Searching for Nonlinear Feedback Shift Registers with Parallel Computing

Przemysław Dąbrowski, Grzegorz Łabuzek,
Tomasz Rachwalik, Janusz Szmidt

Military Communication Institute
ul. Warszawska 22A, 05-130 Zegrze, Poland
j.szmidt@wil.waw.pl

Abstract. Nonlinear feedback shift registers (*NLFSRs*) are used to construct pseudorandom generators for stream ciphers. Their theory is not so complete as that of linear feedback shift registers (*LFSRs*). In general, it is not known how to construct all *NLFSRs* with maximum period. The direct method is to search for such registers with suitable properties. Advanced technology of parallel computing has been applied both in software and hardware to search for maximum period *NLFSRs* having a fairly simple algebraic normal form.

Key words: nonlinear feedback shift registers, maximum period, quadratic m-sequences, parallel computing, FPGA implementation.

1 Introduction

Feedback shift register (*FSR*) sequences have been widely used in many areas of communication theory, as key stream generators in stream ciphers cryptosystems, pseudorandom number generators in many cryptographic primitives, in the design of correlators for spread spectrum communication systems, global positioning systems or radar systems, and as testing vectors in hardware design. Golomb's book [6] is a pioneering one that discusses this type of sequences. A modern treatment of the subject is contained in the book of Golomb and Gong [7].

The theory of linear feedback shift registers (*LFSRs*) is quite well understood. In particular, it is known how to construct *LFSRs* with maximum period; they correspond to primitive polynomials over the binary field $\mathbb{F}_2$. The order n of *FSR* is the number of its cells and an *FSR* can generate a binary sequence of period up to 2^n. The main drawback of primitive *LFSRs* is that their linear complexity is equal to their order. In recent years, nonlinear feedback shift registers (*NLFSRs*) have received much attention in designing numerous cryptographic algorithms such as stream ciphers and lightweight block ciphers to provide security in communication systems. In most cases, *NLFSRs* have much greater linear complexity than *LFSRs* of the same period. However, there are no general methods of designing maximum period *NLFSRs*. The construction of a special class of *NLFSRs* with maximum period has been given by Mykkeltveit *et al.* [11]. Recently, maximum period *NLFSRs* of order up to $n = 64$ have been constructed in the paper of Mandal and Gong [9], but these *NLFSRs* have very complicated algebraic normal form (*ANF*). Dubrova [3] has given an example of a Galois shift register of order $n = 100$ generating a sequence with maximum period but this sequence does not have the de Bruijn property; i.e., some patterns of n-bits appear more than once in the sequence.

In this article the approach from the papers of Gammel, Goettfert, and Kniffler [5] and Chan, Games, and Rushanan [2] is followed. *NLFSRs* having a simple algebraic normal form and maximum period are directly found by exhaustive searching. In particular, the conjecture posed in [2] on the existence of maximum period *NLFSRs* whose feedback functions have linear terms and one quadratic term has been experimentally investigated. Like the notion of linear m-sequence, these *NLFSRs* generate quadratic m-sequences. The conjecture has been verified up to order $n = 29$. Quadratic m-sequences of this type up to order $n = 21$ have been classified according to the weight (number of terms) of primitive polynomials involved. These experimental investigations support the Chan, Games and Rushanan conjecture of the existence of a special type of quadratic m-sequences for each order n.

Our experiments have applied parallel computing. *NLFSRs* generating quadratic m-sequences have been found by using three servers with 54 CPU cores. The investigations from the previous paper [12] have also been continued using the Field Programmable Gate Arrays to find maximum period *NLFSRs*. Nonlinear feedback shift registers have been applied in [13] to construct modified alternating step generators.

2 Feedback Shift Registers

In this section, the definitions and basic facts about feedback shift registers are given. We use $\mathbb{F}_2$ to denote the binary finite field. $\mathbb{F}_2[x]$ denotes the ring of polynomials in the indeterminate x and with coefficients from $\mathbb{F}_2$. Let $\mathbb{F}_2^n$ be the n-dimensional vector space over $\mathbb{F}_2$ consisting of all n-tuples of elements of $\mathbb{F}_2$. Any function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ is referred to as a *Boolean function* of n variables. A sequence of elements $\mathbf{s} = (s_0, s_1, \ldots)$ of $\mathbb{F}_2$ is called a *binary sequence.* A sequence $\mathbf{s} = (s_i)_{i=0}^{\infty}$ is called *periodic* if there is a positive integer p such that $s_{i+p} = s_i$ for all $i \geq 0$. The least positive integer with this property is called the *period* of $\mathbf{s}$.

A *binary feedback shift register* of order n is a mapping $\mathfrak{F}$ from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$ of the form

$$\mathfrak{F} : (x_0, x_1, \ldots, x_{n-1}) \longmapsto (x_1, x_2, \ldots, x_{n-1}, f(x_0, x_1, \ldots, x_{n-1})),$$

where f is a Boolean function of n variables which is called the *feedback function.* The shift register is called a *linear feedback shift register* (*LFSR*) if $\mathfrak{F}$ is a linear transformation from the vector space $\mathbb{F}_2^n$ into itself. Otherwise, the shift register is called a *nonlinear feedback shift register* (*NLFSR*). The shift register is called *nonsingular* if the mapping $\mathfrak{F}$ is a bijection. Further, we will consider only nonsingular and mostly nonlinear feedback shift registers. It can be proved (see e.g. [6]) that the feedback function of a nonsingular feedback shift register has the form

$$f(x_0, x_1, \ldots, x_{n-1}) = x_0 + g(x_1, \ldots, x_{n-1}), \tag{1}$$

where g is a Boolean function of $n-1$ variables.

Consider a binary sequence $\mathbf{s} = (s_i)_{i=0}^{\infty}$ whose first n terms $s_0, s_1, \ldots, s_{n-1}$ are given and whose remaining terms are uniquely determined by the recurrence relation

$$s_{i+n} = f(s_i, s_{i+1}, \ldots, s_{i+n-1}) \quad \text{for all} \quad i \geq 0. \tag{2}$$

We call $\mathbf{s}$ an *output sequence* of the feedback shift register given by (1). The binary n-tuple $(s_0, s_1, \ldots, s_{n-1})$ is called the *initial state vector* of the sequence $\mathbf{s}$ or the *initial state* of the feedback shift register. The recurrence relation (2) can be implemented in hardware as a special electronic switching circuit consisting of n memory cells which is controlled by an external clock to generate the sequence $\mathbf{s}$.

Definition 1. A *de Bruijn sequence* of order n is a sequence of period 2^n of elements of $\mathbb{F}_2$ in which all different binary n-tuples appear in each period exactly once.

It was proved by Flye Sainte-Marie [4] in 1894 and independently by de Bruijn [1] in 1946 that the number of cyclically non-equivalent sequences satisfying Definition 1 is equal to

$$B_n = 2^{2^{n-1}-n}.$$

Definition 2. A *modified de Bruijn sequence* of order n is a sequence of period $2^n - 1$ obtainned from the de Bruijn sequence of order n by removing one zero from all tuples of n consecutive zeros.

In 1990 Mayhew and Golomb [10] investigated sequences satisfying Definition 2 and calculated their linear complexity. These sequences were called by Gammel *et al.* [5] *primitive.* In the case of linear feedback shift registers such sequences are generated by primitive polynomials from the ring $\mathbb{F}_2[x]$ and their theory is quite well understood [6–8]. The task is to find primitive *NLFSRs* with a simple algebraic normal form and this is a time consuming work. An effective method of constructing such primitive *NLFSRs* is not known and one has to search for them. Gammel *et al.* [5] found simple primitive *NLFSRs* up to order 34 used in the design of the stream cipher Achterbahn, but neither the method of searching nor the average time needed to find such good *NLFSRs* have been revealed.

A search for primitive *NLFSRs* with special purpose hardware devices has been undertaken in [12]. The present paper is a continuation of the previous investigations with additional application of software implementations on parallel cores. The search is restricted to looking for nonlinear primitive *NLFSRs* with a simple *ANF.*

3 Quadratic m-sequences

Chen, Games and Rushanan [2] have investigated the case when the feedback function (1) is a quadratic Boolean function of n variables; i.e., it has the following algebraic normal form:

$$f(x_0, x_1, \ldots, x_{n-1}) = \sum_{0 \leq i \leq j \leq n-1} a_{ij} x_i x_j. \tag{3}$$

Let us note that $x_i^2 = x_i$ for all $i \geq 0$, hence the coefficients a_{ii} correspond to the linear terms of the function f. The recurrence (2) corresponding to the quadratic function (3) has the form

$$s_{n+k} = \sum_{0 \leq i \leq j \leq n-1} a_{ij} s_{i+k} s_{j+k} \tag{4}$$

for all $k \geq 0$. The authors of [2] have introduced a notion of quadratic m-sequences by analogy to the linear ones.

Definition 3. A binary sequence $\mathbf{s}$ is called a *quadratic m-sequence of order n* if it satisfies the quadratic recurrence (4) and has period $2^n - 1$.

The authors of [2] have studied algorithmic generation of special form quadratic m-sequences and enumerated them up to order $n = 12$. Namely, they considered quadratic Boolean functions of the form

$$f(x_0, x_1, \ldots, x_{n-1}) = g(x_0, x_1, \ldots, x_{n-1}) + x_i + x_i x_j \tag{5}$$

where $i \neq j,\ 1 \leq i, j \leq n-1$ and

$$g(x_0, x_1, \ldots, x_{n-1}) = x_0 + c_1 x_1 + \cdots + c_{n-1} x_{n-1}$$

is a linear function which generates the m-sequence, i.e., the corresponding polynomial in the ring $\mathbb{F}_2[x]$

$$p(x) = x^n + c_{n-1} x^{n-1} + \cdots + c_1 x + 1 \tag{6}$$

is primitive. Primitive polynomials of degree n have their roots in the finite Galois field $GF(2^n)$; these roots are primitive elements (generators) of the multiplicative group $GF(2^n)^*$. It is known that there is a one-to-one correspondence between linear m-sequences of period $2^n - 1$ and primitive polynomials of degree n. The number of primitive polynomials of degree n is equal to $\varphi(2^n - 1)/n$, where $\varphi(.)$ is the Euler function. The proofs of all these facts can be found in the books [6–8].

The quadratic recurrences corresponding to Boolean functions (5) are modifications of the linear one. The term $x_i + x_i x_j$ introduces a non-linear *perturbation* to the given m-sequence. The states of the *LFSR* for which the function $x_i + x_i x_j$ equals 1 break or join the corresponding cycles of the *LFSR*. The case when after running over all states of the *FSR* we get only one cycle is the sought-for *NLFSR*. In fact, this is a random phenomenon. The relevant discussion has been presented in [2]. In practice, not all primitive polynomials and terms $x_i + x_i x_j$ lead to a suitable coincidence giving a quadratic m-sequence.

4 Software implementation

We have searched for quadratic m-sequences of order n generated by Boolean functions of the form (5) using our software. The procedure is the following:

- one generates all primitive polynomials of a given degree n;
- for a given primitive polynomial all terms $x_i + x_i x_j$ are checked to find a feedback function of *NLFSR* generating a quadratic m-sequence.

The maximum period can only be verified by calculating all states of the *NLFSR* for given i and j. The procedure has been implemented in standard C programming language to speed the search up and for portability. The numbers ($\sharp$) of special form quadratic m-sequences for orders n up to 21 are presented in the Appendix, Tables 1 and 2 (k denotes the number of terms of primitive polynomials of a given degree). For n large checking the period is the most time consuming part of the algorithm. It can be seen from the tables that for the number of terms of the primitive polynomial (6) near $n/2$, the number of quadratic m-sequences found has a maximum. However, the algorithm can be computed in parallel, which speeds up the search. Nowadays multi-core and multi-thread computing environments are widespread even in desktop PCs. The client-server program has been developed to split the searching space into parts. The server program controls the distribution of subtasks and sends client programs jobs to perform using IP packets.

The client is a Linux/Windows program receiving subtasks from the server, doing its jobs and sending back the results. The subtasks are performed as threads on multi-core computers. The server program collects results from all clients over TCP/IP networks. It is worth noting that the subtasks are distributed to platform independent clients (Windows, Linux, Mac OS X). The loss of a client due to e.g. power failure or no network connection is no problem for our server program, because it sends the lost subtasks once again to other clients. The feedback functions of special form quadratic m-sequences up to order $n = 29$ are given in the Appendix.

The above results have been found by using three servers with overall 54 threads connected through an TCP/IP network at Military Communication Institute Interoperability Laboratory [1]. This turned out to be a coarse-grain type work in parallel computing due to low communication between clients and server and using distributed memory. In the future we hope our software can be ported to high-performance computing using Open MPI API to find more quadratic m-sequences for greater orders.

5 Hardware implementation

One way to search for *NLFSRs* in hardware is to duplicate a searching module. For this purpose, the hardware platform containing Field Programmable Gate Array (*FPGA*) of Altera *EP3C80* and the microcontroller have been used. 32 independent programmable searching modules in one *EP3C80* device have been fit. The duplicated modules are managed by the control machine which detects the module readiness and directs the result to the output interface of *FPGA*. The microcontroller transfers the resulting vector to a computer by Ethernet interface. The diagram of a single module is shown in Figure 1.

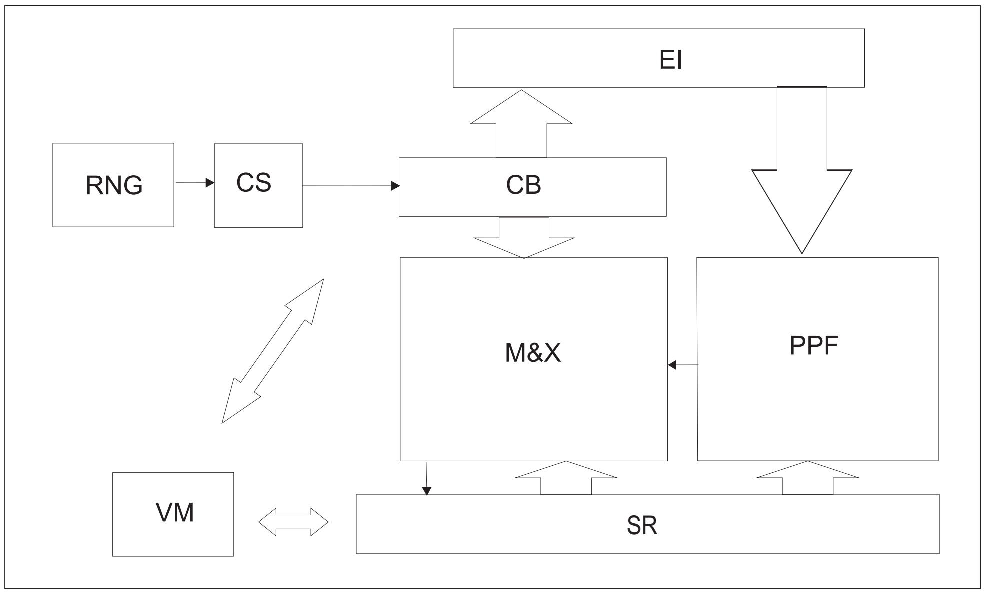

Fig. 1. The diagram of a single searching module

The former version [12] consisted of multiplexers and an *XOR* block called *M&X* which was configured by the coefficients buffer *CB*. The present version adds a primitive polynomial feedback *PPF* which helps to find a *NLFSR* feedback functions faster.

In this search, random numbers for *M&X* are taken from *RNG*. The coefficients are downloaded byte by byte into the coefficient selector *CS*, where their values and repetitions are checked. Then the bytes go to *CB*, whose task is to store combinations of coefficients during the test. The multiplexers partially define the random feedback function of *NLFSR* according to the data buffered in *CB*. Their outputs are connected to the *XOR* gate with a feedback function from *PPF*. The result of *XOR* feeds the shift register *SR*. The *SR* is set with a seed value at the beginning of a searching process by the verification machine *VM* and it starts shifting. After the first repetition of the seed the test is finished. An acceptable result is sent to the Ethernet interface *EI* by *VM*. An unacceptable result starts a new process of random generation and testing. The tests have been carried out on seven such platforms. The results of the search are given in the Appendix.

[1] The equipment has been purchased thanks to a grant of the Polish Ministry of Science and Higher Education decision no 6251/IA/690/2012.

6 Conclusion

In the paper, we have searched for *NLFSRs* generating modified de Bruijn sequences. Methods of parallel computing have been applied both in software and hardware to speed the search up. The software has been used to look for special form quadratic m-sequences. An enumeration of all quadratic m-sequences up to order $n = 21$ generated by feedback functions with the term $x_i + x_i x_j$ and linear terms defined by primitive polynomials has been provided and classified by the weights of these polynomials. The conjecture of Chan, Games and Rushanan [2] has been verified numerically up to order $n = 29$ for *NLFSRs* whose feedback functions have the special algebraic normal form.

Hardware search of *NLFSRs* is a continuation of the previous work [12]. The feedback functions of the registers of orders $n = 29, 30$ and 31 we found are expressed in terms of negations of some arguments; this gives simpler formulae than *ANFs* and it is suitable for implementation.

References

1. N. G. de Bruijn. *A combinatorial problem.* Indag. Math., 8(1946), pp. 461-467.
2. A. H. Chan, R. A. Games, J. J. Rushanan. *On the quadratic m-sequences.* Proceedings of Fast Software Encryption. LNCS vol. 809, pp. 166-173. Springer-Verlag, 1994.
3. E. Dubrova. *A scalable method for constructing Galois NLFSRs with period* $2^n - 1$ *using cross-join pairs.* IEEE Trans. on Inform. Theory, vol. 59(1)(2013), pp. 703-709.
4. C. Flye Sainte-Marie. *Solution to question nr. 48.* L'Intermédiaire des Mathématiciens 1(1894). pp. 107-110.
5. B. M. Gammel, R. Goetffert, O. Kniffler. *Achterbahn 128/80.* The eSTREAM project, www.ecrypt.eu.org/stream/, www.matpack.de/achterbahn
6. S. W. Golomb. *Shift Register Sequences.* San Francisco, Holden-Day, 1967, revised edition, Laguna Hills, CA, Aegean Park Press, 1982.
7. S. W. Golomb, G. Gong. *Signal Design for Good Correlation. For Wireless Communication, Cryptography, and Radar.* Cambridge University Press, 2005.
8. R. Lidl, H. Niederreiter. *Introduction to Finite Fields and their Applications (Revisited Edition).* Cambridge University Press, Cambridge(1994).
9. K. Mandal, G. Gong. *Cryptographically strong de Bruijn sequences with large periods.* Selected Areas in Cryptography. L. R. Knudsen, K. Wu (Eds.). LNCS, vol. 7707, pp. 104-118. Springer-Verlag 2012.
10. G. L. Mayhew, S. W. Golomb. *Linear spans of modified de Bruijn sequences.* IEEE Trans. on Inform. Theory, 36(5)(1990), pp. 1166-1167.
11. J. Mykkeltveit, M-K. Siu, P. Tong. *On the cyclic structure of some nonlinear shift register sequences.* Inform. and Control, vol. 43(1979), pp. 202-215.
12. T. Rachwalik, J. Szmidt, R. Wicik, J. Zabłocki. *Generation of nonlinear feedback shift registers with special purpose hardware.* 2012 Military Communications and Information Systems Conference. MCC 2012, pp. 151-154. IEEE Xplore Digital Library. Cryptology ePrint Archive, 2012/314.
13. R. Wicik, T. Rachwalik. *Modified alternating step generators.* Submitted to MCC2013.

Appendix

The feedback functions of the form (5) generating quadratic m-sequences up to order $n = 29$:

- $n = 22$, $x_0 + x_3 + x_7 + x_9 + x_{13} + x_{14} + x_{18} + x_{19} + x_{20} + x_5 x_{11}$
- $n = 23$, $x_0 + x_6 + x_8 + x_9 + x_{11} + x_{13} + x_{14} + x_{15} + x_{17} + x_{19} + x_{20} + x_{21} + x_{22} + x_2 x_{10}$
- $n = 24$, $x_0 + x_9 + x_{10} + x_{14} + x_{16} + x_{17} + x_{19} + x_{10} x_{15}$
- $n = 25$, $x_0 + x_1 + x_3 + x_5 + x_8 + x_{10} + x_{12} + x_{15} + x_{16} + x_{18} + x_{19} + x_{22} + x_{23} + x_5 x_{12}$
- $n = 26$, $x_0 + x_2 + x_4 + x_5 + x_8 + x_{15} + x_{19} + x_{21} + x_{22} + x_6 x_{16}$
- $n = 27$, $x_0 + x_1 + x_2 + x_4 + x_8 + x_{10} + x_{11} + x_{14} + x_{17} + x_{19} + x_{21} + x_6 x_{10}$
- $n = 28$, $x_0 + x_4 + x_5 + x_6 + x_8 + x_{11} + x_{14} + x_{18} + x_{19} + x_{21} + x_{22} + x_{26} + x_{27} + x_8 x_{27}$
- $n = 29$, $x_0 + x_3 + x_5 + x_6 + x_{11} + x_{12} + x_{16} + x_{19} + x_{22} + x_{23} + x_{27} + x_{20} x_{28}$

The nonlinear feedback functions of shift registers of order n found with the *FPGA* devices:

- $n = 26$, $x_0 + x_{13} + x_{14} + x_{19} + x_{23} + x_{18} x_{22} x_{23} x_{25}$
- $n = 27$, $x_0 + x_3 + x_5 + x_9 + x_{10} + x_{11} + x_{18} + x_{20} + x_{24} + x_9 x_{11} x_{23} x_{26}$
- $n = 28$, $x_0 + x_1 + x_8 + x_{17} + x_{18} + x_{19} + x_{21} + x_{22} + x_{23} + x_4 x_{11} x_{14} x_{18}$

- $n = 29$, $x_0 + x_1 + x_3 + x_4 + x_{13} + x_{16} + x_{18} + x_{19} + x_{20} + x_{22} + x_{23} + x_{27} +$ $x_6\overline{x}_{10}\overline{x}_{19}\overline{x}_{20}x_{21}x_{24}\overline{x}_{26}x_{27} + \overline{x}_6\overline{x}_{10}\overline{x}_{19}x_{20}x_{21}\overline{x}_{24}x_{26}x_{27}$
- $n = 30$, $x_0 + x_4 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{22} + x_{27} + x_{28} +$ $x_1\overline{x}_3x_7\overline{x}_9\overline{x}_{11}x_{13}\overline{x}_{15}\overline{x}_{19}x_{20}\overline{x}_{24}x_{25}x_{27} + \overline{x}_1\overline{x}_3x_7x_9\overline{x}_{11}\overline{x}_{13}\overline{x}_{15}x_{19}\overline{x}_{20}x_{24}\overline{x}_{25}\overline{x}_{27}$
- $n = 31$, $x_0 + x_2 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{14} + x_{15} + x_{16} + x_{20} + x_{21} + x_{26} + x_{29} +$ $x_4x_5x_7x_9x_{12}\overline{x}_{19}\overline{x}_{21}x_{22}x_{24}x_{25}\overline{x}_{26}\overline{x}_{29} + \overline{x}_4x_5\overline{x}_7\overline{x}_9\overline{x}_{12}x_{19}x_{21}\overline{x}_{22}\overline{x}_{24}\overline{x}_{25}x_{26}x_{29}$
  where $\overline{x}_i = x_i + 1$

Table 1. The number of the special form quadratic m-sequences of orders $4 \le n \le 12$

n = 4		n = 5		n = 6		n = 7		n = 8		n = 9		n = 10		n = 11		n = 12	
k	♯	k	♯	k	♯	k	♯	k	♯	k	♯	k	♯	k	♯	k	♯
3	4	3	4	3	6	3	12	3	0	3	2	3	0	3	2	3	0
		5	6	5	6	5	8	5	16	5	12	5	14	5	14	5	4
						7	4	7	4	7	10	7	12	7	26	7	16
										9	0	9	8	9	14	9	6
														11	0	11	2
total	4	total	10	total	12	total	24	total	20	total	24	total	34	total	56	total	28

Table 2. The number of the special form quadratic m-sequences of orders $13 \le n \le 21$

n = 13		n = 14		n = 15		n = 16		n = 17		n = 18		n = 19		n = 20		n = 21	
k	♯	k	♯	k	♯	k	♯	k	♯	k	♯	k	♯	k	♯	k	♯
3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0
5	14	5	6	5	8	5	0	5	0	5	0	5	2	5	0	5	0
7	40	7	20	7	12	7	14	7	18	7	6	7	8	7	4	7	6
9	36	9	36	9	42	9	24	9	50	9	20	9	32	9	8	9	34
11	12	11	22	11	20	11	10	11	46	11	46	11	74	11	28	11	36
13	0	13	0	13	6	13	2	13	10	13	26	13	46	13	18	13	54
				15	0	15	0	15	0	15	6	15	18	15	14	15	32
								17	0	17	0	17	0	17	0	17	10
												19	0	19	0	19	0
total	96	total	84	total	88	total	50	total	124	total	104	total	180	total	72	total	172