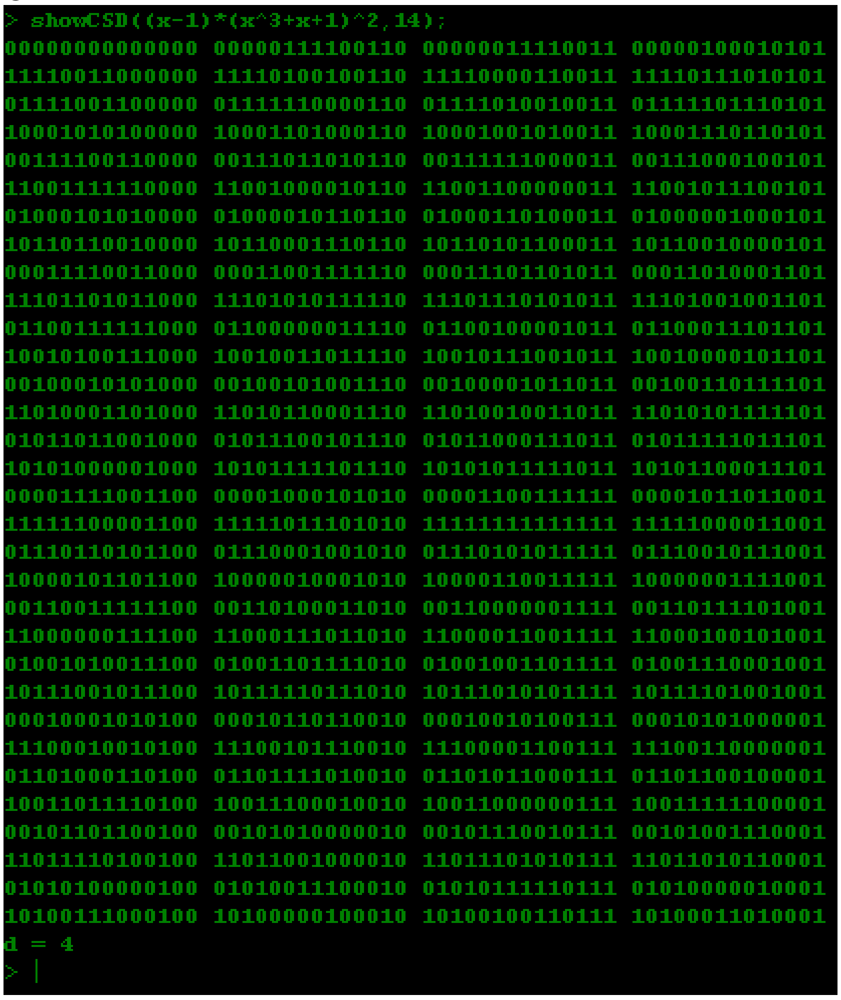
rijksuniversiteit groningen

faculteit Wiskunde en Natuurwetenschappen

Finding the Minimal Distance of Cyclic Self-Dual Codes



```
> showCSD((x-1)*(x^3+x+1)^2,14);
00000000000000 00000111100110 00000011110011 00000100010101
11110011000000 11110100100110 11110000110011 11110111010101
01111001100000 01111110000110 01111010010011 01111101110101
10001010100000 10001101000110 10001001010011 10001110110101
00111100110000 00111011010110 00111111000011 00111000100101
11001111110000 11001000010110 11001100000011 11001011100101
01000101010000 01000010110110 01000110100011 01000001000101
10110110010000 10110001110110 10110101100011 10110010000101
00011110011000 00011001111110 00011101101011 00011010001101
11101101011000 11101010111110 11101110101011 11101001001101
01100111111000 01100000011110 01100100001011 01100011101101
10010100111000 10010011011110 10010111001011 10010000101101
00100010101000 00100101001110 00100001011011 00100110111101
11010001101000 11010110001110 11010010011011 11010101111101
01011011001000 01011100101110 01011000111011 01011111011101
10101000001000 10101111101110 10101011111011 10101100011101
00001111001100 00001000101010 00001100111111 00001011011001
11111100001100 11111011101010 11111111111111 11111000011001
01110110101100 01110001001010 01110101011111 01110010111001
10000101101100 10000010001010 10000110011111 10000001111001
00110011111100 00110100011010 00110000001111 00110111101001
11000000111100 11000111011010 11000011001111 11000100101001
01001010011100 01001101111010 01001001101111 01001110001001
10111001011100 10111110111010 10111010101111 10111101001001
00010001010100 00010110110010 00010010100111 00010101000001
11100010010100 11100101110010 11100001100111 11100110000001
01101000110100 01101111010010 01101011000111 01101100100001
10011011110100 10011100010010 10011000000111 10011111100001
00101101100100 00101010000010 00101110010111 00101001110001
11011110100100 11011001000010 11011101010111 11011010110001
01010100000100 01010011100010 01010111110111 01010000010001
10100111000100 10100000100010 10100100110111 10100011010001
d = 4
> |
```

Bachelor's thesis, Mathematics

March 2009

Student: E.J.H. Brandenburg

Supervisor: Prof.dr. J. Top

Contents

Introduction

When advocating mathematics, one of the best subjects to discuss is its importance to securing data-transmission, especially when this concerns financial issues on the internet: not only do third parties need to be disabled from reading the data (for which RSA can be used) also does the data need to arrive as it was sent.
The latter can be done by enforcing certain restrictions on the admissible data, which will enable the receiver to detect errors and possibly correct them.
For exampel, few people will have problems reading this sentense, although it contains errors.

Two interesting vocabularies (or *codes* in mathematical terms) are the so-called *Cyclic* and *Self-Dual* ones. Their intriguing properties have been investigated extensively throughout the years; however, research on their combination is rather scarce. Nonetheless, an interesting result is a list of 2003 by *Carmen-Simona Nedeloaia* containing the *minimal distances* of all binary Cyclic Self-Dual (hence CSD for convenience) codes up to lengths of 120 digits.[2]
These *minimal distances* are important, because they give us information about the possibility that corrupted data can be restored. For this reason the aim of the thesis will be to improve this list and also to make an attempt to discover the properties of the *best* CSD's.
The first chapter will introduce the minimal distance and is followed by 3 chapters with basic theory concerning Cyclic, Self-Dual and CSD codes. Chapter 5 deals with the algorithms for finding minimal distances of CSD's and the next chapter contains the results of the research.

1 Minimal Distance

Definition 1.1. C is called a code if it is a linear subspace of $\mathbb{F}_q^n$.

EXAMPLE: $\{(0,0,0),(1,0,1)\} \subseteq \mathbb{F}_2^3$ is a code, while $\{(0,0),(1,0,1)\}$ ($\nsubseteq \mathbb{F}_2^n$ for any n) and $\{(1,0,1)\}$ (not linear) are *not*.
Note: since this thesis will restrict itself to binary codes, I will simply write $\mathbb{F}$ for $\mathbb{F}_2$.

Definition 1.2. The *weight* of a word x is the number of non-zero digits.

EXAMPLE: $wt(0,0,0) = 0 \quad wt(0,1,0) = 1 \quad wt(1,0,1) = 2 \quad wt(1,1,1) = 3$

Definition 1.3. The *minimal distance* of C is $d(C) = \min\{wt(u-v) : u,v \in C,\ u \neq v\}$.

Since we work with linear subspaces, we have $u, v \in C \Rightarrow u - v \in C$. Therefore we can rewrite the definition to $d(C) = \min\{wt(u) :\ u \in C,\ u \neq 0\}$.

EXAMPLE: If $C = \{(0,0,0),(1,0,1),(0,1,0),(1,1,1)\}$, then using the previous example, we find $d(C) = 1$.
Use this example to verify that the two definitions indeed give the same minimal distance for C.

When one reads a word that is 'not correct', i.e. not in C, one notices an 'error' and automatically replaces it by a 'correct' word that 'looks most like it'.
The *minimal distance* gives us an indication how many digits may go wrong, before the corrupted word will start to look more like another word. For example: consider $C = \{(0,0,0),(1,1,1)\},\ d(C) = 3$. If we receive $(1,0,0)$, then we notice an error and replace it by $(0,0,0)$, because that 'looks most like it'.
We should try to get the minimal distance as large as possible, without making our words to long.

2 Cyclic Codes

Definition 2.1. C is called *cyclic* if $(u_1, u_2, \ldots, u_{n-1}, u_n) \in C$ implements $(u_n, u_1, u_2, \ldots, u_{n-1}) \in C$.

Cyclic codes have a nice property, which is that a single word can already generate an entire code. For example: if C is a cyclic code generated by $(1,0,1,0)$, then C also contains $(0,1,0,1)$ because of the cyclic shift and $(0,0,0,0),(1,1,1,1)$ because of the linearity.
But there is more, because the cyclic shift has striking similarities with multiplying by x on a polynomial field. Compare for example $(0,0,1) \rightarrow (1,0,0)$ and $(0+0\cdot x+x^2) \bmod (x^3-1) \rightarrow x^3 \bmod (x^3-1) = (1+0\cdot x+0\cdot x^2) \bmod (x^3-1)$. This similarity will bring us to the following theorem:

Theorem 2.2. Under the isomorphism $\varphi(u_1,u_2,\ldots,u_n) = u_1+u_2x+\ldots+u_nx^{n-1}$, the cyclic shift σ on $\mathbb{F}^n$ corresponds to multiplication by x on $\mathbb{F}[x]/(x^n-1)$.

Proof:
$\varphi(\sigma(u_1,u_2,\ldots,u_{n-1},u_n))$
$= \varphi[(u_n,u_1,u_2,\ldots,u_{n-1})]$
$= u_n+u_1x+u_2x^2+\ldots+u_{n-1}x^{n-1}$
$= (u_1x+u_2x^2+\ldots+u_{n-1}x^{n-1}+u_nx^n) \bmod (x^n-1)$
$= x(u_1+u_2x+\ldots+u_{n-1}x^{n-2}+u_nx^{n-1}) \bmod (x^n-1)$
$= x\varphi[(u_1u_2\ldots u_{n-1}u_n)] \bmod (x^n-1)$
QED

Using this isomorphism, it is easy to see that $\varphi(C)$ (and therefore C) is an *ideal*, because since both multiplication and adding are defined, we have $f,g \in \varphi(C) \Rightarrow \gcd(f,g) \in \varphi(C)$. In other words: while we already stated that a cyclic code can be generated by a single word, we can even state that *every* cyclic code is generated by a single word.

Because the nice arithmetical properties of the polynomial field, the rest of the text will treat codes as subsets of $\mathbb{F}[x]/(x^n-1)$, where n is the length of the words. For this reason I shall introduce another notation:

Definition 2.3. Let f be a polynomial, then C_f is the cyclic code $(f\cdot\mathbb{F}[x])/(x^n-1)$.

Lemma 2.4. $f|(x^n-1) \Rightarrow \dim C_f = n-\deg(f)$.

Proof: Since $\forall g \in \mathbb{F}[x] : \ gf \in C_f$ and $\exists f^\perp : \ f\cdot f^\perp = x^n-1 = 0 \bmod (x^n-1)$, we have $C_f \cong F[x]/f^\perp$.
Because $f^\perp = \frac{x^n-1}{f} \Rightarrow \deg(f^\perp) = \deg(x^n-1)-\deg(f) = n-\deg(f)$, this

results in $\dim C_f = \dim(F[x]/f^{\perp}) = \deg(f^{\perp}) = n - \deg(f)$.
QED

3 Self-Dual Codes

Definition 3.1. If C is a code, then its dual is defined as $C^{\perp} = \{u : < u, v >= 0 \; \forall v \in C\}$.

From linear algebra we already know that $\dim C_f^{\perp} = n - \dim C_f$, but we can say more about dual codes by investigating the inner product.
Given $f = \alpha_0 + \alpha_1 x + ... + \alpha_{n-1}x^{n-1}$ and $g = \beta_0 + \beta_1 x + ...\beta_{n-1}x^{n-1}$ the inner product results in $< f, g >= \Sigma_{j=0}^{n-1}\alpha_j\beta_j$.
Compare this to the product in $\mathbb{F}[x]/(x^n - 1)$, which is given as $(f \cdot g) \bmod (x^n - 1) = \Sigma_{i,j=0}^{n-1}\alpha_{j \bmod n}\beta_{(i-j) \bmod n}x^i$.
The inner product makes both α_j and β_j go up, keeping the difference of their indices constant, while the other product makes β_{i-j} go down, keeping the sum of the indices constant.
For this reason I will introduce the following definition:

Definition 3.2. If $f = \alpha_0 + \alpha_1 x + ... + \alpha_m x^m,\ \alpha_m \neq 0$, then define its *reciprocal* by $f^* = \alpha_0 x^m + \alpha_1 x^{m-1} + ... + \alpha_m = x^{\deg(f)}f(\frac{1}{x})$.

Now one can see that $(f \cdot g) \bmod (x^n - 1) = \Sigma_{i=0}^{n-1} < f, x^{i-\deg(g)}g^* > x^i$, which will lead us to the following theorem:

Theorem 3.3. $g = \frac{x^n-1}{f} \Leftrightarrow C_f^{\perp} = C_{g^*}$.

Proof: Define $\tilde{f}$ as the generator of $C_f^{\perp}$. This means $< uf, v\tilde{f}^* >= 0 \; \forall u, v \in \mathbb{F}[x]$, because $uf \cdot vf^* = 0 \; \forall u, v \in \mathbb{F}[x]$. Using linearity of the inner product and $< x^a f, x^b \tilde{f} >=< x^{a-1}f, x^{b-1}\tilde{f} >$, this can be reduced to $< f, x^b\tilde{f} >= 0 \; \forall b \in \mathbb{Z}$.
Since $\Sigma_{i=0}^{n-1} < f, x^{i-\deg(\tilde{f}^*)}\tilde{f} > x^i = (f \cdot \tilde{f}^*) \bmod (x^n - 1)$, we need every coefficient of $(f \cdot \tilde{f}^*) \bmod (x^n - 1)$ to be equal to 0, which happens then and only then if $(x^n - 1)|(f \cdot \tilde{f}^*) \Rightarrow g|\tilde{f}^*$.
Thus we can say $C_f^{\perp} = C_{\tilde{f}} \subseteq C_{g^*}$.
Because $\dim C_f^{\perp} = n - \dim C_f = n - (n - \deg(f)) = n - \deg(g) = n - \deg(g^*) = \dim(C_{g^*})$, we have $C_{g^*} = C_f^{\perp}$.
QED

The importance of this theorem will become clear in the next chapter, when we will create CSD's. First, however, it is necessary to discuss the meaning of *self-dual*.

Definition 3.4. C is *self-dual* if $C = C^{\perp}$.

Proposition 3.5. If C is a *self-dual* code, then:

1. $2|n(C)$
2. $2|d(C)$

Proof:
1) $\dim C^{\perp} = n - \dim C \Rightarrow n = \dim C + \dim C^{\perp} = 2 \cdot \dim C$.
2) If $u \in C$, then $< u, v >= 0 \ \forall v \in C$, especially $< u, u >= 0$. Since $wt(u) \bmod 2 =< u, u >$, this means that $2|wt(u)$. Because u is arbitrary, we have $2|wt(v) \ \forall v \in C$, therefore $2|[d(C) = \min\{wt(v) : \ v \in C\}]$.
QED

4 CSD-Codes

4.1 Introduction

Definition 4.1.1. C is called *cyclic self-dual* (CSD) if it is both *cyclic* and *self-dual*.

Using what we found in the previous chapter, we can already tell a lot about the way CSD's look like: $C_f = C_f^{\perp} \Rightarrow f^* = \frac{x^n-1}{f}$. So if f is the generator of a CSD, then $f \cdot f^* = x^n - 1$.
Also we have $2|n$ and $2|d(C)$.
So just by using the previous chapters, we can already tell 3 things about CSD's:

- C_f is a CSD then and only then if $f \cdot f^* = x^n - 1$ (in $\mathbb{F}_2$).
- If C_f is a CSD, then $2|n$.
- If C_f is a CSD, then $2|d(C)$.

Now let's have a closer look.

4.2 Trivial CSD's

We have stated that if n is odd, then we don't have a CSD. But can we also say the opposite: if n is even, then does a CSD exist? The answer is "yes". In fact it is the code when not only $C_f = C_f^\perp$, but also $f = f^*$, which is the code generated by $f = \sqrt{x^n - 1} = x^{\frac{1}{2}n} + 1$.
This code is called a *repetition code* and has a very easy construction: all words can be written as $g \cdot (x^{\frac{1}{2}n} - 1)$, where $g \in \mathbb{F}[x]/\frac{x^n-1}{x^{\frac{1}{2}n}-1} = \mathbb{F}[x]/(x^{\frac{1}{2}n} - 1)$. Note that this means $\deg(g) < \frac{1}{2}n$, after which we can conclude that the trivial code just says to send the same word twice.
For example, the trivial code for $n = 6$ looks like:
$C = \{(0,0,0,0,0,0),(0,0,1,0,0,1),(0,1,0,0,1,0),(0,1,1,0,1,1),$
$(1,0,0,1,0,0),(1,0,1,1,0,1),(1,1,0,1,1,0),(1,1,1,1,1,1)\}$.
So the next question would be: "Are there also other CSD's than the 'trivial' ones?" Again the answer is "yes", but in order to find f, such that $f \neq f^*$ and $f \cdot f^* = x^n - 1$, one first needs to be able to factorize $x^n - 1$.

4.3 Factorizing $x^n - 1$

Define $n = 2^a b$, with b odd, then, since we are working on $\mathbb{F}_2$, we have $(x+y)^2 = x^2 + 2xy + y^2 = x^2 + y^2$, therefore we can rewrite $x^n - 1 = x^{2^a b} - 1 = (x^b - 1)^{2^a}$.
Now comes the tricky part: it is easy to see that $(x-1)|(x^b-1)$, but how to see that $x^5 - 1$ can not be further factorized than $(x-1)(x^4+x^3+x^2+x+1)$, while $(x^7-1) = (x-1)(x^3+x+1)(x^3+x^2+1)$?
This paragraph shall show that the factorization of $x^b - 1$ depends on the behaviour of 2 in $\mathbb{Z}/b\mathbb{Z}$.

First of all a little lemma:
Lemma 4.3.1. b is odd $\Leftrightarrow \exists \zeta \in \bar{\mathbb{F}}_2$, such that b is the smallest integer for which $\zeta^b - 1 = 0$.
Proof: If b is odd and such a ζ would not exist, then we would need that $\forall \zeta \in \bar{\mathbb{F}}\ \exists s < b$, such that s is the smallest integer for which $\zeta^s - 1 = 0$.
But since we have $x^b - 1 = 0$, we need $s|b$, so we can write $b = st$ and see that $\zeta^b - 1 = \zeta^{st} - 1 = (\zeta^s - 1)\Sigma_{i=1}^t \zeta^{s(t-i)} = 0$. Because b is odd, t is also odd, therefore if $\zeta^s - 1 = 0 \Rightarrow \Sigma_{i=1}^t \zeta^{s(t-i)} = \Sigma_{i=1}^t 1^{(t-i)} = 1$, which means there also exists a $\zeta :\ \zeta^s - 1 \neq 0$, for which $\zeta^b - 1 = 0$.
The other way around we see that if b is even, then we can choose $t = 2$ and see that $\zeta^b - 1 = 0 \Rightarrow \zeta^s - 1 = 0$. **QED**

Now, if ζ is a root of $x^b - 1 = 0$, then also $\zeta^2, \zeta^3, \ldots, \zeta^b$. Note that

$\zeta^{b+1} = \zeta^b \cdot \zeta = 1 \cdot \zeta = \zeta$. Therefore, thanks to the lemma, we can say that $x^b + 1 = \Pi_{i=0}^{b-1}(x - \zeta^i)$ for some ζ in the algebraic closure of $\mathbb{F}_2$.
The trick is to find out which combinations of roots in $\bar{\mathbb{F}}_2$ form an irreducible polynomial in $\mathbb{F}_2$.

Proposition 4.3.2. If $f|(x^b - 1)$ is an *irreducible polynomial* and ζ is a root of f, then $f = \Pi_{i=0}^{m-1}(x - \zeta^{2^i})$, where m is the smallest integer such that $2^m = 1 \bmod b$.

Proof: It is easy to see that $S = \{\zeta, \zeta^2, \zeta^4, \ldots, \zeta^{2^{m-1}}\}$ are all roots of f, because $f(\zeta) = 0 \Rightarrow f(\zeta^2) = f(\zeta)^2 = 0$. So knowing we need at least these roots, we now need to prove they are also sufficient.
Let us take a look at the coefficients of $\Pi_{i=0}^{m-1}(x - \zeta^{2^i}) = \alpha_0 + \alpha_1 x + \ldots + \alpha_m x^m$. We need to prove that $\alpha_i \in \mathbb{F}_2 \; \forall i \in \{0, 1, \ldots, m-1\}$.
To prove this, we need that $\alpha_i^2 = \alpha_i$, because $\alpha_i^2 = \alpha_i \Rightarrow \alpha_i^2 - \alpha_i = 0 \Rightarrow \alpha_i(\alpha_i - 1) = 0$. In other words: α_i can only be equal to its square if $\alpha_i = 0$ or if $\alpha_i = 1$, which are the elements of $\mathbb{F}_2$.
Take $\varphi : f \mapsto f^2$ and $\psi : x \mapsto x^2$, then we see that $\varphi(f) = f^2 = (\Sigma_{i=0}^m \alpha_i x^i)^2 = \Sigma_{i=0}^m \alpha_i^2 x^{2i}$ and $\psi(f) = \Sigma_{i=0}^m \alpha_i x^{2i}$. Now we see we need to prove $\varphi(f) = \psi(f)$. To do that, we need f's other notation:
$\varphi(f) = [\Pi_{i=0}^{m-1}(x - \zeta^{2^i})]^2 = \Pi_{i=0}^{m-1}(x^2 - \zeta^{2^{i+1}}) = \Pi_{i=0}^{m-1}(x^2 - \zeta^{2^i}) = \psi(f)$.
QED

Example: Take $x^7 - 1$ and a ζ, such that $\zeta^7 = 1$, $\zeta \neq 1$. Then according to the proposition the minimal polynomial f also contains ζ^2 and ζ^4. So the irreducible polynomial will become $f = \Pi_{i=0}^{2}(x - \zeta^{2^i})$, where we notice that $\zeta^{2^3} = \zeta^8 = \zeta^7 \cdot \zeta = \zeta$.
Factorizing gives $\Pi_{i=0}^{2}(x - \zeta^{2^i}) = \Sigma_{i=0}^{3}(\Sigma_{s \subset S,\, |s|=3-i} \Pi_{j \in s} \zeta^{2^j}) x^i$
$= (\Sigma_{s \subset S,\, |s|=3} \Pi_{j \subset s} \zeta^{2^j}) + (\Sigma_{s \subset S,\, |s|=2} \Pi_{j \in s} \zeta^{2^j}) x + (\Sigma_{s \subset S,\, |s|=1} \Pi_{j \in s} \zeta^{2^j}) x^2$
$+ (\Sigma_{s \subset S,\, |s|=0} \Pi_{j \in s} \zeta^{2^j}) x^3$
$= (\zeta^2 \cdot \zeta^4 \cdot \zeta^1) + (\zeta^2 \cdot \zeta^4 + \zeta^4 \cdot \zeta^1 + \zeta^1 \cdot \zeta^2) x + (\zeta^2 + \zeta^4 + \zeta^1) x^2 + x^3$
$= 1 + (\zeta^6 + \zeta^5 + \zeta^3) x + (\zeta^2 + \zeta^4 + \zeta^1) x^2 + x^3$.
Now we see $(\zeta^2 + \zeta^4 + \zeta^1)^2 = \zeta^4 + \zeta^8 + \zeta^2 = \zeta^2 + \zeta^4 + \zeta^1$ and of course $(\zeta^6 + \zeta^5 + \zeta^3)^2 = \zeta^{12} + \zeta^{10} + \zeta^6 = \zeta^6 + \zeta^5 + \zeta^3$. This means the coefficients of f are indeed elements of $\mathbb{F}_2$.

Note: We don't know, yet, what the irreducible polynomial looks like, but we know already enough to decide for which n a CSD can be found.

4.4 Non-Trivial CSD's

We needed to find $f: \ f \cdot f^* = x^n - 1, \ f \neq f^*$. From the previous paragraph we know $\sqrt[2^a]{x^n - 1}$ can be factorized into irreducible polynomials, which depend on $2^i \bmod b$.
Therefore we need to translate the *reciprocal* to the roots. For this reason define $S_i = \{\zeta^{2^j i \bmod b} : \ j \in \mathbb{Z}\}$ and f_{S_i} as the minimal polynomial of ζ^i, after which we can state the following theorem:

Theorem 4.4.1. $f^*_{S_i} = f_{S_{-i}}$

Proof: The roots of $f^*_{S_i}$ are the roots of $f_{S_i}(\frac{1}{x})$ and those are the x such that $\frac{1}{x} \in S_i \Rightarrow x \in S_{-i}$.
QED

To form a CSD, we need to split the roots of $x^n - 1$ in two groups, fixed under $x \mapsto x^2$, in such a way that if a certain root is placed in one group, then its inverse has to be placed in the other group.
This implies a non-trivial CSD can only exist if there is an i such that $S_i \neq S_{-i}$, which brings us to the most important theorem of this chapter:

Theorem 4.4.2. Non-trivial CSD's exist for $n = 2^a b \Leftrightarrow \forall l \in \mathbb{Z} : \ 2^l \neq -1 \bmod b$.

Proof: If there does not exists an l such that $2^l = -1 \bmod b$, then $S_1 \cap S_{-1} = \emptyset$, which means we can form a CSD by adding S_1 to f and S_{-1} to f^*.
The other way around we need an i such that $S_i \cap S_{-i} = \emptyset$, which means $\{\zeta^{2^j i \bmod b} : \ j \in \mathbb{Z}\} \neq \{\zeta^{-2^j i \bmod b} : \ j \in \mathbb{Z}\}$. Now if there would exist an l such that $2^l = -1 \bmod b$, then
$S_i = \{\zeta^{2^j i \bmod b} : \ j \in \mathbb{Z}\}$
$= \{\zeta^{-2^{j-l} i \bmod b} : \ j \in \mathbb{Z}\}$
$= \{\zeta^{-2^j i \bmod b} : \ j \in \mathbb{Z}\} = S_{-i}$
QED

Example: Now we finally know enough to construct the first CSD.
If $n = 3 \cdot 2^a$, then we find $2^1 = -1 \bmod 3$, which means there exist no CSD.
If $n = 5 \cdot 2^a$, then we find $2^2 = -1 \bmod 5$, which means again no CSD.
If $n = 7 \cdot 2^a$, then we find $< 2 >= \{2, 4, 1\}$. So we can find a CSD.
Let us take the smallest possibility, which is $n = 14$, then we have $x^{14} - 1 = (x^7 - 1)^2 = (f_{S_0} f_{S_1} f_{S_3})^2$.
$f_{S_0} = (x - \zeta^0) = x - 1$ is symmetric, so should be added to both f and its

reciprocal. But $S_1 \neq S_3 = S_{-1}$, so we can create a CSD by adding S_1 twice to f, after which we have to add S_3 twice to f^*. This results in $f = f_{S_0} f_{S_1}^2$. Since $(fg)^* = f^* g^*$, we get $f^* = (f_{S_0} f_{S_1}^2)^* = f_{S_0} f_{S_3}^2$ and we see $f \cdot f^* = f_{S_0} f_{S_1}^2 \cdot f_{S_0} f_{S_3}^2 = x^{14} - 1$, which means C_f (and of course C_{f^*}) is a CSD.

5 Algorithm

Being familiar with the basic properties of CSD's, time has come to find their minimal distances. However, calculating the weights of all words of a code will be very time-consuming, so this chapter will treat ways to safe work.

5.1 Equivalent Codes

The first way to safe work is by doing nothing at all: if we already know the minimal distance of the code, then it would be a waste of time to calculate the weights of the words.
In the previous chapter we saw there were two codes for $x^{14} - 1$, which were the codes generated by $f_{S_0} f_{S_1}^2$ and the one generated by its reciprocal $f_{S_0} f_{S_3}^2$. Now since $(fg)^* = x^{\deg(fg)}(fg)(\frac{1}{x}) = x^{\deg(f)} f(\frac{1}{x}) \cdot x^{\deg(g)} g(\frac{1}{x}) = f^* g^*$, we have that the reciprocal of every word in C_f is a multiple of f^* and therefore contained in C_{f^*}.
Also one can easily verify that $wt(f) = wt(f^*)$, which will lead us to say that if C_f contains a word of a certain weight, then C_{f^*} should also contain a word of that weight, after which we conclude $d(C_f) = d(C_{f^*})$.
This appears to be merely a special case of an even more general theorem:

Theorem 5.1.1. $\gcd(\alpha, n) = 1 \Rightarrow d(C_{f_{S_i}}) = d(C_{f_{S_{\alpha i}}})$

Proof: The words of $C_{f_{S_{\alpha i}}}$ can be written as $g(x) f_{S_{\alpha i}} = g(x) \Pi_{j=1}^{\# S_{\alpha i}} (x - \zeta^{2^j \alpha i})$, where g is an arbitrary polynomial.
Since the weight function only counts the different x's, we have $\forall \alpha : \gcd(\alpha, n) = 1 \Rightarrow wt[g(x) \Pi_{j=1}^{\# S_{\alpha i}} (x - \zeta^{2^j \alpha i})] = wt[g(x^\alpha) \Pi_{j=1}^{\# S_{\alpha i}} (x^\alpha - \zeta^{2^j \alpha i})]$. The latter appears to be a word in $C_{f_{S_i}}$, for it gives zero for every element of $S_i = \{\zeta^{2^r i} : r \in \mathbb{Z}\}$:
$g((\zeta^{2^r i})^\alpha) \Pi_{j=1}^{\# S_{f_{\alpha i}}} ((\zeta^{2^r i})^\alpha - \zeta^{2^j \alpha i})$
$= g(\zeta^{2^r \alpha i}) \Pi_{j=1}^{\# S_{f_i}} (\zeta^{2^r \alpha i} - \zeta^{2^j \alpha i})$
$= g(\zeta^{2^r \alpha i})(\zeta^{2^r \alpha i} - \zeta^{2^r \alpha i}) \Pi_{j=1,\, j \neq r}^{\# S_{f_i}} (\zeta^{2^r \alpha i} - \zeta^{2^j \alpha i})$
$= 0$
So if there exists a word of certain weight in $C_{f_{S_{\alpha i}}}$, there exists a word of

equal weight in $C_{f_{S_i}}$, from which it follows: $d(C_{f_{S_{\alpha i}}}) \geq d(C_{f_{S_i}})$.
Since $\exists \alpha^{-1}$ and $\gcd(\alpha^{-1}, n) = 1$, the process is reversible, which means $d(C_{f_{S_i}}) \leq d(C_{f_{S_{\alpha i}}})$.
Combining the two we get: $d(C_{f_{S_i}}) = d(C_{f_{S_{\alpha i}}})$.
QED

In the above proof we used the map $\varphi : \mathbb{F}[x]/(x^n - 1) \mapsto \mathbb{F}[x]/(x^n - 1) : x \mapsto x^\alpha$, where $\gcd(\alpha, n) = 1$. This happens to have some very interesting properties:

- φ is an homomorphism, since $\varphi(f + g) = \varphi(f) + \varphi(g)$ and $\varphi(fg) = \varphi(f)\varphi(g)$.
- Since $\gcd(a, n) = 1$, φ is invertible and therefore an *isomorphism*.
- $C_{f_{S_i}}$ is an ideal in $\mathbb{F}[x]/(x^n - 1)\mathbb{F}[x]$, so $C_{\varphi(f_{S_i})} = C_{f_{S_{\alpha i}}}$ is also an ideal, which means $f_{S_{\alpha i}} = \gcd(f_{s_i}(x^\alpha), x^n - 1)$.
- Also very important: this map preserves the weights.
- If $\alpha = -1$ then $\varphi(C_{f_{S_i}}) = C_{f_{S_{-i}}} = C_{f^*_{S_i}}$.

We call C_f and C_g *equivalent* if there exists such a $\varphi : \varphi(f) = g$. Replacing the *minimal polynomial* by *a product of minimal polynomials* in the above proof, it can be shown that $d(C_f) = d(C_g)$ if C_f and C_g are equivalent.

5.2 Product Codes

Define C_f^n as the code generated by f in the n^{th}-dimension. If we know the minimal distances of $C_{f_1}^n, C_{f_2}^n$, what can we say about the minimal distance of $C_{f_1 f_2}^{2n}$?
Again we start with the 'trivial' case: $f_1 = f_2$. Then we can easily state the following theorem:

Proposition 5.2.1. $d(C_{f^2}^{2n}) = d(C_f^n)$

Proof: The words of $d(C_{f^2}^{2n})$ can be described as gf^2, $g = \beta_0 + \beta_1 x + \ldots + \beta_n x^n$, these can be split into an *odd* and *even* part: $(\beta_0 + \beta_2 x^2 + \ldots)f^2 + (\beta_1 x + \beta_3 x^3 + \ldots)f^2$.
This can be rewritten to $[(\beta_0 + \beta_2 x + \ldots)f]^2 + x[(\beta_1 + \beta_3 x + \ldots)f]^2$: a sum of 2 words in C_f^n.
Since both words live separately (one on the even places, the other on the

odd) they do not influence each other's weight. So the smallest word for $2n$ can be formed by choosing one of them to be 0 and the other the smallest word of f.
QED

Again things get a little more complicated if we look at 'non-trivial' cases. Before starting an investigation on such codes, we can, however, already state the following:

Proposition 5.2.2. $d(C^{2n}_{f_1 f_2}) \leq 2\min\{d(C^n_{f_1}), d(C^n_{f_2})\}$

Proof: Define f'_i as the word of smallest weight in $C^n_{f_i}$, then since $\deg(f'_i) < n$, we have $wt[(x^n-1)f'_i] = 2wt(f'_i)$. Because f'_i is a multiple of f_i and $x^n - 1 = f_{3-i} \cdot f^{\perp}_{3-i}$, $(x^n-1)f'_i$ is contained in $C^{2n}_{f_1 f_2}$, therefore $d(C^{2n}_{f_1 f_2}) \leq 2wt(f'_i)$.
QED

So if we have a CSD for $2n$, it is never better than a CSD for n. Nevertheless, we want to be able to calculate the exact minimal distance, for which reason we need a theorem by Van Lint.[1]

Theorem 5.2.3. If n is odd and $(fg)|(x^n-1)$, then $C^{2n}_{f^2 g} \cong \{(x^n-1)u + v : u \in C^n_f,\ v \in C^n_{fg}\}$.

Proof: Again we use the trick of splitting in odd and even parts: $v = v_{\text{even}} + v_{\text{odd}} = v_e + v_o$. Next we easily see that $(x^n-1)u + v \cong (x^n-1)u + v_o + x^n v_e$. Now we need to prove that the right handside of the equation symbol is just another way for writing the elements of $C^{2n}_{f^2 g}$. Since the dimensions are equal, this means $\forall u, v:\ (f^2 g)|[(x^n-1)u + v_o + x^n v_e]$.
First we see that $(fg)|(x^n-1)$ and $u \in C^n_f \Rightarrow f|u$, so we have $(f^2 g)|[(x^n-1)u]$.
Next we see that $v_o + x^n v_e = v_o + v_e - v_e + x^n v_e = v + (x^n-1)v_e$ and since $(fg)|v$ and $(fg)|(x^n-1)$, we have $(fg)|[v_o + (x^n-1)v_e]$.
Finally we will have to prove that not only $f|(v_o + x^n v_e)$, but also $f^2|(v_o + x^n v_e)$. Therefore we notice that $v_o + x^n v_e$ only contains odd powers of x, so we can write $v_o + x^n v_e = xw$, where w only contains even powers of x and where we also see that $\gcd(fg, x) = 1 \Rightarrow (fg)|w$.
Now if $f^2|(xw)$, then $f|\frac{d}{dx}(xw)$. This just so happens to be the case: $\frac{d}{dx}(xw) = x'w + w'x = w + 0 \cdot x = w$.
QED

The importance of Van Lint's theorem will become clear with the following lemma:

Lemma 5.2.4. $d(\{(x^n-1)u+v:\ u\in C_f^n,\ v\in C_g^n\}) = \min\{2d(C_f^n), d(C_g^n)\}$, where f,g voluntary.
Proof: This is a sum of $v-u$ and $x^n u$. Now every x which u eliminates from v below the x^n is added above, so $wt[(x^n-1)u+v]\geq wt(v)$. **QED**

So we can use this Van Lint's theorem to say $d(C_{f^2g}^{2n}) = \min\{2d(C_f^n), d(C_{fg}^n)\}$.
It has, however, one flaw: what if n is even?
Luckily one can generalize his theorem with a small adaption.

Theorem 5.2.5. Write $n=2^a b$, where b is odd, and factorize $f|(x^{2n}-1)$ to $f=f_1^{2^{a+1}}f_2^{2^{a+1}-1}\cdots f_{2^{a+1}}$, where $i\neq j\Rightarrow \gcd(f_i,f_j)=1$. (Note that we allow $f_i=1$) Next define $\tilde{n}=2^a$, $F_i=f_{1+i\tilde{n}}^{\tilde{n}}f_{2+i\tilde{n}}^{\tilde{n}-1}\cdots f_{\tilde{n}+i\tilde{n}}$, for $i\in\{0,1\}$, and $G=\Pi_{i=1}^n f_i$, so that we have $f=F_0\cdot G^{\tilde{n}}\cdot F_1$.
Then $C_f^{2n}\cong\{(x^n-1)u+v:\ u\in C_{F_0}^n,\ v\in C_{G^{\tilde{n}}\cdot F_1}^n\}$.

Proof: Just as before we see that $(x^n-1)u+v\cong(x^n-1)u+v_o+x^n v_e$ and $f|[(x^n-1)u]$. Also it is easy to see that $(G^{\tilde{n}}F_1)|[v_o+x^n v_e = v+(x^n-1)v_e]$.
But how do we prove it is also divisible by $F_0 G^{\tilde{n}} F_1$?
To see that, we first need to notice that $F_0|G^{\tilde{n}}$, after which we can conclude that it is sufficient to prove that $G^{2\tilde{n}}|(v_o+x^n v_e)$.
Again we look at $\frac{d}{dx}(v_o+x^n v_e)=v_o'+nx^{n-1}v_e+x^n v_e$. If n is odd, then we can use the proof of above, else we are stuck with $\frac{1}{x}v_o$.
Now we need a trick: rewrite $v=G^{\tilde{n}}w$, then $v_o=(G_o^{\tilde{n}}w_e+w_o G_e^{\tilde{n}})$. But since $\tilde{n}$ is even, we have $G_o^{\tilde{n}}=0$ and $G_e^{\tilde{n}}=G^{\tilde{n}}$, therefore $v_o=w_o G^{\tilde{n}}$. After which we conclude v_o and therefore $\frac{d}{dx}(v_o+x^n v_e)$ are multiples of $G^{\tilde{n}}$.
QED

Using recursion this theorem will lead us to:

$$C_f^{2n}=\Sigma_{i=1}^{2^{a+1}}(x^b-1)^{2^{a+1}-i}u_i:\ u_i=\Pi_{j=1}^i f_i$$

Furthermore combining recursion with $d(C_{F_0GF_1}^{2n})=\min\{2d(C_{F_0}^n), d(C_{GF_1}^n)\}$, we get:

$$d(C_f^{2n})=\min\{wt[(x^b-1)^{2^{a+1}-i}]\cdot d(C_{u_i}^b):\ 1\leq i\leq 2^{a+1}\}$$

6 Results

The goal of this thesis was to find the properties of the *best* CSD's and without even calculating a code, we can already say what property a good CSD should *not* have:

Proposition 6.1. $\exists g \in \mathbb{F}[x]/(x^b-1), g \neq 0 : \ \forall f : \ f \cdot f^* = x^{2^{a+1}b} - 1 \Rightarrow d(C_f^{2^{a+1}b}) \leq 2d(C_g^b)$.

Proof: Since $(x-1)^* = x-1$, we have $\gcd((x-1)^{2^{a+1}}, f) = (x-1)^{2^a}$. Now if u_i is defined as in the end of the previous section, then $u_{2^a} | (\frac{x^b-1}{x-1})$, which means $u_{2^a} \neq 0$. Therefore $C^b_{u_{2^a}}$ contains a non-zero word, for which reason $d(C^b_{u_{2^a}}) > 0$.
So we conclude:
$d(C_f^{2^{a+1}b}) \leq \min\{wt[(x^b-1)^{2^{a+1}-i}] \cdot d(C^b_{u_i}) : \ 1 \leq i \leq 2^{a+1}\}$
$\leq wt[(x^b-1)^{2^{a+1}-2^a}] \cdot d(C^b_{u_{2^a}}) = wt[(x^b-1)^{2^a}] \cdot d(C^b_{u_{2^a}}) = 2d(C^b_{u_{2^a}})$.
QED

This proposition is rather surprising, since the upper limit for the minimal distances appears to be independent of a. So it doesn't just say that the code won't improve much after doubling many times, but even that it won't improve at all.
For example, take $b = 7$, then for all $a \in \mathbb{N}$ and $f : \ f \cdot f^* = x^{7 \cdot 2^a} - 1$, we have $d(C_f^{7 \cdot 2^a}) \leq 2d(C^7_{x^3+x+1}) \leq 6$.
Indeed we see that if we take the sequence $n = 14, 28, 56, 112, \ldots$ we get more and more codes with a minimal distance of 6, but never one with a minimal distance greater than 6.

Now let's turn our attention to a property we would expect a good code to have: a high BCH-bound. The BCH-bound says that if we have a code C_f^b, b odd, and a ζ such that $\text{ord}(\zeta) = b$ and all $\zeta^{\alpha+1}, \zeta^{\alpha+2}, \zeta^{\alpha+3}, \ldots, \zeta^{\alpha+\delta}$ are roots of f, then $d(C_f^b) > \delta$.
It is this BCH-bound that is responsible for the best result now known for an open problem on CSD's: one would like to have a sequence of codes, such that both $\frac{\dim(C_i)}{n_i}$ and $\frac{d_i}{n_i}$ converge to a non-zero constant. Since we are working with CSD's, we have $\forall i \in \mathbb{Z} : \ \frac{\dim(C_i)}{n_i} = \frac{1}{2}$. But a sequence for which $\lim_{i \to \infty} \frac{d_i}{n_i} > 0$ has, for now, only been found for the *non*-cyclic case.
In fact, the best known limit for CSD's says for every δ there exists $n < 4\delta^2 - 2$ such that for a well-chosen f: $d(C_f^n) \geq \delta$.[4] This sequence chooses n such that $x^n - 1$ has many factors, after which one chooses his f as a product of the min-

imal polynomials of $\zeta^1, \zeta^2, \zeta^3, \dots, \zeta^{\delta-1}$. As a limit it has $\liminf_{i\to\infty} \frac{d_i}{\sqrt{n_i}} = 1$.

That this limit is a little rude may become clear if we take $\delta = 6$, after which we have $n < 4 \cdot 6^2 - 2 = 142$. It turns out, however, that we already have a code with $d = 6$ for $n = 30$ and that for $n = 126 < 142$ we even have a code such that $d = 14$, while $\delta = 14$ would require an upper limit of $4 \cdot 14^2 - 2 = 782$.

So let's investigate the influence of a high BCH-bound by looking at the following list, which has been found using Magma (see Appendix). It lists the best minimal distance for every length and also the number of asymmetric pairs, or in symbols: $A = \{S_i \cup S_{-i} : \ S_i \neq S_{-i}\}$; a high $\#A$ means we can construct a high BCH-bound.
Note: we only look at $n \bmod 4 = 2$ for the reason given earlier in this chapter.

n	$\#A$	d_{opt}	n	$\#A$	d_{opt}	n	$\#A$	d_{opt}
14	1	4	126	5	14	206	1	20
30	1	6	138	2	12	210	6	18
42	2	8	142	1	12	222	1	6
46	1	8	146	4	18	230	2	16
62	3	10	150	2	6	234	4	18
70	2	8	154	2	8	238	3	14
78	1	6	158	1	16	246	2	6
90	2	8	170	4	10			
94	1	12	174	1	6			
98	2	6	178	4	20			
102	2	6	182	4	14			
110	1	10	186	6	20			
			190	1	10			

In the table we see that if for some n we have a relative low d_{opt}, then $\#A$ is also low. Take, for example, $n = 174$. However, the opposite is not true. Compare, for example, $n = 158$, which gives $d_{\text{opt}} = 16$, and $n = 170$, which gives $d_{\text{opt}} = 10$.
So although a high $\#A$ ensures us of at least a reasonable d_{opt}, which is something we like during these times of economic crisis, the n's with low values for $\#A$ may, although risky, give a better result. This is nicely illustrated by the fact that in the list of Nedeloaia, $n \leq 120$, the best result is obtained for $n = 94$ with $\#A = 1$!

However, a high $\#A$ does not just assure us that our d_{opt} won't be to low, it even seems to increase our chances of finding a good minimal distance, as is nicely illustrated by the next sequence:

$n_i - n_{i-1}$		16	16	16	32	32	32	48
n_i	14	30	46	62	94	126	158	206
d_i	4	6	8	10	12	14	16	20

It turns out in our table that, with a few exceptions, $\forall n < n_i : \ d(C^n) < d_i$ and that those exceptions $(46, 146, 178, 186)$ have a relative high $\#A$.
So like we concluded that if d_{opt} is relatively low, then the same goes for $\#A$, we now expect that if d_{opt} is relatively high, then again the same goes for $\#A$. However, I am not sure whether that is really true, leave alone how to prove it.
Finally I would like to spend a few words on the sequence of the last table. Based on the results on d_{opt}, I expect that the following sequence will give an improving d_{opt}:

$$n_i = n_{i-1} + 16 \cdot \lceil \frac{i-1}{3} \rceil, \text{ and } n_0 = 14 \Rightarrow n_i = 16\lceil \frac{i}{3} \rceil i + 14$$

Concerning d_{opt} I wish to make another bold statement: $d_i \geq 2\lceil \frac{i}{6} \rceil i + 4$. We then have $\frac{d_i}{n_i} \geq \frac{2\cdot\frac{i}{6}\cdot i+4}{16(\frac{i}{3}+1)i+14} = \frac{\frac{1}{3}i^2+4}{\frac{16}{3}i^2+16i+14} \rightarrow \frac{1}{16} > 0$.
But it needs to be said that this expectation is merely based on the calculations we did for $n \leq 246$ and has no theoretic foundation. So it may very well be that there exists a large n_i for which this doesn't work.

7 Conclusion

Now let's summarize what has been achieved:

- We have extended the list of Nedeloaia with *minimal weights of CSD's* from $n \leq 120$ to $n \leq 246$. This turns out to be relatively easy thanks to MAGMA and Theorem 5.2.5.

- The minimal distance of a CSD with length $2^a b$ has an upper bound of twice the minimal distance of a certain code with length b. So it is independent of a.

- We may have found a sequence of CSD's for which $\lim_{i\to\infty} \frac{d_i}{n_i} > 0$.

Biblography

1. J.H. van Lint, "Repeated-Root Cyclic Codes", *IEEE Transactions on Information Theory*, vol.37, no.2 pp..343-345, March 1991.

2. Carmen-Simona Nedeloaia, "Weight Distributions of Cyclic Self-Dual Codes", *IEEE Transactions on Information Theory*, Vol.49, no.6, JUNE 2003.

3. Bas Heijne, "Cyclic Self-Dual Codes", Master's Thesis Rijks*universiteit* Groningen, 7 MAY 2007,
http://scripties.fwn.eldoc.ub.rug.nl/FILES/scripties/Wiskunde/Masters/2007/Heijne.B./Bas_Heijne_doctoraal_WM_2007.pdf

4. Bas Heijne and Jaap Top, "On the Minimal Distance of Self-Dual Cyclic Codes", Submitted for publication, 2008.

Appendix: Magma

Magma proved to be ideal for calculating the minimal distances of the CSD's: the demo-version (*http://magma.maths.usyd.edu.au/calc/*) allowed only 20 seconds, but that was enough for all cases, except $n = 206$, where we needed a few more seconds.
I entered the text below to find d_{opt}, with one exception: I also used the results of paragraph 5.1, but I manually found out which codes did not need to be calculated. To explain how that can be done is complicated and not necessary, since the program below works as well.

```
P<x> := PolynomialRing(FiniteField(2));

/* divides the factors of x^b-1 in 2 groups
 * Ls = a list of all symmetric factors
 * La = a list of all asymmetric pairs
 */
SplitFactors := function(b)
 Ls:=[]; La:=[]; F:=Factorization(x^b+1);
 done:=[]; for i:=1 to #F do Append(~done,false); end for;
 for i:=1 to #F do
  if not done[i] then
   g:=GCD(Evaluate(F[i][1],x^(b-1)),x^b-1); done[i]:=true;
   if F[i][1] eq g then
    Append(~Ls,F[i][1]); done[i] := true;
   else
    index:=i;
    for j:=index+1 to #F do
     if g eq F[j][1] then index := j; end if;
    end for;
    done[index] := true;
    for j:=1 to F[i][2] do
     Append(~La,[F[i][1],g]);
    end for;
   end if;
  end if;
 end for;
 return Ls, La;
end function;

// finds the minimal distance of a cyclic code of length ‘b’, ‘b’ odd.
dC:=function(Ls,La,s,v1,v2,b)
 f:=1; for i:=1 to #Ls do f:=f*Ls[i]^s; end for;
 for i:=1 to #La do
  f:=f*La[i][1]^v1[i]*La[i][2]^v2[i];
 end for;
 C:=CyclicCode(b,f);
 return MinimumWeight(C);
end function;
```

```
/* calculates the minimal distance of a CSD of length ‘n=2b’, ‘b’ odd
 * v in F_3^#La = v chooses which one from each asymmetric pair
 *  should be contained in the CSD
 *  0 = second twice; 1 = both once; 2 = first twice
 */
dCSD:=function(Ls,La,v,b)
 v1f:=v; for i:=1 to #v1f do v1f[i]:=v1f[i]*(v1f[i]-1)/2; end for;
 v2f:=v; for i:=1 to #v1f do v2f[i]:=(v2f[i]-1)*(v2f[i]-2)/2; end for;
 v1g:=v; for i:=1 to #v1g do v1g[i]:=v1g[i]-v1f[i]; end for;
 v2g:=v; for i:=1 to #v1g do v2g[i]:=2-(v2g[i]+v2f[i]); end for;
 df:=2*dC(Ls,La,0,v1f,v2f,b);
 dg:=dC(Ls,La,1,v1g,v2g,b);
 return Minimum(df,dg),"@",df,dg,"@",v;
end function;

// CHANGE THE VALUE OF ‘n’ TO FIND THE OPTIMAL DISTANCE OF OTHER ‘n’.
n:=42;
Ls,La:=SplitFactors(n div 2);
for i:=0 to 3^#La-1 do
 v:=[]; res:=i;
 for j:=1 to #La do Append(~v,res mod 3); res:=res div 3; end for;
 dCSD(Ls,La,v,n div 2);
end for;
```