Analysis of the Golomb Ruler and the Sidon Set Problems, and Determination of Large, Near-Optimal Golomb Rulers

by

Apostolos Dimitromanolakis

Department of Electronic and Computer Engineering
Technical University of Crete

June 2002

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The discrete mathematics problems of Sidon sets and Golomb rulers have been studied since the 1930's and 1960's respectively by non-overlapping groups of researchers. The main contribution of this thesis will be a study of the relationship between the two problems and the computational verification of a conjecture by Erdős on Sidon sets up to a much larger bound than the one previously known.

Golomb rulers are sets of positive integer numbers having all the differences between any pair of elements of the set to be unique. These numbers can be thought of as ruler marks (at integer locations) as an analogy with common rulers. Golomb rulers have many applications ranging from constructions for error correcting codes, to placement of radio telescopes in linear arrays. They were first studied by Babock in 1953 who was led to their definition to solve a problem in interference between communication channels. Golomb was the first researcher to systematically study the subject in the 1960's and since then his name is associated with these constructions. The function $G(n)$, referred to as the length of an *optimal Golomb ruler*, is defined as the smallest possible length of a ruler with n marks. A review of the work that has been done on Golomb rulers will be presented in chapter 2.

Sidon sets or B_2 sequences is a related problem from combinatorial number theory. These sequences are subsets of $\{1, \ldots, n\}$ having distinct pair wise sums between the elements. Sidon sets are named after Fourier analyst Simon Sidon who defined these sets in order to solve a problem in harmonic analysis. Sidon communicated the problem to Erdős who, together with

Paul Turan, made the first publication on the topic in 1934. It was Erdős who gave the name Sidon sets to these constructions. The function $F_2(n)$ is define as the largest number of element which can be selected from the first n positive integers forming a Sidon set. In chapter 3 we will review the results known for Sidon sets and bounds for the function $F_2(n)$.

Although from the nature of these two problems it is apparent that they are related, they have been studied for the most part independently. Consequently, important theoretical results from Sidon set theory were never applied to Golomb rulers. This is one of the main contributions of the present thesis. In chapter 4, the problems of Sidon sets and Golomb rulers are proven to be equivalent, once an appropriate formalism is used to account for disparate formulations that had been used to date. Once the equivalence is established, a known bound for the function $F_2(n)$ is used to prove an improved bound for the function $G(n)$, more specifically that

$$G(n) > n^2 - 2n\sqrt{n} + \sqrt{n} - 2.$$

When constructing a ruler with a large number of marks, placing the marks so that the maximum mark is in the lowest possible location is a difficult combinatorial optimization problem. There exists no known closed-form formula to generate $G(n)$, and the optimality proofs of such constructions can only be made with exhaustive search methods. The computational complexity of the problem is such, that even with distributed computer search with tens of thousands of computers beyond the year 2000, constructions and optimal Golomb rulers are known only up to 23 marks. The reason is that the time needed for an exhaustive proof of the optimality of a Golomb ruler increases exponentially with the size of the problem.

For values of n larger than 24, one has to resort to near-optimal rulers, which have length close to the optimal one. In chapter 5, a review of all the known constructions from number theory, which lead to near-optimal Golomb rulers, will be presented. All the known constructions apply only for prime or prime power number of marks. We will discuss how one can form a near-optimal Golomb ruler with any number of marks from such constructions.

Moreover the constructions are modular, that is the differences or sums between any two elements must be different modulo some integer z. This modular form allows two similarity transformations to be applied, which can lead to better Golomb rulers. We will describe how one can find the best Golomb ruler that can be formed from a modular construction.

Near-optimal rulers are known for up to 150 marks. For all these near-optimal rulers, the length of the largest element is less than n^2. A yet uproven conjecture, originally stated by Erdős for Sidon sets in 1934 states that Golomb ruler of such length exist for any number of marks. The topic of the last two chapters will be the computational proof of this conjecture for up to 65000 marks.

In chapter 6, we will discuss the various constructions and develop fast algorithms that will allow such a search for near-optimal rulers.

Finally, in chapter 7 we will present the results of this search and announce the proof of the Erdős conjecture for rulers up to 65000 marks that

$$G(n) < n^2 \quad \text{for all } n < 65000$$

or in Sidon set terms

$$F_2(n) < n^{1/2} \quad \text{for all } n < 65000^2.$$

For the purposes of the search a distributed computer network with 10 nodes was utilized and about 21 CPU days were used for the computations.

Chapter 2

Golomb Rulers

2.1 Golomb Rulers

Common rulers have their marks equally spaced in some unit of measure (for example 1 cm), so someone can measure any distance between 1 and the length of the ruler by placing an object between any two marks with the desired distance. For example to measure a distance of 5 cm, it is possible to place an object between marks of 0 and 5 cm or 1 and 6 cm etc.

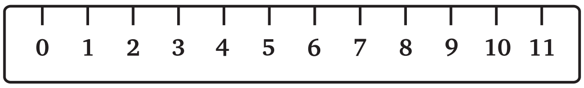


Figure 2.1: A common ruler

Figure 2.2: A Golomb ruler

Golomb rulers can be thought as a special kind of rulers. For a ruler to be Golomb, you must have only one choice if you want to measure a specific distance. More specifically every distance between two numbers (or marks) must be different from all the others. If this holds then a given ruler is Golomb.

For example if there is a mark at position 2 and another one at position 5, then no other pair of marks must be seperated by a distance of 3. From this definition it is obvious that a common ruler with more than 2 marks is not Golomb.

Using the Golomb ruler in figure 2.2 one can measure the distances $\{1, 2, 3, 4, 5, 7, 8, 9, 10, 11\}$ by a suitable choice of two marks but no other distances can be measured. Moreover for each of these distances, only one pair of marks can be used to make such a measurement, therefore the Golomb property is satisfied.

2.1.1 Uses of Golomb rulers

Golomb rulers are named after Solomon W. Golomb, professor of Engineering and Mathematics of the University of Southern California.

Babock[6] was the first to use Golomb rulers, under a different name, to solve a problem in inteference between radio communication channels. If the frequencies of the channels are assigned in proportion to the marks of a Golomb ruler, then Babock has found that third-order interference between the channels is eliminated.

Although Babock was the first to study Golomb rulers, they were named by Golomb who was the first to do a systematic treatment of the subject. Similar constructions has been studied by other authors [47] [5] under different names like time-hopping patterns and DDS (distinct difference sets).

Professor's Golomb name is more commonly associated with such constructions and we will this name for our purposes.

Since then, Golomb rulers have been applied to number of applications, ranging from coding theory to radio astronomy.

Particularly, in radio astronomy [9] [10] astronomers often use an array of telescopes in a single line to measure different measurements of the light or electromagnetic radiation of a distant star. By a process called interferometry, which works by finding the difference of the measurements between two telescopes taken precisely at the same time, more information can be extracted than by simple examining individual observations of the telescopes.

A measurement is different from another if the distance between the two telescopes used for the first measurement is different from the distance used for the second. If the telescopes are placed in positions dictated by the marks of a Golomb ruler the number of different pairwise distances will be maximized as we will shortly see.

Other applications of Golomb rulers are in the construction of radio system withour third order intermodulation by Babock [6] and the construction of convolutional self-orthogonal codes (CSOC) by Robinson and Bernstein [47]. For a more thorough investigation of Golomb ruler uses in various field of science see [46].

2.1.2 Formal definition

A Golomb ruler consists of a set of integer numbers. These integer number are called **marks** as in the case of common wooden rulers.

We now proceed to formally define the notion of Golomb rulers.

Definition (Golomb ruler). A set of integers

$$A = \{a_1, a_2, \ldots, a_n\} \quad a_1 < a_2 < \ldots < a_n$$

is called a Golomb ruler if for each integer $x \neq 0$ there is at most one solution to the equation

$$x = a_j - a_i \quad a_j, a_i \in A$$

Notice that a Golomb ruler does not necessarily start at position 0, it can begin at some positive or even negative point. However, usually our constructions will begin at position 0 and we will define later a canonical form of Golomb rulers that always has it's first mark at position 0.

The difference between the largest and the smallest element of the set $a_n - a_1$ is called the **length** of the ruler. Examining a Golomb ruler, one can see than it is difficult to pack a large number of marks inside a ruler with small length. The problem of finding the smallest ruler length that can hold a given number of marks is difficult. It has been studied extensively but up to now no exact solution exists.

We are interested in rulers which have the smallest possible length for a given number of marks. These rulers are called **optimal Golomb rulers**. If a ruler is not optimal but its length is close to optimal it will be called a **near-optimal** Golomb ruler.

From now on, let $G(n)$ be the minimum length of a ruler with n marks.

Definition (G(n)). For every integer $n > 0$, $G(n) = d$ if for ruler having n marks d is its smallest possible length.

We will be interested in providing exact values or bounds for the function $G(n)$. The first lower bound for function $G(n)$ can be easily be found using a double counting argument.

2.1.3 Elementary properties of G

As there are $\frac{1}{2}n(n-1)$ positive differences between any two marks and all of them must be distinct, a Golomb ruler measure exactly that many differenct distances. All these distances must belong to the set $\mathbf{N}^{+} = 1, 2, \ldots$ and the largest of them is the length of the ruler. From this obervation it follows that the length of the ruler is at least $\frac{1}{2}n(n-1)$, that is

$$G(n) \geqslant \frac{1}{2}n(n-1) \tag{2.1}$$

This is very close to the best lower bound known for $G(n)$.

Another interesting property of $G(n)$ is that it is a strictly monotonically increasing function. To prove this consider the optimal Golomb ruler with n marks with length $G(n) = d$. If we remove the largest element from the ruler, then we have a ruler with $n-1$ marks and distance less than d. It follows that $G(n-1)$ must be less than or equal to d, so generally it holds that:

$$G(n) > G(n-1) \tag{2.2}$$

Sometimes, we will also be interested in counting how many distinct Golomb rulers exist for a given length d with n marks. We define the function $C(n, d)$ to denote this number.

Definition ($C(n, d)$). For every n, d define

$$C(n, d) = |A = \{a_1, \ldots, a_n\} \ : \ a_n - a_1 = d \text{ and } A \text{ is a Golomb ruler}|$$

the number of Golomb rulers with n marks and length exactly d.

With this definition $G(n) = d$, if d is the smallest possible integer satisfying $C(n, d) > 0$.

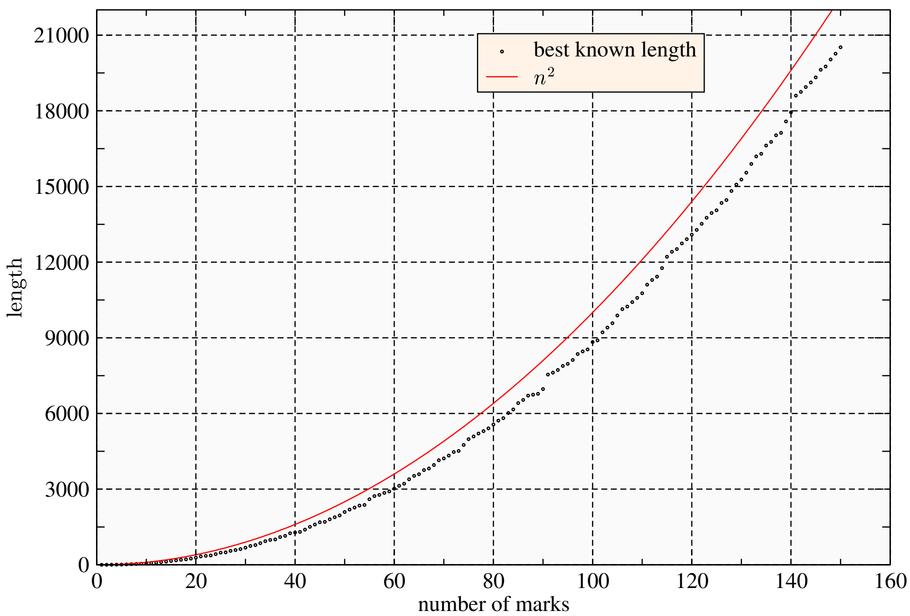


Figure 2.3: Smallest known values for $G(n)$

2.2 The search for optimal Golomb rulers

Fining an optimal Golomb ruler is a difficult computational problem. Although it has not been proved to be $NP - hard$, that is requiring exponential time by today's best algorithms, it is believed that no polynomial time algorithm exists for this problem.

The problem of finding an exact value for $G(n)$, consists of two parts. First one must exhibit the ruler to be shown optimal which should verified to be Golomb and then prove that this ruler has the shortest possible length.

Currently the only way known to prove a statement like this is to exhaustively search all the possible rulers with shorter lengths and n marks, and prove that none has the Golomb property. This might seem as a dumb brute force attack to the problem but currently no other algorithm is known to be better.

The best algorithms known[46, 19], employ heuristics and a clever way of skipping the examinination of rulers that are surely not Golomb. Still the time required to find a value of $G(n)$ increases by a factor of more than 10 when n is increased by 1.

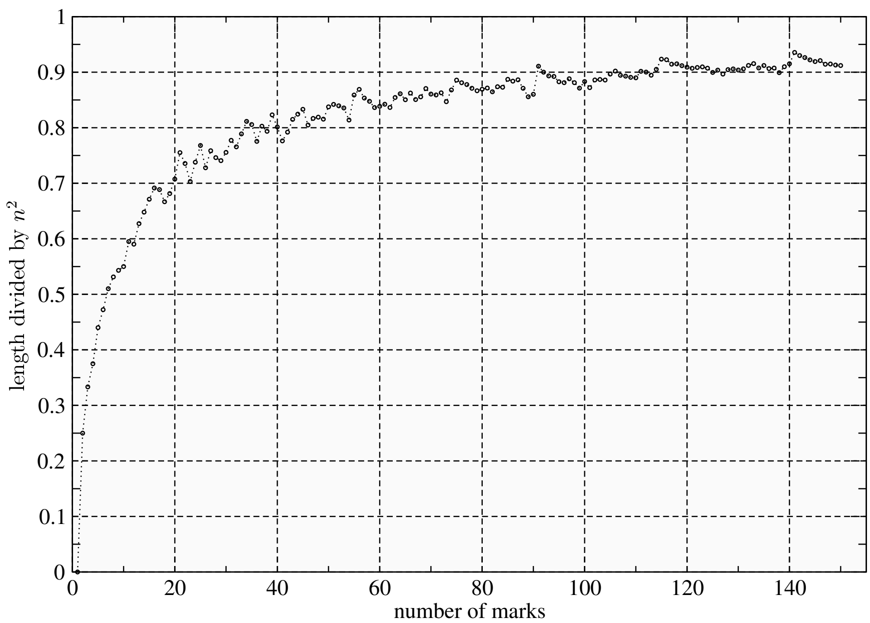


Figure 2.4: Smallest known values for $G(n)$ divided by n^2

However, it is possible that better algorithms than brute force search exist. As an analogy, consider the integer factoring problem: given an integer find it's prime factors. One can of course think that the only way to do this is to start dividing the number with all the prime numbers until a factor is found. Indeed, it was the case that this simple algorithm was the best known for many years.

Yet, as in the past few years the factoring problem became suddenly more attractive, being used for almost all the encryption and decryption of data taking place in the internet and all digital banking transactions, faster algorithms were developed. These algorithms do not have anything to do with trial division of the number, in fact some of them don't even do divisions at all.

Still today, only the brute force attack on the Golomb ruler problem is known.

A personal computer (with CPU clock about 1GHz) can find the values of $G(n)$ for n up to 18 in some hours. However the search for $n = 24$, required computations by a distributed computer network of thousands of computers for almost a year. This network was coordinated by distributed.net[1] and they have found optimal Golomb rulers with sizes 20,21,22 and 23.

As of the writing of this paper, the verification of $G(24)$ is still in progress by distributed.net and the computation of $G(25)$ has began.

Table 2.1: Known values of $G(n)$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$G(n)$	0	1	3	6	11	17	25	34	44	55	72	85	106	127

n	15	16	17	18	19	20	21	22	23	24
$G(n)$	151	177	199	216	246	283	333	356	372	425

The known values of $G(n)$ which have been proved to be optimal are shown in table 2.1. In figure 2.3 the smallest known values for $G(n)$ are compared against n^2.

Values for n more than 24 are not proven to be optimal and are the shortest rulers found to date using different methods we will describe later.

Looking to the graph, there seem to always exist rulers with length less than n^2 that is $G(n) < n^2$ for all n. This is an unproven conjecture and it is one of the most challenging open problems concerning Golomb rulers. Figure 2.4 gives the ratio $G(n)/n^2$. It appears that $G(n)/n^2$ is asymptotic to 1 as n approaches infinity which also supports the conjecture.

2.3 Similarity transformations

Golomb rulers adhere to two simple similarity transformations that produce new rulers which also have the Golomb property: the translation property and the multiplication property.

Much later we will use this two properties in another, stronger, form to oftern improve the length of a Golomb ruler.

Property 1 (Translation). *If the set $A = \{a_1, a_2, \dots, a_n\}$ is a Golomb ruler then so is the set*

$$A' = \{x + a_1, x + a_2, \dots, x + a_n\}$$

[1] For more information see `http://distributed.net/ogr`.

for every integer x.

Proof. If A' is not a Golomb ruler then there must exist i, j, k, l such that $(x+a_i)-(x+a_j) = (x+a_k)-(x+a_l)$ but this would imply that $a_i - a_j = a_k - a_l$ a contradiction since A is a Golomb ruler. □

Property 2 (Multiplication). *If the set $A = \{a_1, a_2, \ldots, a_n\}$ is a Golomb ruler then so is the set*

$$A' = \{za_1, za_2, \ldots, za_n\}$$

for every non-zero integer z.

Proof. As in the previous case, if A' is not a Golomb ruler then there must exist i, j, k, l such that $za_i - za_j = za_k - za_l$ but this would imply that $a_i - a_j = a_k - a_l$ a contradiction. □

We will use the notation $t + A$ to refer to the translation of ruler A by t and zA to refer to the multiplication of Golomb ruler A by z.

Using the translation property every ruler can be translated so that $a_1 = 0$ and from now on when we refer to Golomb rulers we will assume that the ruler begins at 0. We call it the **canonical form** of a Golomb ruler.

Now, suppose you put a mirror on the left end of the ruler. Then the marks will be projected to some other points in the other half plane, but their mutual distances will remain the same. Therefore, this forms a new Golomb ruler. We can form this ruler by using both properties as

$$A' = a_n - 1A = \{a_n - a_1, a_n - a_2, \ldots, a_n - a_n\}$$

This ruler is called the **mirror image** of the original. It begins in 0 and has the same length as the original, which is equal to the position of the last mark a_n. The marks of the mirror image can be thought to be produced by mirroring the marks of the original ruler in front of a mirror.

Every Golomb ruler has a mirror image. Except for the case of $\{0, 1\}$ all the other mirror images are provably different from the original ruler. This is the case because if the two images were the same, the ruler would measure the same distances in the left half and in the right half. So for any given length d and $n > 2$ there are an even number of Golomb rulers, that is $C(n, d)$ is even for $n > 2$.

For the purposes of our discussion, we regard both the ruler and it's mirror image as equivalent, however we will count both of them when referring to the function C.

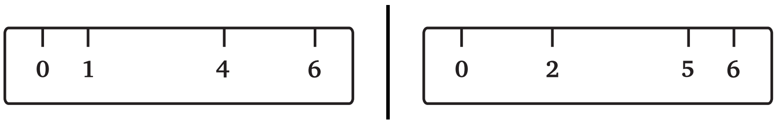


Figure 2.5: A ruler and it's mirror image

2.4 Perfect Golomb rulers

A Golomb ruler with n marks measures exactly $\frac{1}{2}n(n-1)$ distances. When these distances are exactly the first $\frac{1}{2}n(n-1)$ positive integers, we have a perfect Golomb ruler. For example, the ruler $\{0, 1, 4, 6\}$ measures the distances $\{1, 2, 3, 4, 5, 6\}$ and is perfect. The following rulers (and their mirror images) are perfect:

$$\begin{array}{lllll}
(n=1) & 0 & & & \\
(n=2) & 0 & 1 & & \\
(n=3) & 0 & 1 & 3 & \\
(n=4) & 0 & 1 & 4 & 6
\end{array}$$

Actually, these four are the only perfect Golomb rulers. For $n > 4$ no perfect Golomb rulers exist and we can present a simple proof of this fact.

Theorem 2.1. (Golomb) *A Golomb ruler with more that 4 marks cannot be perfect, that is for* $n > 4$

$$G(n) > \frac{1}{2}n(n-1). \tag{2.3}$$

Proof. Suppose we have a perfect ruler with $n > 4$ and $G(n) = d$ where $d = \frac{1}{2}n(n-1)$. Trying to place the marks of the ruler will lead us to a contradiction.

The ruler must measure distance $d - 1$ and this distance must be measured starting from one of the two edges of the ruler. This means that the distance 1 is measured starting from the other edge of the ruler. By the mirroring property, without loss of generality we can assume that 1 is measure starting from the left edge that is there marks at positions 0 and 1.

Now consider distance $d-2$. It must be measured either between $(0, d-2)$ or $(1, d-1)$ or $(2, d)$. The first possibility generates two times the distance $d-1$ and is invalid. Also, the second possibility generates two times the distance 1. Thus only the third possibility is valid, this means there are marks at positions $d-2$ and d.

Until now, we proved that there must be marks at positions: $\{0, 1, d-2, d\}$. Distance $d-3$ is also measured between 1 and $d-2$.

We also have to measure distance $d-4$. The possible positions of two marks to measure this distance are:

$$(0, d-4) \quad (1, d-3) \quad (2, d-2) \quad (3, d-1) \quad (4, d)$$

Any of these posibilities would violate the Golomb property. For example placing a mark at $d-4$ would generate twice the distance 2.

The proof is valid only if $d-4$ is greater than 2, or else if $d-4 \leqslant 2$ we would have already placed distance $d-4$ at the final step. That means

$$d-4>2 \;\Rightarrow\; d>6 \;\Rightarrow\; \frac{1}{2}n(n-1)>6 \;\Rightarrow\; n(n-1)>12$$

This holds for $n>4$. $\square$

2.5 Optimal Golomb rulers

When n exceeds 4 then no perfect Golomb rulers exist. In this case the best one can do is find the shortest possible Golomb ruler with n marks, an optimal Golomb ruler.

Optimal Golomb rulers have been found and proved to be optimal for up to 23 marks. Rulers with 5 to 7 marks have been proved by Robinson and Bernstein [47] and for 8 to 11 marks by William Mixon.

The search was continued by Robinson[47] who found the optimal rulers for 12 and 13 marks in 1979 and Shearer [51] who gave rulers with 14 to 17 marks in 1990.

The next two rulers 17 and 18 were proved by Olin Silbert (unpublished, in 1993) and the 19 mark ruler was found by Rankin[46] also in 1993 by computer search using about 36,200 CPU hours.

From this point, the search was continued as a web project by Mark Garry, David Vanderschel and others and later moved to distributed.net[2]. They finished the search for the 20,21,22 and 23 mark rulers and currently the 24 and 25 mark ruler search is continued.

All known optimal Golomb rulers, not including their mirror images, are shown in table 2.2.

In view of these results and the exponential increase in computational time for proving optimality, it is unlikely that optimal Golomb rulers with many more marks will be proved in the following years.

2.6 Near optimal Golomb rulers

When n exceeds 23, no optimal Golomb rulers are known. In this case the best one can do is find a near-optimal ruler, that is one with length near the optimal one.

Near optimal rulers for large number of marks cannot be found by exhaustive computer search. In this case one has to resort to algebraic constructions that give near-optimal Golomb rulers. Most of these rulers proved to be optimal have been found in this way and so one is justified to call rulers generated by such constructions near-optimal.

Currently, the computation of near optimal Golomb rulers has been done for up to 100 marks by Atkinson, Santoro and Urrutia[5] and up to 150 marks by Lam and Sarwate[40] in 1988. From this point, no other near-optimal rulers have been found. The lengths of the best near optimal rulers are plotted in figure 2.3.

2.7 Summary

This chapter consisted of an introduction to Golomb rulers. Apart from the definition of the problem, the notion of optimal and near-optimal Golomb rulers was defined. Optimal Golomb rulers are known only for up to 23 marks and the computational power needed to verify a Golomb ruler

[2] `distributed.net/ogr`

Table 2.2: Known optimal rulers (not including mirror images)

n	length	position of marks
1	0	0
2	1	0 1
3	3	0 1 3
4	6	0 1 4 6
5	11	0 1 4 9 11 0 3 4 9 11
6	17	0 1 4 10 12 17 0 1 4 10 15 17 0 3 5 9 16 17 0 4 6 9 16 17
7	25	0 1 4 10 18 23 25 0 2 3 10 16 21 25 0 2 6 9 14 24 25 0 1 7 11 20 23 25 0 3 4 12 18 23 25
8	34	0 1 4 9 15 22 32 34
9	44	0 3 9 17 19 32 39 43 44
10	55	0 1 6 10 23 26 34 41 53 55
11	72	0 1 4 13 28 33 47 54 64 70 72 0 1 9 19 24 31 52 56 58 69 72
12	85	0 2 6 24 29 40 43 55 68 75 76 85
13	106	0 7 8 17 21 36 47 63 69 81 101 104 106
14	127	0 5 28 38 41 49 50 68 75 92 107 121 123 127
15	151	0 6 7 15 28 40 51 75 89 92 94 121 131 147 151
16	177	0 1 4 11 26 32 56 68 76 115 117 134 150 163 168 177
17	199	0 5 7 17 52 56 67 80 81 100 122 138 159 165 ⋯ ⋯ 168 191 199
18	216	0 2 10 22 53 56 82 83 89 98 130 148 153 167 ⋯ ⋯ 188 192 205 216
19	246	0 4 13 15 42 56 59 77 93 116 126 138 146 174 ⋯ ⋯ 214 221 240 245 246
20	283	0 24 30 43 55 71 75 89 104 125 127 162 167 189 ⋯ ⋯ 206 215 272 275 282 283
21	333	0 4 23 37 40 48 68 78 138 147 154 189 204 238 ⋯ ⋯ 250 251 256 277 309 331 333
22	356	0 1 9 14 43 70 106 122 124 128 159 179 204 223 ⋯ ⋯ 253 263 270 291 330 341 353 356
23	372	0 6 22 24 43 56 95 126 137 146 172 173 201 213 ⋯ ⋯ 258 273 281 306 311 355 365 369 372

increases exponentially. This means that unless new algorithms are found, optimal Golomb rulers with 26 or more marks are not likely to be known in the following years.

When the number of marks exceeds this point, the best one can do is find a near-optimal ruler, which has length close to the optimal one. The search for near-optimal rulers has been done for up to 150 marks. Later, in chapter 6 and 7, we will extend this search to a much larger bound. In the next chapter, temporarily we will forget what we know about Golomb rulers and describe a related problem from number theory.

Chapter 3

Sidon Sets

After the discussion of Golomb rulers, we will focus on a problem from additive number theory closely related to Golomb rulers: Sidon sequences, also called $\mathcal{B}_2$ sequences by some authors. In this chapter, we will present a discussion of Sidon sets and the known results regarding upper and lower limit and

3.1 History of the problem

Sidon sets are named after Simon Sidon, a Fourier analyst which was the first to pose this problem in 1932 [52]. Sidon considered the problem when he investigated problems related to Fourier series.

Paul Erdös, probably the most famous mathematician of the 20^{th} century, met Sidon, who described him the problem. Erdös was fascinated as it involved both combinatorial and number theory, the two fields in which he worked most of his time. He named the problem Sidon sequences and together with his friend Paul Turan, published the classical paper of 1934 *On a problem of Sidon in additive number theory* [25]. This paper was the first systematic treatment of the problem.

Since then, a number of authors have improved the results of Erdös and Turan. Nevertheless, the best efforts have resulted only in bounding the possible solutions of the problem and not providing a general solution.

3.2 Definition of a Sidon sequence

A Sidon set is a subset of the set $A = 1, \ldots, N$ of positive integers which have the property that for each two elements a_i, a_j of the set, their sum $a_i + a_j$ is different from all other sums.

A more formal definition will allow us to generalize the problem later. Define the representation function for every integer x as the number of ways that this integer can be represented as a sum of two elements of the set.

Definition. For every $x \in \mathbf{N}$ denote the representation function $r_A(x)$ of x in any set A as

$$r_A(x) = |\ \{(a,b) : a,b \in A,\ a \leqslant b,\ x = a + b\}\ |$$

We can now define a Sidon sequence more formally.

Definition (Sidon set). A Sidon set or $\mathcal{B}_2$ sequence $A = \{a_1 < a_2 < \ldots < a_k\}$ is a subset of $\{1, 2, \ldots, n\}$ such that

$$r_A(x) \leqslant 1 \qquad \forall x \in \mathbb{N}^+$$

Sidon set are also called $\mathcal{B}_2$ sequences by some authors as the name Sidon has a very different meaning in harmonic analysis. From now on, we will use the name $\mathcal{B}_2$ sequence to describe such sequences.

3.2.1 Generalizations of $\mathcal{B}_2$ sequences

Sidon sets are called $\mathcal{B}_2$ sequences every sum of 2 elements of the set is different from all others. Generalizing, we can define $\mathcal{B}_n$ sequences for every $n > 0$. A $\mathcal{B}_n$ sequence has the property that every sum

$$a_1 + a_2 + \ldots + a_n$$

of n elements of the sequence is different from all others.

Also, by using a positive integer constant a and let the representation function be less than or equal to a that is $r_A(x) \leqslant a$ we can define $\mathcal{B}_2[a]$ sequences for every $a \geqslant 1$. This sequences has the property that every integer can be represented at most a time as sums of two elements of the set.

From now on, we shall only be concerned with $\mathcal{B}_2[1]$ sequences or Sidon sets.

3.2.2 Dense $\mathcal{B}_2$ sequences

Much like the treatment of Golomb rulers, we will be mostly interested in dense Sidon sets.

Since not all elements of $\{1, 2, \ldots, n\}$ can be selected for a $\mathcal{B}_2$ sequence there is a maximum number of elements that can be selected. The problem of finding that maximum number is hard and no closed-form solution exists, as the case is with Golomb rulers. Define this number for any n as $F_2(n)$.

Definition (F_2). Let $F_2(d)$ be the maximum size of a $\mathcal{B}_2$ sequence contained in $\{1, \ldots, d\}$ that is

$$F_2(d) = k$$

if k is the maximum cardinality of a Sidon set $\mathcal{B}_2$ contained in the first d positive integers.

F_2 obviously is a non-decreasing function as it not possible to select more integers that form a $\mathcal{B}_2$ sequence in a smaller interval.

3.3 A survey of results in Sidon sets

3.3.1 Upper Bounds of F_2

We can find a trivial upper bound for $F_2(d)$ by counting the differences it a Sidon set measures. Notice a Sidon set with size $F_2(d)$ measures

$$\binom{F_2(d)}{2}$$

distinct positive differences. Since there are only d possible positive integers in $\{1, \ldots, d\}$ we must have that

$$\binom{F_2(d)}{2} \leqslant d \quad \Rightarrow \quad F_2(d) \cdot (F_2(d) - 1) \leqslant 2d$$

so

$$F_2(d) \leqslant \sqrt{2}\, d^{1/2}$$

Erdös and Turan were the first to improve this bound in 1941. They proved in [25] that

$$F_2(d) \leqslant d^{1/2} + O(d^{1/4})$$

This lower bound was further improved by Lindström[42] and independently by Klazar[33] and the following is, as of today, the tightest upper bound known. We will provide a simple combinatorial proof similar to the one of Lindström.

Theorem 3.1 (Lindström).

$$F_2(d) < d^{1/2} + d^{1/4} + 1$$

Proof. Let $A = a_1 < a_2 < \ldots < a_r$ be a $\mathcal{B}_2$ sequence from the set $\{1, 2, \ldots, d\}$. The differences $a_j - a_i$, $1 \leqslant i < j \leqslant r$ must be all different. We call the positive number $j - i$ the *order* of the difference $a_j - a_i$.

For a given order ν consider the sum of all differences of order ν

$$\Sigma_\nu = \sum_{i=1}^{r-\nu} (a_{i+\nu} - a_i)$$

The sum can be split in ν sequences of the form

$$\begin{gathered}(a_{\nu+1} - a_1) + (a_{2\nu+1} - a_{\nu+1}) + (a_{3\nu+1} - a_{2\nu+1}) + \ldots \\ (a_{\nu+2} - a_2) + (a_{2\nu+2} - a_{\nu+2}) + (a_{3\nu+2} - a_{2\nu+2}) + \ldots \\ \vdots\end{gathered}$$

As a result of cancellations, each of the ν sequences has sum at most d and the total sum of all differences of order ν is at most νd.

Consequently, the sum of all differences of order at most m is at most

$$\Sigma_1 + \Sigma_2 + \ldots + \Sigma_m < (1 + 2 + \ldots + m)d = \frac{1}{2}m(m+1)n. \tag{3.1}$$

There are $r - \nu$ differences of order ν. The number of differences of order at most m is

$$(r-1)+(r-2)+\ldots+(r-m)=mr-\frac{1}{2}m(m+1)=ms$$

where $s=r-\frac{1}{2}(m+1)$. Since all these differences must be different, we find that

$$\Sigma_1+\Sigma_2+\ldots+\Sigma_m \geqslant 1+2+\ldots+ms=\frac{1}{2}ms(ms+1) \tag{3.2}$$

Using equations 3.1 and 3.2 and the inequality $ms(ms+1)>m^2s^2$, we bound s from above

$$\frac{1}{2}m^2s^2<\frac{1}{2}m(m+1)d \quad \Longrightarrow \quad s<n^{1/2}\sqrt{1+m^{-1}}$$

To simplify the expression, notice that for all x, $\sqrt{1+x}<1+\frac{1}{2}x$ and let $x=m^{-1}$. Moreover, since m^{-1} is less than 1, we have a good approximation $\sqrt{1+x}\approx 1+\frac{1}{2}x$ of the radical. By substituting s we get

$$r<\frac{1}{2}(m+1)+d^{1/2}(1+\frac{1}{2}m^{-1})$$

The optimal choice of m, since it must be an integer is $m=\lfloor d^{1/4}\rfloor \leqslant d^{1/4}$. Substituting m, we conclude that

$$r<d^{1/2}+d^{1/4}+1$$

□

3.3.2 Lower Bounds of F_2

In theorem 3.1 a strict upper bound on function F_2 is proved. On the other hand, strict lower bounds for F_2 are quite harder to prove. The reason is that to assert $F_2(n)\geqslant d$, the only method is to actually construct a Sidon set that exhibits this bound.

In chapter 5 we will see that for special cases of the number of marks there exist constructions that give strict lower bounds for $F_2(n)$. However, strict lower bounds have not been found for any possible choice of n.

The best lower bound known which is asymptotic to n^2 is

$$F_2(n) > n^{1/2} - O(n^{5/16})$$

which has been proved by Erdös and Turan [25].

3.3.3 Well Distribution in residue classes

Lindström [44] has proved that the numbers that form a Sidon sequence of size more than $n^{1/2}$ are well distributed in residue classes modulo m. That is about $1/m$ of all the elements fall in each of the m residue classes.

Theorem 3.2 (Lindström). *Let $A \subseteq [1, n]$ be a Sidon set with $r = |A| = (1 + o(1))n^{1/2}$. For a fixed integer $m \geqslant 2$ let $A_i = \{a \in A : a \equiv i \pmod{m}\}$ and $r_i = |A_i|$, $0 \leqslant i < m$. Then $r_i/\sqrt{n} \to 1/m$ when $n \to \infty$.*

He also proved than in the special case when $m = 2$ the number of even and odd elements is almost equal.

Theorem 3.3 (Lindström). *Let $A \subseteq [1, n]$ be a Sidon set of size $r \geqslant n^{1/2}$. Then for the number r_0 of even elements and the number r_1 of odd elements*

$$|r_0 - r_1| < 4\sqrt{r_0^{3/2} + r_1^{3/2}} = O(n^{3/8}).$$

Kolountzakis [39] has proved a more general result that does not impose the restriction of Lindtröm's theorem on the number of elements of the Sidon set. Let

Theorem 3.4 (Kolountzakis). *Let $A \subseteq [1, n]$ be a Sidon set with size*

$$r = |A| \geqslant n^{1/2} + l(n)$$

where $l(n) = o(n^{1/2})$.

For each modulus m define

$$a(x) = |\{a \in A : a \equiv x \pmod{m}\}|$$

be the number of elements that fall in each residue class. Then, if $m = o(n^{1/2})$

$$||a(x) - \frac{r}{m}||_2 \leqslant C \begin{cases} \dfrac{n^{3/8}}{m^{1/4}} & \textit{if } l \leqslant n^{1/4}m^{1/2} \\ \\ \dfrac{n^{1/4}l^{1/2}}{m^{1/2}} & \textit{otherwise.} \end{cases}$$

where $||f(x)||_2 = \sqrt{\sum_{x \in \mathbb{Z}_m} |f(x)|^2}$.

The theorem of Lindsröm follows from this result.

3.3.4 Linear distribution of elements

Another interesting property of Sidon sets is that the elements of a large Sidon set are well distributed in the interval $[1, n]$. If a_i, $1 \leqslant i \leqslant k$ is a Sidon set in $[1, n]$ then

$$a_i \approx \frac{i}{n}.$$

This has been proved by Erdős and Freud in [23]. Graham [27] has proved a more precise result.

Theorem 3.5 (Graham). *Let* $A \subseteq [1, n]$ *be a Sidon set with* $n^{1/2} + O(n^{1/4})$ *elements. Then any interval of length* cn *contains* $cn^{1/2} + O(n^{3/8})$ *elements.*

It follows from this theorem that the maximum gap between any two consecutive elements of the set is

$$\max\{a_{i+1} - a_i\} = O(n^{3/8}).$$

Cilleruelo has further improved this result in [15].

Theorem 3.6 (Cilleruelo). *Let* $A \subseteq [1, n]$ *be a Sidon set with* $n^{1/2} - L$ *elements. Then any interval of length* cn *contains* $c|A| + E_I$ *elements where*

$$|E_I| \leqslant 54n^{1/4}(1 + c^{1/2}n^{1/8})(1 + L_+^{1/2}N^{-1/8}), \quad L_+ = \max\{0, L\}$$

In particular one can deduce from this theorem that the maximum gap that occurs in a sequence with $n^{1/2} + O(n^{1/4})$ elements is

$$\max\{a_{i+1} - a_i\} = O(n^{3/4}).$$

3.4 Summary

In this chapter, Sidon sets were introduced and the known properties of the elements of a Sidon set were reviewed.

The elements of Sidon sets have all the good properties one might hope: they are well distributed linearly inside the selected interval, they are well distributed in residue classes and their size is about $n^{1/2}$. However all the theorems that have been proved only provide asymptotic results.

When one looks into a Sidon set more closely the (small asymptotically) variation between exact linear distribution and the actual distribution of the elements is significant. Moreover, the difference between $n^{1/2}$ and the size of a Sidon set can approach to 0 as a limit but this difference is enough to make the problem of determining the exact maximum size of a Sidon set still an open problem.

Chapter 4

Equivalence of the two problems

Sidon sequences and Golomb rulers have equivalent definitions, as we will shortly prove. However, no systematic treatment of the relation between the two problems has been done up to now. In fact, some authors seem to ignore the relation between the two problems and reprove old results using their own methods. Distinct notation is what has prevented bounds between the two equivalent problems to be united.

We will study the equivalence between the two problems and unite in some sense the results on bounds of the sizes of Sidon sets and Golomb rulers. Theorems that will allow future results on one problem to be easily restated to the other will be presented.

Using the main result of this chapter, theorem 4.5, we will prove an improved lower bound in theorem 4.9 for the length of optimal Golomb rulers.

4.1 Equivalence of Sidon sets and Golomb rulers

From a quick look at the definition of Sidon sets it is not hard to notice the relation to the Golomb ruler problem: a set having distinct differences between any two elements will also have distinct sums and vice versa. To illustrate this, consider that for any four elements a_i, a_j, a_k, a_l we have that

$$a_i + a_j = a_k + a_l \iff a_i - a_k = a_l - a_j. \tag{4.1}$$

Now we can prove that the two definitions are equivalent.

Proposition 4.1. If a set A is a $\mathcal{B}_2$ sequence then it is a Golomb ruler and vice versa.

Proof. Suppose that A is $\mathcal{B}_2$ but not a Golomb ruler. Since it not a Golomb ruler, there must exist elements a_i, a_j, a_k, a_l with $a_i - a_j = a_k - a_l$. This would imply, by equation 4.1, that $a_i + a_l = a_k + a_j$, a contradiction since it is a Sidon set.

Now suppose A is a Golomb ruler but not a Sidon set. Since it is not a $\mathcal{B}_2$ sequence, there must exist elements a_i, a_j, a_k, a_l having $a_i + a_j = a_k + a_l$ with $\{i, j\} \neq \{k, l\}$.

At most 2 of these elements can be identical and suppose $i \neq l$. Then we can arrange them in differences $a_i - a_l = a_k - a_j$ with $\{i, l\} \neq \{k, j\}$.

This is a contradiction since the original set is a Golomb ruler. □

If we view $\mathcal{B}_2$ sequences in Golomb ruler terms, $F_2(d)$ is the maximum number of marks that can be placed in the interval $\{1, \ldots, d\}$.

4.2 Relations between $G(n)$ and $F_2(n)$

In the past many authors have worked independently between the two problems. Sometimes essentially the same results have been reproved and published in different journals, like the fact that $G(n)$ asymptotically approaches n^2 as $n \to \infty$. It has been proved by Singer and Erdös but has been republished in [5] in 1986.

To enable us to restate results between the two problems, an investigation of the relation between $G(n)$ and $F_2(n)$ will be presented. Remember that, by definition 2.1.2 (page 8), $G(n)$ is the minimum length of a Golomb ruler with n marks.

It is should be clear by now that between these two function there is some sense of inverse relation: $G(n)$ refers to the minimum size of a Golomb ruler given the number of marks and $F_2(n)$ refers to the maximum number of marks in relation to the length of the ruler.

4.2.1 Equality relations

First, we will consider equality relation between the two functions. We will prove two lemmas that will help us later prove our two main theorems for inequality relations.

Notice that $\mathcal{B}_2$ sequences do not necessarily begin at 1 or end at n. However since every $\mathcal{B}_2$ sequence is a Golomb ruler, the two properties of translation and mirroring also apply. By the translation $A' = A - min\{a_i\}$ we get a Golomb ruler that begins at position 0 from any $\mathcal{B}_2$ sequence.

First suppose that we know the exact value of $F_2(d)$ for some d. The following lemma tells us what we can learn about $G(n)$.

Lemma 4.1. *For every d, if*

$$F_2(d) = n \quad \Longleftrightarrow \quad \begin{array}{l} G(n) \leqslant d-1 \\ G(n+1) > d-1 \end{array}$$

Proof. If the maximum size of a Sidon set contained in $\{1, \ldots, d\}$ is n then there is a Golomb ruler with k marks that has elements

$$A' = \{a_i - min(a_i) \quad \forall a_i \in A\}$$

where A is the Sidon set and $min(a_i)$ it's minimum element which is at least 1. This ruler begins at position 0 and has length $\leqslant d-1$ thus $G(n) \leqslant d-1$.

Also, since at most n marks can be placed in $\{1, \ldots, d\}$, also at most n marks can be placed in $\{0, \ldots, d-1\}$. It follows that $G(k+1) > d-1$.

For the opposite direction notice that if the two inequality relations hold for $G(n)$ then at most n marks can be placed in $\{0, \ldots, d-1\}$ (and $G(n) \leqslant d-1$ guarantees that such placement exists) or equivalently in $\{1, \ldots, d\}$, so $F_2(d) = n$. $\square$

Inversly, if we know for some n an exact value of $G(n)$, the following lemma will help us find two exact values for $F_2(d)$.

Lemma 4.2. *For every n, d, if*

$$G(n) = d \quad \Longleftrightarrow \quad \begin{array}{l} F_2(d) = n-1 \\ F_2(d+1) = n \end{array}$$

Proof. If $G(n) = d$ then there exist a Golomb ruler contained in the set $\{0, 1, \ldots, d\}$ with n marks but no such ruler exists in $\{0, 1, \ldots, d-1\}$.

In turn, by the translation property a Sidon set with n elements exists contained in $\{1, 2, \ldots, d+1\}$ which means that $F_2(d+1) \geqslant n$. But $F_2(d+1)$ cannot be $n+1$ as it would imply that $G(n+1) = d$ and G is strictly increasing so $F_2(d+1) = n$.

Also, since $G(n) = d$ and G is monotonically increasing (lemma 2.2), we have that $G(n-1) \leqslant d-1$ and $G(n) > d-1$. These two inequalities and lemma 4.1 imply that $F_2(d) = n-1$.

For the opposite direction, $F_2(d+1) = n$ implies that n marks can be placed in $[1, d+1]$ or equivalently in $[0, d]$. By $F_2(d) = n-1$, n marks cannot be placed in $[0, d-1]$ and consequently $G(n) = d$. $\square$

It is evident that the two functions can provide us with the same information of the properties of the Sidon sets and Golomb rulers. In some sense, exact values of G are stronger than exact values of F_2, since the former transform directly into two exact values of F_2 while the opposite does not hold. However, complete knowledge of the values of one function, either G or F_2, provides the values of the other function.

4.2.2 Inequality relations

Less that 25 exact values are known for G or F_2. Most results concerning these two functions are in the form of lower and upper bounds for the exact value. To be able to translate these results between the two problems we must translate inequality relations between F_2 and G.

First consider the case that we know a lower or an upper bound for $F_2(d)$. The following two lemmas will help in bound $G(n)$.

Lemma 4.3. *For every n, d if*

$$F_2(d) > n \quad \Longrightarrow \quad G(n) < d-1$$

Proof. Suppose $F_2(d) = n' > n$. Then by lemma 4.1: $G(n') \leqslant d-1$. Since $n' > n$ and $G(n)$ is monotonically increasing by lemma 2.2, we have that $G(n') > G(n)$ so

$$G(n) < G(n') \leqslant d-1$$

$\square$

Lemma 4.4. *For every n, d if*

$$F_2(d) < n \quad \Longrightarrow \quad G(n) > d - 1$$

Proof. Suppose $F_2(d) = n' < n$. Then by lemma 4.1: $G(n' + 1) > d - 1$. Since $n \geqslant n' + 1 \Rightarrow G(n) \geqslant G(n' + 1)$ so

$$G(n) \geqslant G(n' + 1) > d - 1$$

$\square$

These two lemmas enable us to bound G using known bounds for F_2. Suppose that we have a function $l(d)$ that bounds F_2 from below: $F_2(d) > l(d)$ for each d. Usually $l(d)$ will be a monotonically increasing function so it admits an inverse function $l^{-1}(n)$, so that $l^{-1}(l(d)) = d$. Then if for all d:

$$F_2(d) > l(d) \implies G(l(d)) < d - 1 \implies$$
$$G(n) < l^{-1}(n) - 1 \tag{4.2}$$

Accordingly, if $F_2(d) < u(d)$ for each d and $u^{-1}(n)$ exists then $G(n) > u^{-1}(n) - 1$

We have proved that

Theorem 4.5. *Suppose $l(d)$ and $u(d)$ are well-defined and admit inverse functions $l^{-1}(n)$ and $u^{-1}(n)$ inside an interval $A \subseteq \mathbb{N}$. If*

$$l(d) < F_2(d) < u(d)$$

then

$$u^{-1}(n) < G(n) + 1 < l^{-1}(n)$$

Now consider the opposite case where we know some bound for $G(n)$ and wish to bound $F(d)$.

Lemma 4.6. *For every n, d if $G(n) < d$ then $F_2(d) \geqslant n$*

Proof. Since $G(n) < d$, there exists a Golomb ruler with n marks having $0 = a_1 < a_2 < \ldots < a_n < d$.

The $\mathcal{B}_2$ sequence $b_i = a_i + 1$ is contained in $[1, d]$ since $b_n \leqslant d$ so $F_2(d)$ is at least n. □

Lemma 4.7. *For every n, d if $G(n) > d$ then $F_2(d) \leqslant n$.*

Proof. By the hypothesis, in the set $\{0, \ldots, d\}$ we cannot select n marks. That implies that the maximum number of integers we can select from $\{1, \ldots, d+1\}$ is less that n. Then, $F_2(d+1) < n$ and also $F_2(d) \leqslant n$ since F_2 is non-decreasing. □

The following theorem unites the two lemmas

Theorem 4.8. *Suppose $l(n)$ and $u(n)$ are well-defined and admit inverse functions $l^{-1}(d)$ and $u^{-1}(d)$ inside an interval $A \subseteq \mathbb{N}$. If*

$$l(n) < G(n) < u(n) \qquad \textit{then} \qquad u^{-1}(d) \leqslant F_2(d) \leqslant l^{-1}(d)$$

4.3 Improving lower bounds of $G(n)$

By using the upper bound we proved for F_2 we can find a better lower bound for the optimal length of Golomb rulers.

Theorem 4.9. *For all n*

$$G(n) > n^2 - 2n\sqrt{n} + \sqrt{n} - 2$$

Proof. By theorem 3.1 we know that $F_2(d) < d^{1/2} + d^{1/4} + 1$. Consider the function $u(d) = d^{1/2} + d^{1/4} + 1$. This function is monotonically increasing in the interval $(0, \infty)$ and we can find it's inverse by the substitution $d = y^4$. Solving for y we have that

$$y = \sqrt{n - 3/4} - 1/2$$

and consequently

$$u^{-1}(n) = \left(\sqrt{n - \tfrac{3}{4}} - \tfrac{1}{2}\right)^4 \tag{4.3}$$

$$= n^2 - 2n\sqrt{n - \tfrac{3}{4}} + \sqrt{n - \tfrac{3}{4}} - \tfrac{1}{2} \tag{4.4}$$

$$\tag{4.5}$$

Using theorem 4.5 we find that

$$G(n) > n^2 - 2n\sqrt{n - \tfrac{3}{4}} + \sqrt{n - \tfrac{3}{4}} - \tfrac{3}{2} \tag{4.6}$$
$$> n^2 - 2n\sqrt{n} + \sqrt{n - \tfrac{3}{4}} - \tfrac{3}{2} \tag{4.7}$$

By the inequality $\sqrt{n - \frac{3}{4}} \geqslant \sqrt{n} - \frac{1}{2}$ for $n \geqslant 1$, we conclude that

$$G(n) > n^2 - 2n\sqrt{n} + \sqrt{n} - 2$$

□

This is an improvement over the trivial lower bound $G(n) \geqslant \frac{1}{2}n(n-1)$ for $n > 13$. As n becomes larger the gap between the two bounds increases.

In figure 4.3 the bound we proved is compared against the trivial lower bound $\frac{1}{2}n(n-1)$ and the known optimal values of the length of Golomb rulers for $n < 24$.

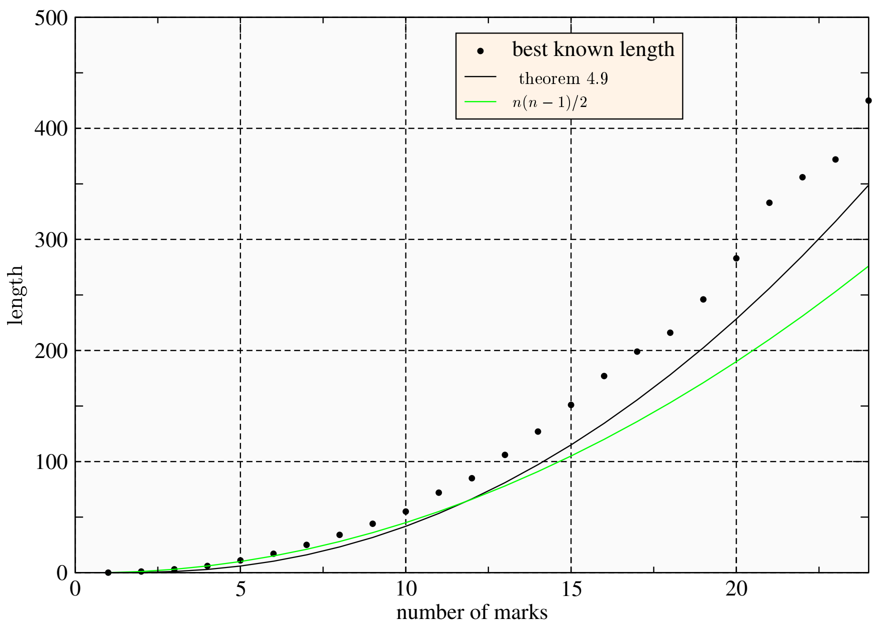


Figure 4.1: Lower bounds known versus known optimal ruler sizes

4.4 Summary

Golomb rulers and Sidon sets are closely related: both have equivalent definitions. In this chapter we discussed the relation between $F_2(n)$ and $G(n)$, the two functions that give the best possible size of a Sidon set or length of a Golomb ruler and proved theorems that allow the restatement of known or future results on $G(n)$ or $F_2(n)$ to the other problem. Using these results we have derived a better lower bound for $G(n)$ which is contained in theorem 4.3.

Chapter 5

Constructions for Golomb rulers

When the number of marks exceeds 25, brute-force algorithms that search all the possible rulers do not stand a chance of finding a near-optimal ruler with size less that n^2. In this case one has to resort to algebraic constructions that generate sequences of integers having *a priori* the desired properties.

In this chapter we will consider constructions that generate Golomb rulers with large number of marks and size near n^2. All of our construction will be based on the properties of finite fields.

We will first introduce basic facts from number theory and finite fields for the reader which is not acquainted with the subject and then we will describe the constructions.

The main constructions we will use are the Bose-Chowla construction described in [11] and I. Ruzsa [50] construction as extended by Lindström[43].

5.1 Introduction to number theory

Before we begin describing the constructions for Golomb rulers (and Sidon sets) we will first give a quick review of the number theoretic facts we will use from now on. For a more complete treatment see the classic text in number theory by Hardy and Wright[30].

The reader who is acquainted with number theory and finite fields can skip this section.

5.1.1 Prime numbers and Euler's ϕ fucntion

An integer $a > 1$ is **prime** if it has no other divisors but 1 and a. Primes have many special properties and play a critical role in number theory. The first few primes are:

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47$$

If a number is not prime it is a **composite** number. For example 39 is composite as $39 = 3 * 13$. The integer 1 is neither a prime nor a composite and is called a **unit**. Similarly 0 is neither prime or composite.

For every two integers a, b we can find the largest integer that divides both of then. This integer is called the **greatest common divisor** of a and b and is symbolized as $gcd(a, b)$ or more commonly just (a, b).

If a, b do not share common divisors larger that 1 then they are **relative prime** and $(a, b) = 1$. For example, 8 and 15 are relative prime since no integer larger than 1 divides both of them.

For a given number n, the number of integers, smaller than a, that are relative prime to n is symbolized as $\phi(n)$, the **Euler's phi function**. By $|S|$ we denote the number of elements of set S.

$$\phi(n) = |\{a < n \quad \text{and} \quad gcd(a, n) = 1\}|$$

For example $\phi(10) = 7$ since the integers that do not share a common divisor with 10 are $7 : 1, 3, 4, 6, 7, 8, 9$.

Note that if p is a prime number then neither of the integers $1, 2, \ldots, p-1$ has a common divisor with p so

$$\phi(p) = p - 1 \quad \text{if } p \text{ is prime}$$

5.1.2 Integer division

Elementary mathematics state that every integer a when divided by b has an unique integer quotient q and remainder r, so that $a = q \cdot b + r$. For example 26 divided by 3, gives $q = 8$ and $r = 2$: $26 = 8 \cdot 3 + 2$.

Theorem 5.1 (Division). *For every integer a and b, there are unique* *integers q and r so that $0 \leqslant r < b$ and*

$$a = q \cdot b + r.$$

Proof. Suppose that there exist two different solutions

$$a = q_1 b + r_1 = q_2 b + r_2.$$

q_1 must be different from q_2, so $(q_1 - q_2)b = r_1 - r_2$ must be a non-zero multiple of b. Consequently it is greater than b in absolute value.

We must have that $|r_2 - r_1| \geqslant b$, a contradiction since $-b < r_2 - r_1 < b$ by $0 \leqslant r_1, r_2 < b$. $\square$

The integer q is called the **modulus** of a when divided by b and as we shown it belongs to the interval $[0, b-1]$. Usually we denote it as

$$a \bmod b = q \quad \text{or} \quad (a)_b = q$$

If $q = 0$ then b divides a and we write $b|a$. Of course if $b|a$ then $a \bmod b = 0$.

When two integers a_1,a_2 share the same modulus when divided by b we say that they are equivalent modulo b and denote it as

$$a_1 \equiv a_2 \bmod b$$

For example $45 \equiv 25 \bmod 20$.

The modulus has some important properties that will be very useful from now on:

$$(a)_n + (b)_n \equiv a + b \bmod n \tag{5.1}$$

$$(a)_n \cdot (b)_n \equiv a \cdot b \bmod n \tag{5.2}$$

We can define addition and multiplication modulo some integer n as normal addition and multiplication but taking the modulo n of the results. Sometimes the symbols $+_n$ and $\cdot_n$ will be used to denote such operations. Subtraction $-_n$ can be defined likewise. For example $2 +_9 17 = 1$ and $2 \cdot_5 10 = 0$.

These two operations and especially multiplication modulo p will play a central role from now on so one must have a good grasp of the theory. We will now define more formally these operations.

Define the set $Z_n = \{0, 1, \ldots, n-1\}$. We can think Z_n as the set of all possible remainders modulo n. Suppose we have a, $b \in Z_n$. Then also $a +_n b$ is in Z_n and always using $+_n$ we will never get outside of the set Z_n. Forgetting the usual addition, if we always use addition modulo n then say that we belong to a **group**, more specifically the group of Z_n under the operation of addition modulo n. It is symbolized as group $Z_n^+ = (Z_n, +_n)$, the **additive group modulo n**.

More generally a group is a set S together with an operation $\odot$ defined on S and is symbolized as $(S, \odot)$. The following properties must hold for a group:

1. **Closure:** For all $a, b \in S$, we have $a \odot b \in S$.
2. **Identity:** There exist an element $e \in S$, called the identity of the group, such that $e \odot a = a \odot e = a$ for all $a \in S$.
3. **Associativity:** For all $a, b, c \in S$, we have $(a \odot b) \odot c = a \odot (b \odot c)$.
4. **Inverse element:** For each $a \in S$, there exists a unique element $b \in S$, called the **inverse** of a, such that $a \odot b = b \odot a = e$.

The size of a group is the number of elements in it's defining set. In this case the number of elements in $(Z_n, +_n)$ is $|Z_n| = n$.

5.1.3 The multiplicative group modulo n

Another group, more interesting for our discussion, is the **multiplicative group modulo n**, $(Z_n^*, \cdot_n)$, where the operation is multiplication modulo n. It can be proved that 0 cannot be an element of the group and neither can be any integer that has a common divisor with n or else there would not be a unique inverse for each element .

So, only integers that are relative prime to n can be in this group. For example $Z_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$. By the definition of the Euler's phi function, the size of the group is $|Z_n^*| = \phi(n)$.

We will only consider multiplicative groups with n being a prime number. If p is prime then $Z_p^* = \{1, 2, \ldots, p-1\}$ and $|Z_p^*| = p - 1$.

As multiplication is defined, we can also define powers of an element a of Z_p^* as

Table 5.1: Powers of 2 and 3 in Z_{13}^*

n	0	1	2	3	4	5	6	7	8	9	10	11	12
2^n	1	2	4	8	3	6	12	11	9	5	10	7	1
3^n	1	3	9	1	3	9	1	3	9	1	3	9	1

$$a^k = \underbrace{a \cdot_p a \cdot_p a \ldots a}_{k \text{ times}}$$

Define also $a^0 = 1$. The usual law of exponents holds: $a^k\ a^l = a^{k+l}$.

Since there are only $\phi(n)$ different elements in the multiplicative group, the powers of a must repeat inevitably. Fermat's theorem and Euler's generalization state this fact.

Theorem 5.2 (Euler). *For every element a of Z_n^*,*

$$a^{\phi(n)} \equiv 1 \pmod{n}.$$

Corollary 5.3 (Fermat). *If p is prime, then*

$$a^{p-1} \equiv 1 \pmod{p}.$$

So, if p is prime then $a^{p-1+k} = a^{p-1}a^k = a^k$ that is, the powers of a repeat after, at most, $p-1$ steps. In the following table we list the powers of 2 and 3 $\pmod{13}$.

Notice that the powers of 3 repeat after 3 steps but the powers of 2 generate all the integers $1, 2, \ldots, 12$ in a permutation. The order $ord_p a$ of an element is the period of it's powers in Z_p^*. In this case $ord_{13} 3 = 3$ and $ord_{13} 2 = 12$.

Definition (order). The **order** of an element g of Z_p^* is the least positive integer z for which $g^z \equiv 1 \pmod{p}$. We will use the notation $ord_p g$ to refer to the order of element g in Z_p^*.

We can classify the elements of Z_p^* in two categories: If the powers of a cycle through all elements of group Z_p^* or equivalently if $ord_p a = p - 1$ then a is called a **primitive element** of Z_p^*. On the other hand, if $ord_p a < p - 1$ then a is a not primitive.

Definition (primitive element). If $g^i \equiv 1 \pmod{p}$ does not hold for $1 \leqslant i \leqslant p-1$, or equivalently $ord_p g = p-1$, then g is called a **primitive element** of Z_p^*.

Primitive elements are also called **generators** of Z_p^* since the powers of g generate all the elements of the field. For a primitive element the powers $g^0, g^1, \ldots, g^{p-1}$ are a permutation of $1, 2, \ldots, p-1$.

An important property of primitive elements is that

$$g^i = g^j \quad \Longleftrightarrow \quad i = j \tag{5.3}$$

for $i, j < p-1$, since the elements g^i are unique.

In our example 2 is a primitive element as the values of 2^n for $1 \leqslant n \leqslant 12$ are exactly the integers $1, 2, \ldots 12$. On the other hand the powers of 3 repeat with period 3 and it is not a primitive element.

To find primitive elements the following will be useful.

Lemma 5.4. *The order of an element g of Z_p^* divides $p-1$.*

Proof. We know, by Fermat's theorem 5.3, that $g^{p-1} \equiv 1 \pmod{p}$. Let t be the order of g so $g^t \equiv 1 \pmod{p}$.

Suppose t does not divide $p-1$, then by the division theorem

$$p - 1 = u \cdot t + v$$

with $v \neq 0$ and $v < t$. Now we have that:

$$g^{p-1} \equiv g^{ut} g^v \equiv (g^t)^u g^v \equiv 1^u g^v \equiv g^v$$

Since $g^{p-1} \equiv 1 \pmod{p}$ then we have that $g^v \equiv 1 \pmod{p}$. This is a contradiction because $v < t$ and we assumed t is the least positive integer for which $g^t \equiv 1 \pmod{p}$. □

Corollary 5.5. *If $g^i \equiv 1 \pmod{p}$ does not hold for $1 \leqslant i \leqslant \lfloor \frac{p-1}{2} \rfloor$ then g is a primitive element of Z_p^*.*

Proof. By 5.4, if $g^i \not\equiv 1 \pmod{p}$ for $i \leqslant \lfloor \frac{p-1}{2} \rfloor$ then the order of g is greater than $\frac{1}{2}(p-1)$. Since there are no numbers greater than $\frac{1}{2}(p-1)$ that divide $p-1$ before $p-1$ itself, the period of g is $p-1$. □

Later, we will be interested in findind all the primitive roots of Z_p^*. Instead of trying every element of Z_p^* to see if it's order is $p-1$, we can use the following lemma to find all the primitive elements after we have found the first. The proof is omitted.

Lemma 5.6. *If g is a primitive root of Z_p^* then g^n is also a primitive root if and only if $gcd(n,p-1)=1$.*

By using this lemma, we can find the exact number of primitive roots in Z_p^* which is the number of integers n such that $gcd(n,p-1)=1$.

Lemma 5.7. *There are exactly $\phi(p-1)$ primitive roots in the group Z_p^*.*

5.1.4 Finite fields

We have described the multiplicative group Z_n^* for any n. When n is a prime number p then the elements of the group are all the positive integers 1,2,...,$p-1$ since none of them has a common divisor with p.

Groups with prime number of elements have some additional properties and they have a special name, **finite fields**. The theory of finite fields began with the work of Carl Friedrich Gauss (1777-1855) and Evariste Galois (1811-1832) but it only became of interest for applied mathematicians in recent decades with the emergence of discrete mathematics and the applications in cryptography and other areas. In honor of Galois the finit field with p elements is denoted as $GF(p)$.

It can be proved that finite fields exist not only for prime numbers, but also for any power of a prime number p^n. The field $GF(p^n)$ is called an extension field of $GF(p)$. When $n>1$, extension fields are difficult to describe and we will not continue the discussion here. For a complete treatment of finite fields see [41] or any modern algebra book.

5.2 A simple construction

Before we describe more subtle constructions, we will first see a simple one which follows from elementary mathematics.

Construction 1. *Let n be any positive integer. Then the sequence*

$$\Phi_0(a) = 2n\,a^2 + a \quad , \quad 0 \leqslant a < n$$

forms a Golomb ruler.

Proof. Suppose we have the sum of two elements

$$\Phi_0(a) + \Phi_0(b) = 2n(a^2 + b^2) + (a + b) \tag{5.4}$$

Consider the division of $\Phi_0(a) + \Phi_0(b)$ by $2n$. By the division algorithm there exist unique integers q, r with $0 \leqslant r < 2n$ such that

$$\Phi_0(a) + \Phi_0(b) = 2n\, q + r \tag{5.5}$$

But since $0 \leqslant a + b < 2n$, by equation 5.4 we have $q = a^2 + b^2$, $r = a + b$. This system has a unique solution (up to permutation) for a, b,

$$\{a, b\} = \frac{1}{2}(r \pm \sqrt{2q - r^2})$$

and so there cannot be two different pairs of elements which have the same sum.

□

With a more complex argument, using the differences instead of the sums, it can be shown that dropping the 2 from equation 5.4 will also result in a Golomb ruler.

Construction 2. *Let n be any positive integer. Then the sequence*

$$\Phi_1(a) = n\, a^2 + a \quad , \quad 0 \leqslant a < n$$

forms a Golomb ruler.

The largest element of this sequence is $n^3 - 2n^2 + 2n$ which implies that for all n an upper bound for the optimal length of a Golomb ruler is

$$G(n) \leqslant n^3 - 2n^2 + 2n.$$

Computing these two construction is straightforward and takes time $O(n)$ for every n. In fact the computations can be arranged so that $\Theta(n)$ additions and multiplications are necessary.

5.3 Erdős and Turan construction

Erdős and Turan [25], have given the first construction which lowers the bound and provides rulers with size $\Theta(n^2)$.

Unfortunately this construction and all constructions which produce rulers of size $\Theta(n^2)$ do not work for all choices of prime numbers but only for primes (or prime powers).

Construction 3. *For every prime number p the following sequence forms a Golomb ruler*

$$2p\,a + (a^2)_p \quad , \quad 0 \leqslant a < p$$

This construction was used by Erdős to prove the first upper bound on Sidon sets and consequently Golomb rulers. For for p prime, it is implied by the construction that

$$G(p) \leqslant 2p^2 - p - 1$$

.

Again, it is easy to compute the elements of this construction in time $O(p)$ using $\Theta(p)$ multiplications and additions.

Unfortunately this construction does not produce Golomb rulers of size less than n^2, so we will not be able to use it for out purposes.

5.4 Ruzsa construction

The first construction we will discuss that gives rules of size near n^2 was given by I.Z. Ruzsa in [50].

It is a very fast construction which gives Golomb rulers with $p-1$ elements for every prime number p and size $p^2 - p$.

The computations used are straightforward and can be developed to a very efficient algorithm that is able to find sequences with more than 10000 marks.

We will use some more complicated arguments in this proof regarding finite fields. For the related theorems, see any basic treatment of finite fields like [41].

Construction 4 (Ruzsa). *Let p be a prime number and g a primitive element of the multiplicative group Z_p^*. The following sequence is a Golomb ruler.*

$$R(p,g) = pi + (p-1)g^i \ mod\ p(p-1) \quad for\ \ 1 \leqslant i \leqslant p-1$$

Proof. Let

$$p(i+j) + (p-1)(g^i + g^j) \equiv a \pmod{p}(p-1)$$

be the sum of two elements. Then we can find that

$$g^i + g^j \equiv -a \pmod{p} \tag{5.6}$$

$$i + j \equiv a \pmod{p-1} \tag{5.7}$$

By Fermat's little theorem 5.3 that we proved earlier in page 39:

$$g^i g^j \equiv g^a \pmod{p} \tag{5.8}$$

By 5.6 and 5.8, g^i and g^j are the roots of the quadratic polynomial $X^2 + aX + g^a$ in $GF(p)$, so

$$X^2 + aX + g^a = (X - g^i)(X - g^j).$$

From the uniqueness of the factorization of a quadratic polynomial in $GF(p)$ we infer the uniqueness of g^i, g^j and consequently of i,j up to a permutation. □

For an example when $p = 7$, the primitive elements elements of Z_7^* are 3 and 5. Then

$$R(7,3) = \{6, 10, 15, 23, 25, 26\}$$

$$R(7,5) = \{6, 11, 15, 37, 38, 40\}$$

Lindstrom [43] has proved that if f is an integer relative prime to $p-1$ then the following is also a Golomb ruler:

$$R'(p,g,f) = pfi + (p-1)g^i \ \mathrm{mod}\ p(p-1) \quad for\ \ 1 \leqslant i \leqslant p-1$$

He has also proved in the same paper that by varying the primitive element and f, one does not produce new Golomb ruler but a translation of the original one multiplied by an integer modulo $p(p-1)$. Since the extended construction is equivalent to the original one, we will not consider this extension.

The $p-1$ integers in $R(p, f)$ are reduced modulo $p(p-1)$ and the largest of them is smaller than $p(p-1)$. The following bound for $G(n)$ follows

Lemma 5.8. *$G(n) < n^2 + n$ whenever n+1 is prime*

The computation of $R(p, g)$ depends on finding a primitive element of the appropriate finite field. The construction assumes that a primitive element of the associate finite field can be found fast. We will see in the next chapter that this actually holds and the time of computing all the primitive elements of Z_p^* can be found in time neglible to the other operations.

Assuming that a primitive element is found, the algorithm will take time $O(p)$ to find all the elements of a Golomb ruler. However the elements will not be necessary in increasing order so a sorting algorithm has to be used.

Linear time sorting algorithms exists [4], so the total time to produce a Golomb ruler in sorted order can be $O(p)$. However the constants involved in linear time sorting algorithms are large and classical sorting approaches are more efficient. We will use a classical comparison based sort like mergesort [16] to provide an upper bound of $O(p \log p)$ for the total running time.

5.5 Singer Perfect Difference sets

Singer [53] has proved that if q is a power of a prime then we can find $q+1$ integers which have distinct differences modulo q^2+q+1 and thus form a Golomb ruler. Singer's construction depends on the evaluation of extensions of Galois fields, actually $GF(p^3)$ the 3rd order extension of the multiplicative field Z_p^*.

We will omit the proofs of this construction (as well as the next one); they are quite complicated and beyond the scope of this discussion.

The reader who wishes to find more can refer to [41] or any modern algebra book for a general discussion of the properties of finite fields of higher order. The original proof is in [53].

Construction 5 (Singer). *Let $q = p^n$ be a prime power. There there exist $q+1$ integers*

$$d_0, d_1, \ldots, d_q$$

such that the q^2+q differences $d_i - d_j (i \neq j)$ when reduced modulo q^2+q+1, are all the different non-zero integers less than q^2+q+1.

This implies that whenever $n-1$ is a prime power then, by the substitution $q+1=n$, $G(n) < n^2$.

Corollary 5.9. *$G(n) < n^2 - n + 1$ whenever $n-1$ is a prime power.*

The time required to compute Singer's construction is of the order $O(p^3)$, quite prohibitive for use in our computations.

5.6 Bose-Chowla theorem

Bose [11] and Chowla [12] have proved that for any prime power p^n there exist p^n integers which form a Golomb ruler modulo $p^{2n}-1$.

Construction 6 (Bose). *Let $q = p^n$ be a power of a prime and θ a primitive element in the Galois field $GF(q^2)$. Then the q integers*

$$d_1, \ldots, d_q = \{a : 1 \leqslant a < q^2 \;\; and \;\; \theta^a - \theta \in GF(q)\} \tag{5.9}$$

have distinct pairwise differences modulo q^2-1.

In addition the $q(q-1)$ differences $d_i - d_j$, $i \neq j$, when reduced modulo q^2-1, are all the different non-zero integers less than q^2-1 which are not divisible by $q+1$.

For example when $q = 11$, $\theta = 2x+3$ is a primitive element of $GF(11^2)$. The following sequence which is generated by equation 5.9

$$1, 6, 20, 27, 38, 40, 55, 65, 71, 117, 118$$

has distinct sums and differences modulo 120.

The largest element of this sequence is smaller than q^2-1, so Bose-Chowla theorem implies that whenever $q = p^n$, a Golomb ruler with size less that the square of the number of marks exists.

Corollary 5.10. *$G(q) < q^2 - 1$ whenever q is a prime power.*

We will use the construction of *Bose* later for the case of prime numbers ($n = 1$) to find near-optimal Golomb rulers. Assuming that one can find a primitive element of the field $GF(q^2)$ fast usually by a randomized algorithm, computing the sequence takes time $O(q^2)$.

5.7 Shifting and multiplying a construction

The three constructions we described that produce rulers of size near n^2, work only for a prime (or power of a prime) number of marks. From these constructions we will be using two transformations similar to the ones of Golomb rulers we discussed in chapter 2: addition and multiplication.

Using these two properties one can find new Golomb rulers from the given ones which also have about the same size. Then one can remove some elements of the sequence to get rulers with a smaller number of marks for the purpose of covering the gaps between prime numbers where no construction for Golomb ruler exists. We will be using this properties to prove that $G(n) < n^2$ for all n smaller than a limit and not just prime numbers.

Note that having different pairwise sums modulo some integer z is somewhat stronger than just having different pairwise sums. This is the case as between two elements $a_i > a_j$ there exist two distances. The first is the positive difference $a_i - a_j$ like the case of Golomb rulers. The other one is $a_j - a_i$ which is also positive mod z and must be different from all other distances.

To illustrate this, modular constructions can be thought as points on the circumference of a circle. See for example figure 5.7 where a sequence constructed by Bose-Chowla theorem $\{1, 2, 5, 11, 31, 36, 38\} \mod 48$ is visualized in the circumference of the circle and then unwound to form a Golomb ruler of length 38.

The arc distance between any two points in a modular construction must be different. There are $n(n-1)$ different arc lengths because between each two elements there are two distances, one clockwise and one counterclockwise. So compared to $\frac{1}{2}n(n-1)$ different distances for a Golomb ruler, it is twice as hard to find a modular construction.

In a modular construction the important property is the modulus z. The

$\{1, 2, 5, 11, 31, 36, 38\} \mod 48$

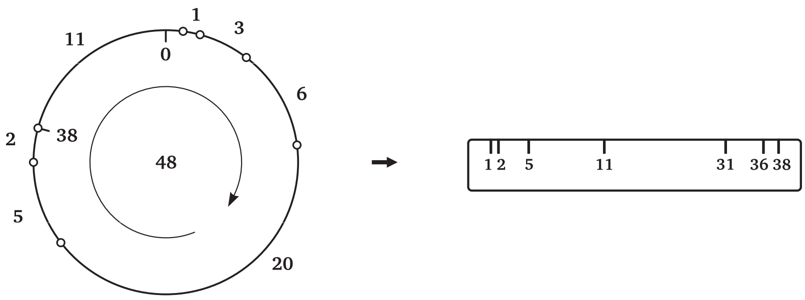


Figure 5.1: Unwounding a modular construction to form a Golomb ruler

modulus is analogous to the length of the Golomb ruler in linear terms. A modular construction with n marks must have modulus at least $n(n-1)$ to ensure that there are that many different integer arc lengths.

5.7.1 Addition

Suppose that we have generated, using one of the aforementioned constructions, a set with elements $\{a_1, a_2, \ldots, a_n\}$ that has distinct sums and differences modulo some integer z.

Adding some integer k to each element and then reducing modulo z will provide us with a different Golomb ruler as the following property states. In our visualization, it corresponds to rotating all the elements together.

Property 3 (Translation). *The following sequence has distinct sums for every integer k*

$$\{a_i + k \;\; mod \;\; z \;,\; 1 \leqslant i \leqslant n\}$$

Proof. If there existed two equal pairwise sums

$$(a_i + z) + (a_j + z) \equiv (a_k + z) + (a_l + z) \mod z$$

then by elementary properties of the modulus we discussed

$$a_i + a_j \equiv a_k + a_l \mod z$$

which contradicts the property of the original sequence. □

By varying the element k, from 0 up to $z-1$ we can get z different Golomb rulers. Using other values of k would not give us any different rulers since adding k to each element is equivalent to adding k mod z.

The importance of these new rulers is that in general the maximum element of this ruler will be different when varying k:

$$\max_i\{a_i\} \neq \max_i\{a_i + k\}$$

In general to obtain the best ruler, one has vary k and find the minimum of all the maximum elements obtained by different choices of k:

$$\min_k \max_i\{a_i + k\}$$

We can find which value of k will lead to the shortest Golomb ruler. If the smallest mark of the ruler after a translation a_1 is greater 0 one could shift the ruler left by a_1 to find a shorter one. That means only the cases of k when the smallest element is 0 are meaningful. In this case the largest element is is $z - d_i$ where $d_i = (a_{i+1} - a_i)_z$ is the set of the distances between consecutive elements. The minimum for the largest element occurs when d_i is maximum that is

$$\min_k \max_i\{a_i + k\} = z - max\{a_{i+1} - a_i \pmod z\}$$

where we assume that the each subscript a_x is reduced modulo z: $a_{i+z} = a_i$.

Generalizing, when one has to remove k elements from the construction to form a Golomb ruler, the maximum difference $a_{i+1+k} - a_i$ must be found. We will use the following lemma to find the minimal maximum element of the sequence in this case.

Lemma 5.11. *If, for a construction with n elements mod z, we have to remove $k \geqslant 0$ elements then the minimal length of the sequence that can be produced by the translation property is*

$$z - \max_{1 \leqslant i \leqslant n}\{a_{(i+1+k)_n} - a_i \pmod z\}$$

Intuitevely, the way to obtain the shortest possible Golomb ruler is to shift the maximum of the differences between consecutive elements $a_{i+1} - a_i$ and shift this difference so that it is placed after the largest mark of the

$$\{1, 2, 5, 11, 31, 36, 38\} \mod 48$$

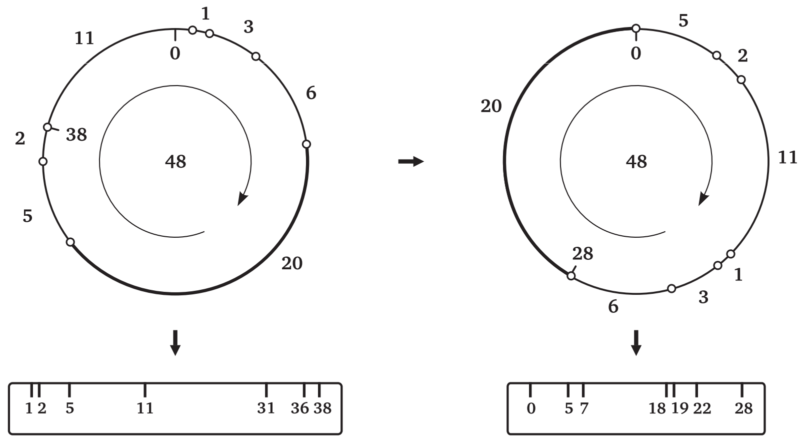


Figure 5.2: Forming a shorter ruler by shifting and truncating

ruler and the end of the modulus.

5.7.2 Multiplication

Consider again a construction $\{a_1, a_2, \dots, a_n\}$ that has distinct sums and differences modulo some integer z. The second important transformation that allows to generate new Golomb rulers modulo z from a given one is the multiplication with an element relative prime to z.

Property 4 (Multiplication). *The following sequence has distinct sums for ever integer c with $gcd(c, z) = 1$*

$$\{c \cdot a_i \ mod \ z \ , \ 1 \leqslant i \leqslant n\}$$

Proof. If there existed two equal pairwise sums

$$(ca_i) + (ca_j) \equiv (ca_k) + (ca_l) \mod z$$

then by elementary properties of the modulus we discussed we can drop c from both sides of the equation, so

$$a_i + a_j \equiv a_k + a_l \mod z$$

which contradicts the property of the original sequence. □

If $gcd(c, z) \neq 1$ then we cannot drop c and the generated sequence will not be a Golomb ruler.

Again, by varying element c one can produce new Golomb rulers. There is no general way to know before multiplying which value will give the shortest Golomb ruler and one has to cycle through all values of c. Altogether there are $\phi(z)$ values of c to consider by the definition of Euler's function.

5.8 Summary

In this chapter we discussed various known constructions for Golomb rulers and Sidon sets. These constructions yield near-optimal sets, although many of the sets generated by these constructions were proved to be optimal.

The important property that the three most important constructions (Singer, Bose-Chowla and Ruzsa) have is that they are modular: the sums or differences between pairs of elements are distinct modulo some integer z.

This property allows us to use two similarity transformations to generate new Golomb rulers from a given one. Using these two transformations one can often improve the length of a given Golomb ruler produced from a modular construction.

A downside is that the constructions that yield near-optimal rulers apply only to prime or prime power number of marks. However by omitting some elements of a set one can fill the gap between the prime numbers and yield near-optimal Golomb rulers for any number of marks.

Chapter 6

Algorithms for near optimal Golomb rulers

Today near-optimal Golomb rulers of sizes less than n^2 for n smaller than 150 are known. Most of these rulers have been found using the constructions we described in chapter 5 and suitable transformations using the two properties discussed.

In this chapter we will be interested in finding quick algorithms that can continue this search and find near-optimal Golomb rulers with sizes less than n^2 for a number of marks up to 65000.

Our aim is to prove that Golomb rulers with size less than n^2 exist for every number of marks up to 65000. To prove this statement rulers of such size must be exhibited for all n. For this purpose, use of constructions described in chapter 5 is needed.

The following theorem will be the mail goal of this and the following chapter. The proof will be computational.

Theorem 6.1 (Main Theorem). *Golomb rulers of size less than n^2 exists for all $n < 65000$. Equivalently*

$$G(n) < n^2 \quad \textit{for } n < 65000$$

or in Sidon set terms

$$F_2(d) > n^{1/2} \quad \textit{for all } d < 4.2 \cdot 10^9$$

6.1 Old results

A number of conjectures on the growth of the functions F_2 and G exist. Originally, most of the conjectures were introduced for the Sidon problem.

An old conjecture by Erdős states that $G(n) < n^2$, that is Golomb rulers with sizes less that n^2 exist for all n.

This has not been proven up to now despite efforts by many mathematicians. The closest form that has been proven is that $G(n)$ asymptotically approaches n^2 as n tends to infinity. It has been shown by Chowla[14] and Erdős[25, Addendum] that

$$F_2(n) \geqslant n^{1/2} - o(n^{1/2})$$

which implies by the results of chapter 4 that

$$G(n) \leqslant n^2 + o(n^2).$$

Computationally it has been proven that $G(n) < n^2$ for all $n < 150$ using Golomb rulers constructed by Singer and Bose constructions. The computations for Golomb rulers have been done by Alex W. Lam and Dilip V. Sarwate [40] and for Sidon sets by Zhang [54].

6.2 Choosing a construction to use

In chapter 5 we discussed various constructions for Golomb rulers which yield near or not so near optimal rulers. We will examine the suitability of these constructions to use in the proof of theorem 6.1 and choose the construction we will use.

Constructions 1 and 2 produce rulers of size n^3, quite above the bound we wish to prove. Also, construction 3 of Erdős give rulers of size $2n^2$, double of what we want to prove. None of these 3 constuctions will be useful.

Only the remaining constructions can provide rulers of size near n^2. Singer's construction depends on the evaluation of the field $GF(p^3)$ and the computational cost is quite prohibitive for rulers with a large number of marks.

The two candidate constructions are by I. Ruzsa (construction 4) and Bose and Chowla (construction 6). We will implement both of them in a quick way to find Golomb rulers with large number of marks.

6.3 A note on the computational model

We will use a simple computational model where each elementary operation, addition or multiplication, takes constant time O(1). This is justified as we shall be handing only integers with bounded length. Most of the computations are performed on 32-bit integers and on some occasions 64-bit integers are used that can be handled naturally by compiler technology in $O(1)$ time.

Using this computational model can provide us a basis for making reasonable estimates of the algorithm's actual running time on a computer. This will be supported by the actual running times that will be presented later.

However for the sake of being complete in our discussion, an asymptotic estimate for the number of bit operations the algorithm performs will be presented. These estimates are dominant by the number multiplications and divisions used and asymptotic estimates for these operations will be given.

A multiplication of two b-bit integers by an ordinary method takes time $\Theta(b^2)$. Similarly the operation of division a b-bit integer by a shorter integer or the operations of taking the remainder of a b-bit integer when divided by a shorter one can be performed also in time $\Theta(b^2)$ by simple algorithms [16].

The estimates of bit operations can be used to predict running times (up to a constant) when arbitrary precision arithmetic is used in the implementation of various arbitrary precision library packages (see for example `gmp`, `cln` and `piologie`).

6.4 Common algorithms for both constructions

6.4.1 Modular multiplication of a construction

In both of the constructions that will be implemented, we will be concerned with modular structures for Golomb rulers. A modular structure as we saw in chapter 5, page 50 can be multplied by an integer z relative prime to the modulus of the construction to yield another Golomb ruler.

The following elementary algorithm will be used in both constructions for this purpose. It multiplies the construction $a[1 \dots n]$ by m and reduce modulo z. After the multiplication the generated sequence is not necessarily in sorted order so we have to sort it.

```
MODULAR-MULTIPLY(a[ ], n, z, m)
for i := 1, 2, ..., n do
    b[i] ← (m · a[i]) mod z
SORT(b[ ], n)
return b[ ]
```

The running time of the algorithm is dominated by the sorting procedure. Linear time sorting algorithms exists [4], however the constants involved in linear time sorting algorithms are often large and classical sorting approaches are efficient. A classical comparison based sort like mergesort [16] will serve to provide an upper bound of $O(n \log n)$ for the total running time.

6.4.2 Truncating and unwounding a construction

As the constructions we will use produce Golomb rulers only for prime number of marks it is necessary to truncate these constructions to find Golomb rulers for any number of marks.

Suppose the we are given a modular construction $a[1 \dots n]$ modulo z and wish to extract from this modular construction, Golomb rulers with sizes l,$l+1$,…,n. By lemma 5.11 we can find the maximum gap between pairs of elements that are apart k positions in the sequence, for k in 1,2 and up to $n - l + 1$.

Before we begin we unwound the construction so that

$$a[n+1], a[n+2], \dots, a[2n] = a[1] + z, a[2] + z, \dots, a[n] + z$$

which eliminates the need of taking modulus every time.

```
MODULAR-EXTRACT(a[ ], n, z, l)
for i := 1, 2, ..., n do
    a[i + n] ← a[i] + z
maxofs[1 ... n − l + 1] ← −∞
maxarg[1 ... n − l + 1] ← 0
for k := 1, 2, ..., n − l + 1 do
    for i := 1, 2, ..., n do
        if a[i + k] − a[i] > maxofs[k] then
            maxofs[k] ← a[i + k] − a[i]
            maxarg[k] ← i
return maxofs[ ],maxarg[ ]
```

The procedure MODULAR-EXTRACT takes time $O(n^2 - nl)$ to complete using $\Theta(n^2 - nl)$ additions and no multiplications.

6.5 A fast algorithm for the construction of Ruzsa

As we proved in chapter 5, when p is a prime number and g a primitive element modulo p, then the numbers

$$pi + (p-1)g^i \mod p(p-1)$$

for $1 \leqslant i \leqslant p-1$ form a Golomb ruler.

The evaluation of the sequence is straightforward in time $O(n)$ but depends on finding a primitive element of the field Z_p^*. The discussion for the procedure of finding primitive elements PRIMITIVE-ELEMENT will be postponed for a bit later.

After the first primitive element is found then it is straightforward to generate the construction. Lindstrom [43] has proved that by varying the primitive element is equivalent to shifting and multiplying the construction by some k relative prime to $p-1$. So, to find all different constructions for a single prime number p it is not necessary to vary the primitive element but just use the MODULAR-MULTIPLY procedure for all suitable mutlipliers.

The following algorithm will exhaust all possible constructions for a given prime p and truncate to Golomb rulers of sizes downto l.

```
    Ruzsa-Extract(l,p)
1   g ← Primitive-Element(p)
2   c ← p
3   d ← (p − 1)g
4   for i := 1, 2, ..., p − 1 do
5       a[i] ← (c + d)  mod p(p − 1)
6       c ← c + p
7       d ← (d · g)  mod p(p − 1)
8   maxofs[1, ..., n − l] ← 0
9   maxarg[1, ..., n − l] ← 0
10  maxmult[1, ..., n − l] ← 0
11  for m := 1, 2, ... n do
12      if gcd(m, p − 1) = 1 then
13          b[ ] ← Modular-Multiply(a[ ], p − 1, p(p − 1), m)
14          ofs[ ], arg[ ] ← Modular-Extract(a[ ], p − 1, p(p − 1), l)
15          for i := 1, 2, ... n − l
16              if ofs[i] > maxofs[i] then
17                  maxofs[i] ← ofs[i]
18                  maxarg[i] ← arg[i]
19                  maxmult[i] ← m
20  for n := l, l + 1, ..., p − 1
21      size[n] ← p(p − 1) − maxofs[p − n]
22  return g, size[ ],maxarg[ ],maxmult[ ]
```

Actual Golomb rulers are not produced by this algorithm but the primitive element and the multiplier m used are output at the end of the program which allows the reconstruction of the Golomb ruler instantly at a later time.

The best sizes of Golomb rulers that were found are left in array size[ ] after the end of the algorithm.

The running time is dominated by the loop of lines 12-19. Asymptotically the algorithm takes time

$$T_1(l,p) = O(\ \phi(p-1)[p\log p + p(p-l)]\ ) \tag{6.1}$$

where the first part in the brackets corresponds to Modular-Multiply procedure and the second to Modular-Extract. It depends on the choice of l whether the order of magnitute of the term in the brackets will be $p \log p$ or p^2. We shall see later that for our computations $p \log p$ will be dominant.

The Euler's $\phi(n)$ function never assumes values greater that $n-1$ and it's order of magnitude is about n. The time of the algorithm thus can be bounded by

$$T_1(l,p) = O(\ p^2 \log p + p^2(p-l)\ ) \tag{6.2}$$

Actually in Hardy and Wright[30] it is proved that $\phi(n)$ is about $\frac{6}{\pi^2}n \approx 0.61n$.

Since p is prime in our algorithm we can prove a slightly better bound, that $\phi(p-1)$ will never assume a value greater that $p/2$.

Theorem 6.2. *If p is an odd prime then $\phi(p-1) \leqslant \frac{p}{2}$.*

Proof. Since p is prime $p-1$ is divisible by 2 which implies that at least half the numbers less than $p-1$ have $\phi(n) \geqslant 2$. □

6.5.1 Finding a primitive element

To produce a corrent Sidon sequence using the construction we described, we have to find a primitive element g of the group Z_p^*.

Testing if an element is primitive

Recall that a primitive element g of the multiplicative group Z_p^* is one that $g^i \equiv 1 \mod p$ does not hold for $i < p-1$. To test the primitiveness of an element g all powers of g modulo p must be verified to be greater than 1.

The recursive computation $g^p = g \cdot g^{p-1}$ is used, which takes time $O(1)$ to compute each power for a total cost of $O(p)$.

By corollary 5.5 we halve the time of the algorithm by computing powers up to $\lfloor \frac{p-1}{2} \rfloor$.

```
TEST-PRIMITIVENESS(g, p)
r ← 1
for i := 1, 2, ..., ⌊(p−1)/2⌋ do
    r ← (g · r) mod p
    if r = 1 then
        then return false
return true
```

The algorithm take $O(p)$ time in the worst case, when we have a primitive element and the loop completes all iterations. In this case, $2\lfloor\frac{p-1}{2}\rfloor \leqslant p-1$ multiplications/divisions are executed.

Finding a primitive element

To find a primitive elements of Z_p^* we should iterate g through all values of the multiplicative group. For $p > 2$, we can omit the test for 1 which is never a primitive element and $p-1$ which is not primitive by the identity

$$(p-1)^2 \equiv p^2 - 2p + 1 \equiv 1 \pmod{p}$$

The following algorithm find a primitive element of Z_p^*.

```
PRIMITIVE-ELEMENT(p)
for g := 2, ..., p − 2 do
    if TEST-PRIMITIVENESS(g, p)
        then return g
```

We have proved in lemma 5.7 that there exist $\phi(p-1)$ primitive elements in a prime field. Computationally these elements are uniformly spaced inside Z_p, thus by counting starting from 2 we can assure that we will find a primitive element soon. By the evaluation of the algorithm, primitive elements less than 35 were found for all fields up to Z_{100000} so the PRIMITIVE-ELEMENT function is guaranteed to complete quick.

6.6 Bose-Chowla construction

The next construction we will implement will be by Bose and Chowla described in [11].

Recall that we will have to find a primitive element θ of the field $GF(q^2)$. Then the elements of the sequence are integers a with $1 \leqslant a < q^2$ having the property that

$$\theta^a - \theta \in GF(q).$$

We will implement the construction only for the case where $q = p$ is a prime number and not a prime power. Evaluating the Galois field for the case of prime powers involves computations with polynomials of large degree and is prohibitive for our purposes.

In the simple second order extension field $GF(q^2)$ of $GF(q)$ we are dealing with polynomials of degree at most 1 modulo q. The computations of such polynomials can be performed in constant time.

The following algorithm will evaluate the construction for a prime number p.

Bose-Chowla(θ,p)

$\zeta \leftarrow \theta$
for $n := 1, 2, \ldots, q^2$ **do**
  if $\zeta - \theta \in GF(q)$ **then**
    $a[i] \leftarrow n$
    $i \leftarrow i + 1$
  $\zeta \leftarrow \zeta \cdot \theta$
return $a[\,]$

An interesting property of the particular construction is that the marks of the ruler are produced in sorted order, something that does not hold for the previous construction we discussed.

The running time of the Bose-Chowla algorithm is $O(q^2)$ if the computations in the field $GF(p^2)$ can be done in constant time (which is the case when p is prime).

Lindstrom [43] has proved that varying the primitive element is equivalent to multiplying and shifting the construction (much like the case of the previous construction).

To find all possible Golomb rulers from this construction one has to exhaust all possibilities of the multiplier m in MODULAR-MULTIPLY procedure. As we proved in property 4 the possible multipliers are the ones being relative prime to the modulus of the construction.

In this case the construction has modulus $p^2 - 1$ and the total number of possible multipliers is $\phi(p^2 - 1)$ which is of order p^2.

Using the same technique as previously we can extract Golomb rulers of any size and not just prime numbers. The following algorithm will find rulers of sizes $l, l+1, \ldots, p$ exhausting all possibilities provided by Bose-Chowla construction.

For the function GF2-PRIMITIVE-ELEMENT, the implementation will be provided by LiDIA, a library for computational number theory.

```
BOSE-EXTRACT(l,p)
θ ← GF2-PRIMITIVE-ELEMENT(p)
a[ ] ← BOSE-CHOWLA(θ, p)
maxofs[1, ..., n − l + 1] ← 0
maxarg[1, ..., n − l + 1] ← 0
maxmult[1, ..., n − l + 1] ← 0
for m := 1, 2, ... p^2 − 1 do
    if gcd(m, p^2 − 1) = 1 then
        b[ ] ← MODULAR-MULTIPLY(a[ ], p, p^2 − 1, m)
        ofs[ ], arg[ ] ← MODULAR-EXTRACT(a[ ], p, p^2 − 1, l)
        for i := 1, 2, ... n − l + 1
            if ofs[i] > maxofs[i] then
                maxofs[i] ← ofs[i]
                maxarg[i] ← arg[i]
                maxmult[i] ← m
for n := l, l + 1, ... p
    size[n] ← p^2 − 1 − maxofs[p − n + 1]
return size[ ],maxarg[ ],maxmult[ ],θ
```

The algorithm follows in the same framework of RUZSA-EXTRACT. It returns the actual size of Golomb rulers found in array size[ ] and all the necessary data to reconstruct the ruler later in the arrays maxarg[ ],maxmult[ ] together with the primitive element θ used.

The running time is now

$$T_2(l,p) = O(\ \phi(p^2-1))[p\log p + p(p-l)]\ ) \tag{6.3}$$

Again, by the properties of the Euler's ϕ function

$$T_2(l,p) = O(\ p^3\log p + p^2(p-l)\ ) \tag{6.4}$$

which is worse by a factor p compared to the previous construction.

6.7 Implementation

Both algorithms were implemented in ANSI C++. All the development was done at the Linux operating system with the GNU C++ compiler. The source code of both constructions is listed in the appendices.

Since we are dealing with large integers near 2^{32} an important topic that should be addressed is the possibility of overflowing machine size integers.

Actually, the choice of 65000 as the maximum element of Golomb rulers co-incides with the limit of 32-bit integers. For $p > 65535$, p^2 overflows 32-bit integers and would have to resort to arbitrary precision arithmetic libraries. Then the efficiency of the algorithms would suffer severely by the speed of these libraries and one should expect the running time to be multiplied by at least 10, since every elementary operation would be replaced by library calls.

There are some places in the algorithms that integers of size greater than p^2 have to be computed. Such dangerous places are:

1. Modular-Multiply algorithm line 2, when $m \cdot a[i] \approx p^3$ if $a[i]$ is about p^2 and the multiplier m is about p.

2. Ruzsa-Extract algorithm line 7, when d is about p^2 and g is about p.

3. Modular-Extract algorithm line 2, where the unwound sequence might be of size $2p^2$.

Fortunately, for these limited cases, we use the GNU g++ compiler's internal support for fast 64-bit data types (data type `long long`). We use such 64-bit integers whenever there is a danger of overflowing 32-bit data types. To overflow 64-bit integers with p^3, p must be at least

$$p^3 > 2^{64} \Longrightarrow p > 2642245$$

quite above the valus of p we will need to prove 6.1.

The running time of the algorithms is only slightly increased by using `long long` carefully and only in places where necessary.

As for the sorting of the sequence, the SORT procedure is an implementation of standard textbook QuickSort which reverts to insertion sort for less than 15 elements.

6.8 Exhaustive search for Golomb rulers

To prove theorem 6.1 we have to extract Golomb rulers using the two consctructions we described for all $n \leqslant 65000$.

6.8.1 Computing the total running time

Since the algorithms apply only to prime numbers, for each prime less than 65000 the algorithms should be run and allowed to extract Golomb rulers down to the previous prime. This will fill the gap between prime numbers and provide Golomb rulers of all sizes.

For example the first few calls to RUZSA-EXTRACT will be

$$\begin{array}{l} \text{Ruzsa-Extract}(2,3) \\ \text{Ruzsa-Extract}(3,5) \\ \text{Ruzsa-Extract}(5,7) \\ \text{Ruzsa-Extract}(7,11) \\ \ldots \\ \text{Ruzsa-Extract}(p_{n-1},p_n) \end{array}$$

where p_n be the n-th prime. Then the total running time of the Ruzsa extract algorithm to extract all Golomb rulers of sizes up to x will be by equation 6.1

$$T_3(x) = O\left(\sum_{n=1}^{p_n<x} \phi(p_n - 1)[p_n \log p_n + p_n(p_n - p_{n-1})] \right)$$

The difference between two consecutive primes $p_n - p_{n-1}$, although completely irregular, averagely is so slowly increasing that computationally it is practically constant. So, the second term in the parenthesis is about p_n and neglible to the first term.

$$T_3(x) = O\left(\sum_{n=1}^{p_n<x} \phi(p_n - 1)p_n \log p_n\right)$$

Similarly for the Bose construction the total running time will be

$$T_4(x) = O\left(\sum_{n=1}^{p_n<x} \phi(p_n^2 - 1)p_n \log p_n\right)$$

Since the sums cannot be evaluated due to the inability to find the exact distribution of primes, an initial test was run of both algorithms for $x < 3000$. Running times are presented for both algorithms in figure 6.8.1.

The running times appear irregular because of the irregularities of the Euler's ϕ functions. However, in the long term the running times follow an polynomial increase which is of the order n^2 for Ruzsa and n^3 for Bose. The running time of Bose appear a bit more regular.

As expected Ruzsa's construction is much faster than Bose.

$T_3(x)$ and $T_4(x)$, the cumulative sum of the running times, fitted with a non-linear curve fitter to the following values

$$\begin{array}{rl} \textsc{Ruzsa-Extract} & T_3(x) \approx 5.56 \cdot 10^{-9}x^3 \\ \textsc{Bose-Extract} & T_4(x) \approx 2.40 \cdot 10^{-9}x^4 \end{array}$$

For $x = 65000$, Ruzsa-Extract would take about 17 days on the reference cpu machine and Bose-Extract about 1358 years so the choice of Ruzsa's construction as our main tool is inevitable.

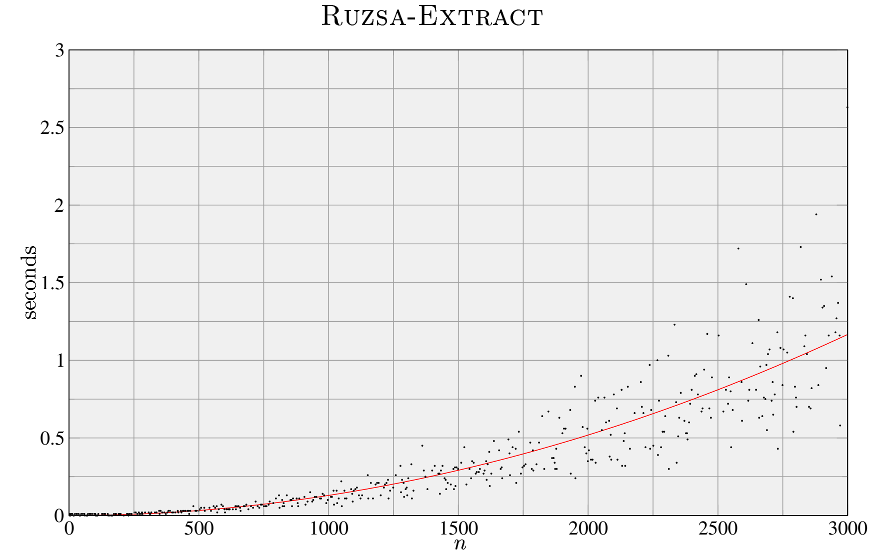
The algorithm was run on a computer network of 10 machines. Since the computations can be split in small units, one for each prime number, the algorithm parallelizes easily.

A distributed client was implemented which reads a stub (work unit) from a central server. It then computes the stub and sends the results back to the server. The communication was done using TCP sockets.

Both client and server were implemented in TCL, which facilitates the quick development of such application. The workhorse routine was implemented in C++ using GNU C++ compiler.

6.9 Summary

In this chapter we introduced the goal of chapters 6 and 7: the computational verification that Golomb rulers of size less than n^2 exist for all $n \leqslant 65000$. We have chosen the algorithm which uses Ruzsa's construction, having the other algorithm as a backup. In the next chapter, we will see that this backup, the algorithm for Bose's construction, will actually be desperately needed to complete our goal.



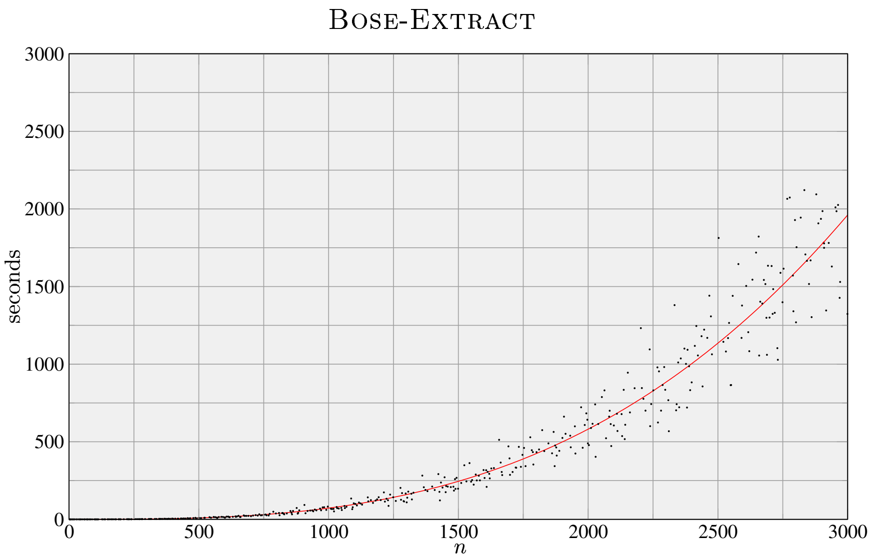


Figure 6.1: Running times of both algorithms for the test run

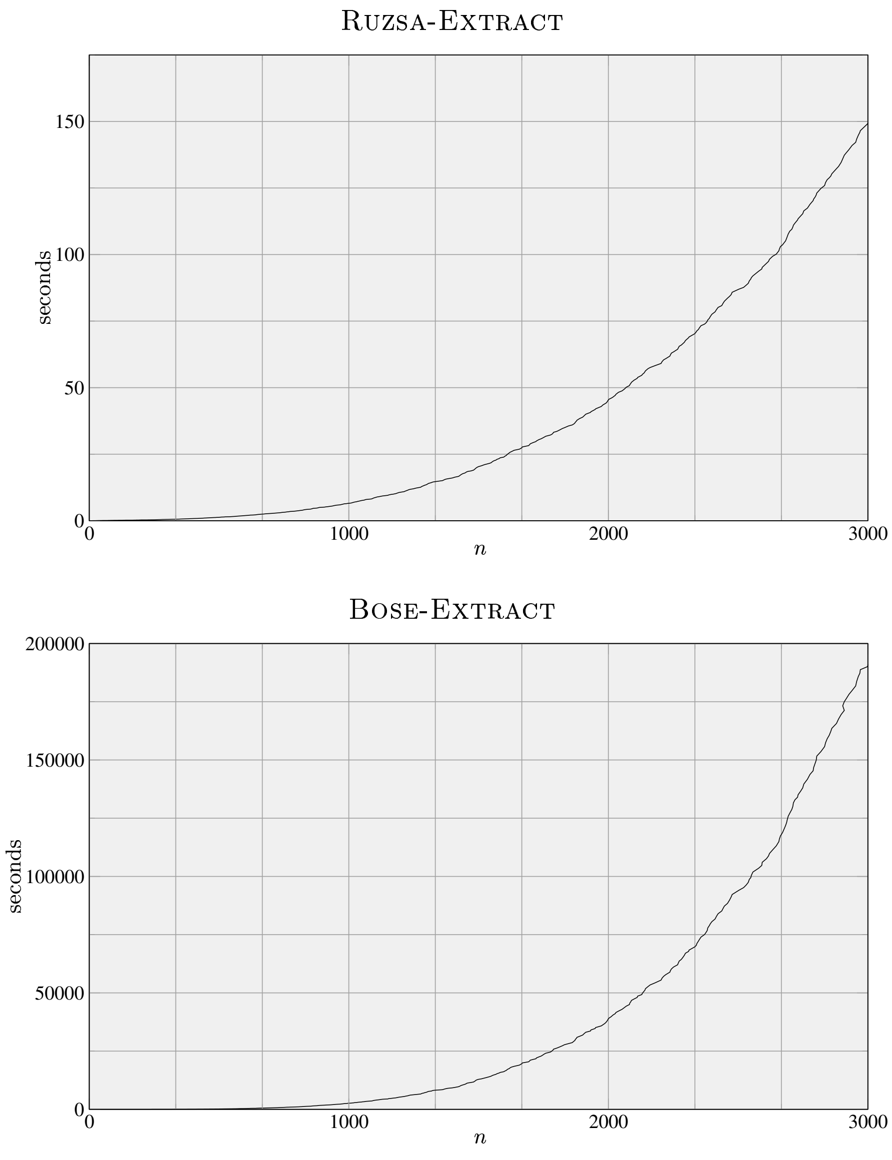


Figure 6.2: Cumulative running times of both algorithms

Chapter 7

Results and proof of main theorem

Computations were carried out first by RUZSA-EXTRACT for up to 65000 marks. Most of the rulers produced had the desired property of having length less than n^2. However is a small number of cases the best rulers found had size more than n^2. In these cases, the much slower algorithm BOSE-EXTRACT was used to correct the situation and find suitable Golomb rulers.

7.1 Rulers found by Ruzsa's construction

First we will consider the much faster construction of Ruzsa and the algorithm RUZSA-EXTRACT we developed in the previous chapter.

Computation of Ruzsa's construction for all rulers up to 65000 number of marks has been carried out in a distributed network consisting of 10 personal computers running linux. The computational power of these workstations varied but most of them had CPU clocks about 1.5GHz.

About 2.1 million cpu seconds or 24 cpu days were used for these computations, close to the estimate of the previous chapter.

7.1.1 Prime number of marks

We will first present the results for prime number of marks in which the algorithm had no problem to find near-optimal ruler of suitable sizes. In

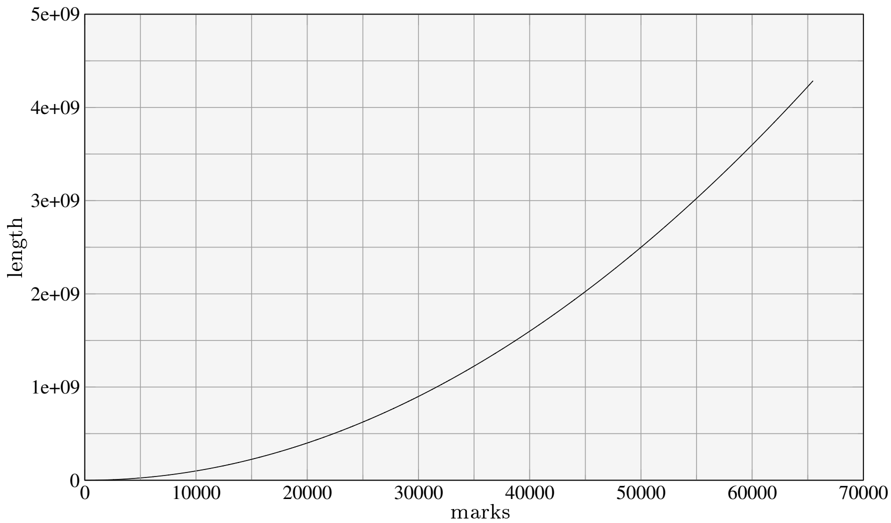


Figure 7.1: Near optimal rulers for prime number of marks

figure 7.1 the length of the rulers produced from the construction for prime number of marks is shown. The lengths of the rulers follow almost exactly the curve n^2 from below.

To have a clearer look at the results in figure 7.2 the difference between n^2 and the length of the best Golomb ruler found is shown.

As it can been seen from the graph Golomb rulers of sizes less than n^2 were found for all the cases of prime numbers. Moreover the difference between n^2 increases almost linearly with the number of marks.

As revealed by a non-linear curve fitter, the data are more precisely described as exponential. The fitter found that the following is a good estimate of the size of Golomb rulers produced by the I. Ruzsa construction

$$n^2 - 6.38n^{1.1}.$$

7.1.2 Non-prime number of marks

For the proof of the main theorem, Golomb rulers of sizes less than n^2 for any number of marks and not just prime numbers must be found.

To fill the gap between the primes RUZSA-EXTRACT algorithm we developed in chapter 6 uses the construction for the next largest prime number

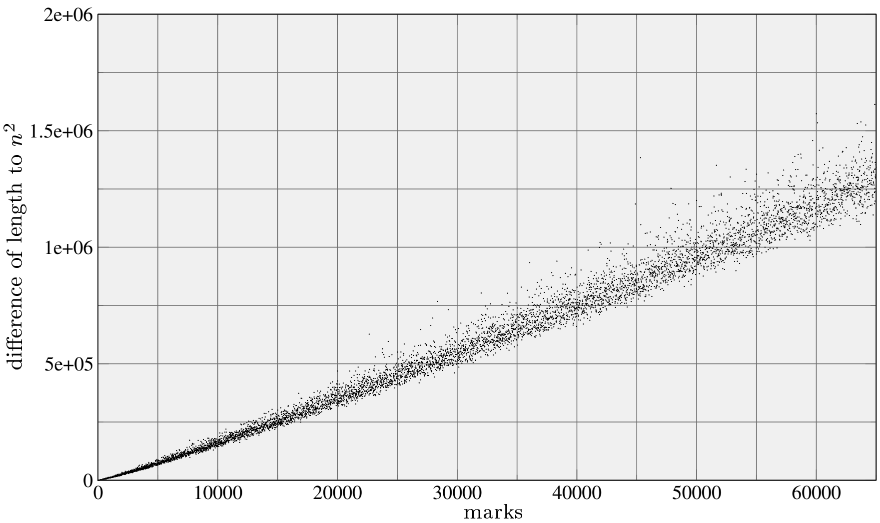


Figure 7.2: Difference of n^2 and ruler size for prime number of marks

and removes the necessary number of marks.

The difference of the length and n^2 is shown in figures 7.3, 7.4, 7.5 and 7.6.

The algorithm produced good Golomb rulers for the 99.999% of the cases. However, there is a small number of negative results, in which the length of the best Golomb ruler produced was more than the goal n^2. These cases have to be resolved for the proof of the main theorem. The results that were negative are shown in table 7.1.

The problem occurs precisely at the points where the difference between two consecutive prime numbers is large. In this case the algorithm has to use a construction for the next largest prime and remove a large number of elements to fill the gap.

The elements of a dense Golomb ruler are approximately linearly distributed over the full length of the ruler [23]. That means that starting with a ruler with n marks and size about n^2 the i-th element is about at position in. Extracting from this ruler another one with m marks, the latter will have size about mn. Thus, as m decreases, mn becomes increasingly larger than the goal m^2.

This is shown in figure 7.7 where the construction of Ruzsa for 277 marks

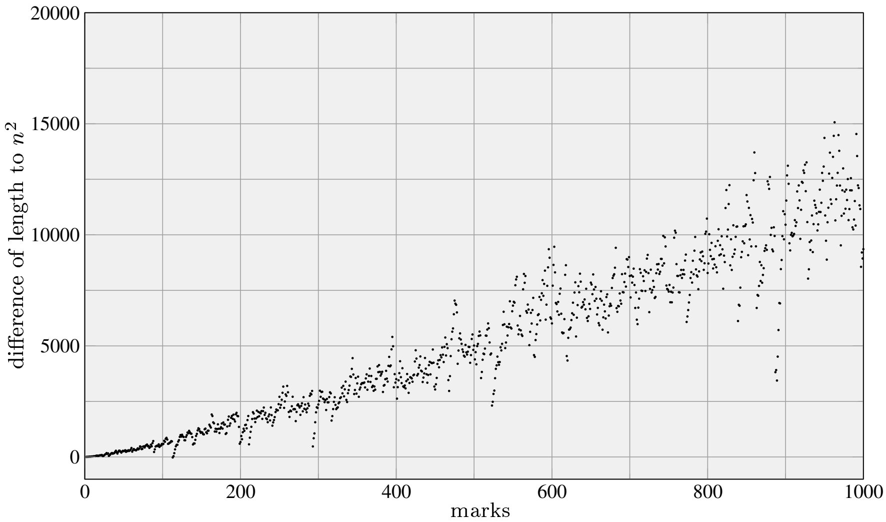


Figure 7.3: Near optimal rulers for any number of marks (1-1000)

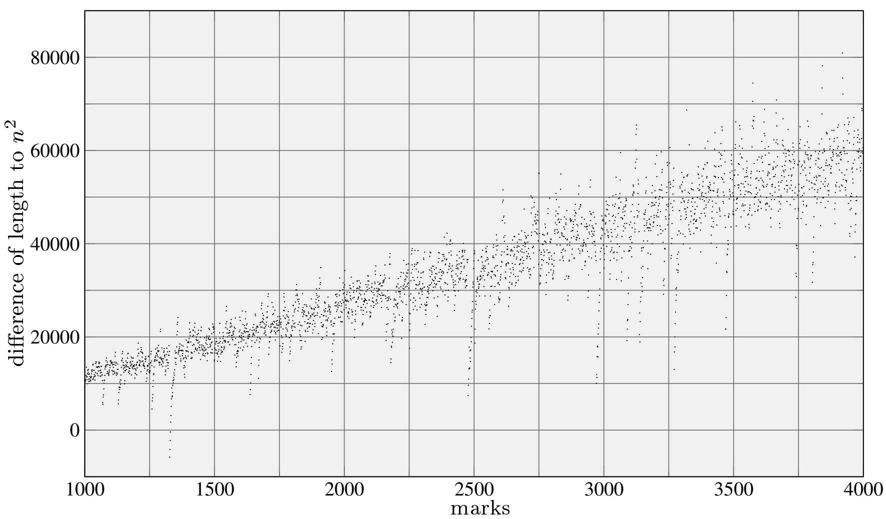


Figure 7.4: Near optimal rulers for any number of marks (1000-4000)

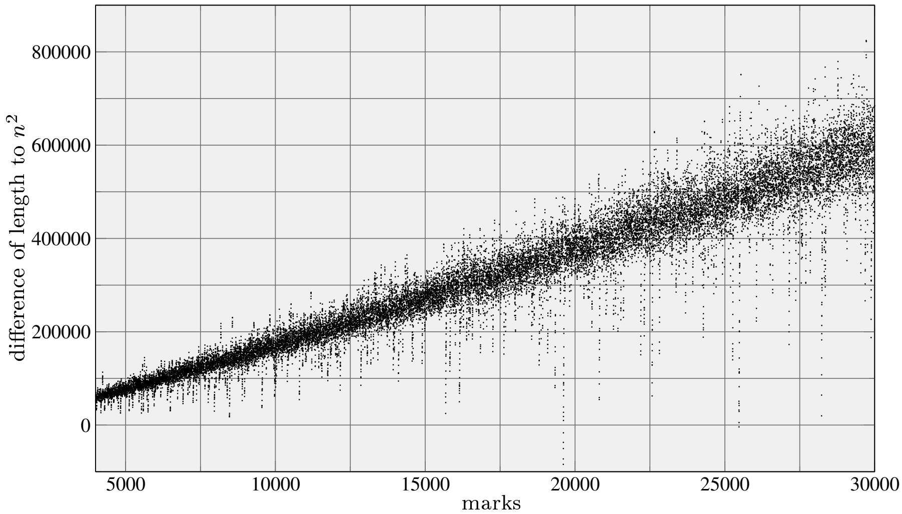


Figure 7.5: Near optimal rulers for any number of marks (4000-30000)

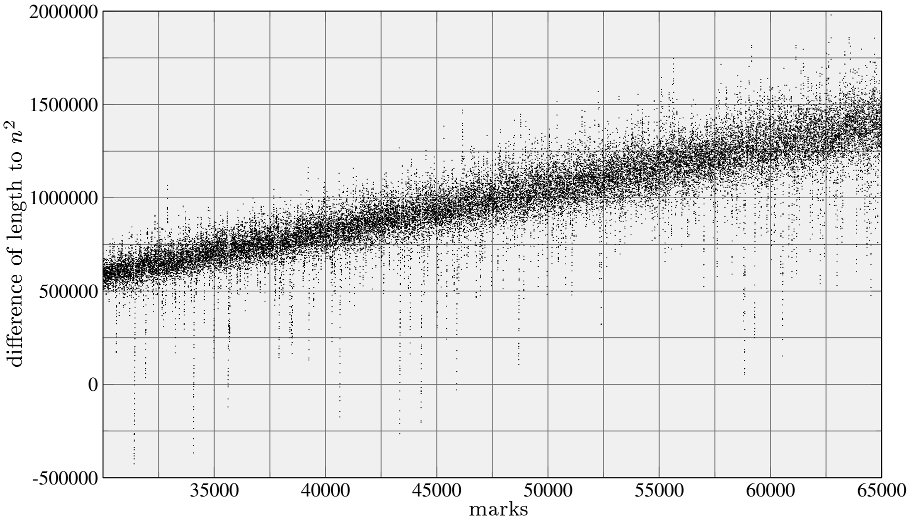


Figure 7.6: Near optimal rulers for any number of marks (30000-65000)

Table 7.1: Negative results

n	l	$n^2 - l$
113	12790	-21
1327	1766741	-5812
1328	1767716	-4132
1329	1768457	-2216
1330	1769309	-409
19609	384597182	-84301
19610	384624625	-72525
19611	384641715	-50394
19612	384667800	-37256
19613	384686390	-16621
25474	648928604	-3928
31397	986197070	-425461
31398	986204645	-370241
31399	986296027	-398826
31400	986343472	-383472
31401	986373159	-350358
31402	986424378	-338774
31403	986452244	-303835
31404	986479109	-267893
31405	986542983	-268958
31406	986635350	-298514
31407	986663424	-263775
31408	986677530	-215066
31409	986716601	-191320
31410	986785033	-196933
31411	986820402	-169481
31412	986869510	-155766
31413	986910763	-134194
31414	986943786	-104390
31415	986979155	-76930
31416	987012077	-47021
31417	987031261	-3372
34061	1160519170	-367449
34062	1160547296	-327452
34063	1160578879	-290910
34064	1160592347	-236251
34065	1160623930	-199705
34066	1160670148	-177792
34067	1160695738	-135249
34068	1160696903	-68279
34069	1160722604	-25843
34070	1160829254	-64354
34071	1160953336	-120295
34072	1160954501	-53317
34073	1160980202	-10873
34074	1161037977	-501
35617	1268691836	-121147
35618	1268720119	-78195
35619	1268726764	-13603
35623	1269001983	-3854
40639	1651703022	-174701
40640	1651754919	-145319
40641	1651757563	-66682
40642	1651859670	-87506
40643	1651867536	-14087
43331	1877839745	-264184
43332	1877870808	-208584
43333	1877934842	-185953
43334	1877977590	-142034
43335	1878001275	-79050
43336	1878044384	-35488
44293	1962072520	-202671
44294	1962154288	-195852
44295	1962171771	-124746
44296	1962230373	-94757
44297	1962259182	-34973
44298	1962330046	-17242
44299	1962450197	-48796
44300	1962526788	-36788
44301	1962615591	-36990
45893	2106197458	-30009

is used to extract rulers down to 1 mark. The difference of the length and n^2 quickly becomes negative. The negative results occur precisely at the points where the difference between two primes assumes a large value.

All negative results can be grouped in the areas shown in table 7.2.

Table 7.2: Negative results and prime gaps

negative results	prime gap	gap length
113	113 – 127	14
1327 – 1330	1327 – 1361	34
19609 – 19613	19609 – 19661	52
25474	25471 – 25523	52
31397 – 31417	31397 – 31469	72
34061 – 34074	34061 – 34123	62
35617 – 35623	35617 – 35671	54
40639 – 40643	40639 – 40693	54
43331 – 43336	43331 – 43391	60
44293 – 44301	44293 – 44351	58
45893	45893 – 45943	50

The average prime gap between 1 and 1500 is 6 and between 1 and 65000 is 10. When this gap becomes quite larger than this, the algorithm is not able to produce good Golomb rulers.

The most discouraging case is between 31397 and 31417 where the prime gap is 72, quite above the average, and Golomb rulers found become very bad. This situation is displayed more precisely in figure 7.8

7.2 Rulers found by Bose-Chowla construction

7.2.1 Finishing the proof of the main theorem

To finish the proof of our theoren, for the points where Ruzsa's construction failed, we used Bose-Extract algorithm for precisely the cases where size of the rulers produced by the first algorithm was negative.

The algorithm was allowed to run until it found a Golomb ruler of suitable size and then stopped. Otherwise running through all possible constructions of Bose-Chowla even for a single number of marks p would take many years of CPU time by the estimate of the previous chapter and the

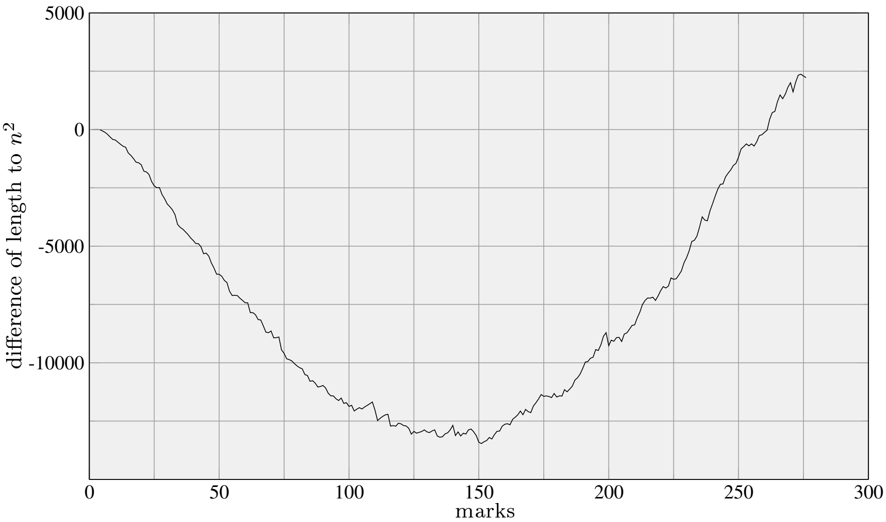


Figure 7.7: Extracted rulers from a 277 marks construction

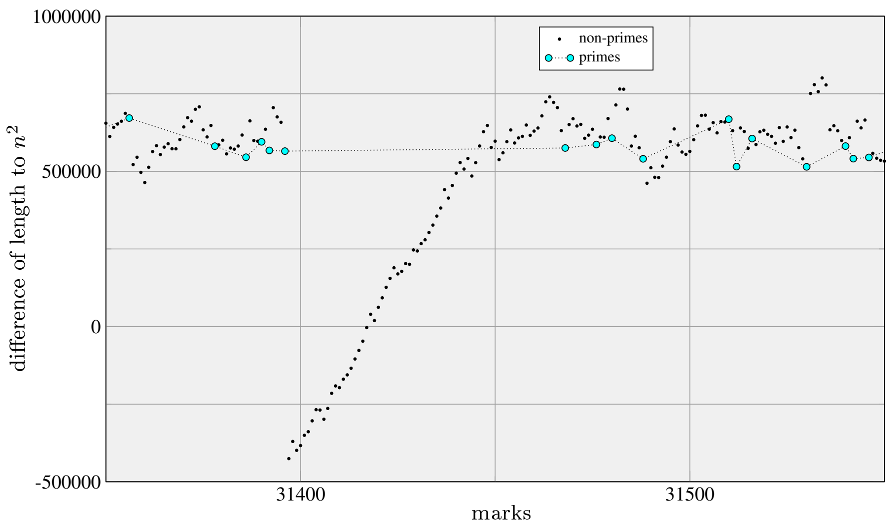


Figure 7.8: The situation between 31397 and 31417

observations of the speed that the multipliers were exhausted.

The optimal Golomb rulers found using this method are presented in the following table, along with the irreducible polynomials and primitive elements used for the computation of the Galois fields.

$$GF(113^2) = \langle 69x + 2 \rangle / x^2 + 13x + 41 \mod 113$$

n	length	m	$n^2 - l$
113	11647	115	1122

$$GF(1362^2) = \langle 1220x + 162 \rangle / x^2 + 276x + 48 \mod 1362$$

n	length	m	$n^2 - l$
1327	1751426	98711	9503
1328	1753632	98711	9952
1329	1756863	29129	9378
1330	1757303	29129	11597

$$GF(19661^2) = \langle 12958x + 10688 \rangle / x^2 + 6013x + 5403 \mod 19661$$

n	length	m	$n^2 - l$
19609	384199114	3415357	313767
19610	384209834	3415357	342266
19611	384255276	3415357	336045
19612	384280268	12122089	350276
19613	384315330	3415357	354439

$$GF(25523^2) = \langle 11709x + 22179 \rangle / x^2 + 9158x + 7230 \mod 25523$$

n	length	m	$n^2 - l$
25474	648887797	20483	36879

$$GF(31397^2) = \langle 21016x + 14783 \rangle / x^2 + 25136x + 23613 \mod 31397$$

n	length	m	$n^2 - l$
31397	985182074	7391	589535

$GF(31469^2) = \langle 1554x + 29508\rangle / x^2 + 29760x + 9050 \mod 31469$

n	length	m	$n^2 - l$
31398	985733211	8556871	101193
31399	985790592	8556871	106609
31400	985835555	8556871	124445
31401	985879154	8556871	143647
31402	985951670	8556871	133934
31403	986001830	8556871	146579
31404	986033653	8556871	177563
31405	986087575	8556871	186450
31406	986108679	8556871	228157
31407	986181195	8556871	218454
31408	986231379	8556871	231085
31409	986279127	8556871	246154
31410	986287982	8556871	300118
31411	986309474	8556871	341447
31412	986366671	8556871	347073
31413	986411811	8556871	364758
31414	986480862	8556871	358534
31415	986503109	8556871	399116
31416	986575625	8556871	389431
31417	986633006	8556871	394883

$GF(34123^2) = \langle 19239x + 16497\rangle / x^2 + 13878x + 14722 \mod 34123$

n	length	m	$n^2 - l$
34061	1160133938	1045519	17783
34062	1160184349	2087299	35495
34063	1160286494	450227	1475
34064	1160328690	444775	27406
34065	1160379857	231775	44368
34066	1160412101	231775	80255
34067	1160520937	163765	39552
34068	1160615428	76973	13196
34069	1160673088	33721	23673
34070	1160702245	33721	62655
34071	1160777943	33721	55098
34072	1160822990	33721	78194
34073	1160852147	33721	117182
34074	1161029988	4751	7488

$$GF(35671^2) = \langle 19447x + 33743\rangle / x^2 + 25262x + 26129 \mod 35671$$

n	length	m	$n^2 - l$
35617	1268463057	88651	107632
35618	1268475498	88651	166426
35619	1268703314	71489	9847
35620	1268760714	71489	23686
35621	1268836365	68011	19276
35622	1268903019	27589	23865
35623	1268976373	4367	21756

$$GF(40693^2) = \langle 5595x + 16764\rangle / x^2 + 21946x + 2344 \mod 40693$$

n	length	m	$n^2 - l$
40639	1651339857	16115	188464
40640	1651398771	16115	210829
40641	1651504446	16115	186435
40642	1651563360	16115	208804
40643	1651667503	16115	185946

$$GF(43391^2) = \langle 39179x + 25402\rangle / x^2 + 4218x + 42993 \mod 43391$$

n	length	m	$n^2 - l$
43331	1877499646	84529	75915
43332	1877540552	84529	121672
43333	1877627142	84529	121747
43334	1877771317	84529	64239
43335	1877812223	84529	110002
43336	1877898813	84529	110083

$$GF(44351^2) = \langle 3722x + 32899\rangle / x^2 + 1508x + 4745 \mod 44351$$

n	length	m	$n^2 - l$
44293	1961850181	380159	19668
44294	1961890065	380159	68371
44295	1961926018	380159	121007
44296	1961965902	380159	169714
44297	1962017702	380159	206507
44298	1962077915	380159	234889
44299	1962257057	380159	144344
44300	1962280901	380159	209099
44301	1962341114	380159	237487

$$GF(45943^2) = \langle 2981x + 13885\rangle / x^2 + 34036x + 4658 \mod 45943$$

n	length	m	$n^2 - l$
45893	2105954557	97321	212892

7.2.2 Complete computations of Bose's construction

Apart from the proof of the main theorem, Bose-Chowla construction was completely evaluated iterating through all possible constructions and multipliers for a number of marks up to 3000 marks.

In this range Bose's construction has produced in most of the cases better Golomb rulers that Ruzsa's construction, of course with much more computational time. Golomb rulers of sizes less than n^2 were found in this range for any number of marks.

The results for up to 3000 marks are presented in figure 7.9. As usual the difference between the length and n^2 is presented.

The function $x\sqrt{x}$ is also plotted which was believed by some authors to be about the difference between n^2 and the length of an optimal Golomb ruler. However, clearly $x\sqrt{x}$ diverges from the results so the conjecture that the length of an optimal ruler is about $x^2 - x\sqrt{x}$ does not seem to hold.

7.3 Summary

The algorithms RUZSA-EXTRACT and BOSE-EXTRACT together were used to provide the proof of the main theorem of chapter 6.

All the results together with the appropriate code and tables of computed Golomb rulers, can be found at the web page devoted to this diploma thesis

http://www.softnet.tuc.gr/~apdim/diploma.

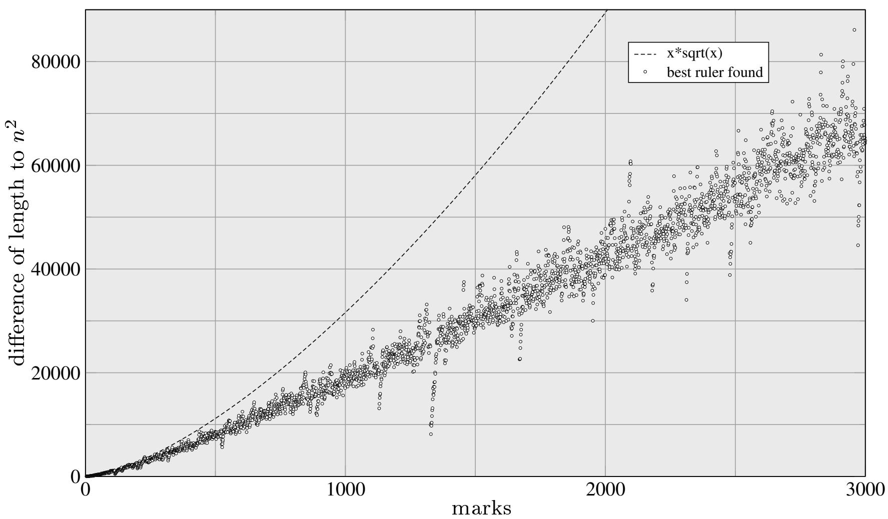


Figure 7.9: Rulers found by Bose-Chowla for up to 3000 marks

Chapter 8

Conclusion

In this thesis, two related problem from number theory, Golomb rulers and Sidon sets were investigated. Although the two problems are closely related, they were studied almost independently, with some authors ignoring and reproving older results.

To clarify this situation, we have concentrated on the equivalence between the two problems and, in chapter 4, we proved theorems that allow bounds from one problem to be restated to the other.

The results were applied to a known upper bound on the size of Sidon sets by Lindström to yield a better lower bound for the length of optimal Golomb rulers: an optimal Golomb ruler must have size at least

$$n^2 - 2n\sqrt{n} + \sqrt{n} - 2$$

In the second part of the thesis, we investigated constructions from number theory that give modular Golomb rulers and Sidon sets. All the known constructions were reviewed and we investigated their application to Golomb rulers. Of more interest are the three constructions by Singer, Bose-Chowla and Ruzsa. All of these constructions can be applied to produce Golomb rulers of size near n^2, whose length is close to the optimal one (and in some cases optimal).

A conjecture from Sidon sets, that their size is at most $n^{1/2}$ is an old conjecture, circulated by Erdős in the early 40's. Restating this to Golomb rulers (using the results of chapter 4), the conjecture reads that the size of an optimal Golomb rulers is less that n^2.

In chapter 6, we have developed algorithms that allow the quick evaluation of both Ruzsa's construction and Bose's construction to yield near-optimal Golomb rulers. These algorithms, BOSE-EXTRACT and RUZSA-EXTRACT have been used to computationally prove the main theorem of chapter 6 and this thesis, that the conjecture of Erdős is true up to a bound on the number of marks of the ruler.

Both algorithms were used, as Ruzsa's construction failed to produce suitable Golom rulers in some cases where there is a large gap between prime numbers.

Using about 21 CPU days on a distributed run of both algorithms, we proved that rulers of size less than n^2 exist for all n less than 65000. Equivalently, in Sidon set terms

$$F_2(n) < n^{1/2} \quad \text{for all } n < 65000^2$$

So, the conjecture of Erdős is true up to $4.225 \cdot 10^9$. It is still not known if this holds for all n. However, the search we have done can be extended (with additional computational power) to larger bounds. Indeed, it appears that Erdős conjecture is true, however the proof of such a statement is beyond the reach of today's mathematical knowledge. □

Bibliography

[1] M. Ajtai, J. Kolmós, and E. Szemerédi, *A dense infinite Sidon sequence*, European Journal of Combinatorics **2** (1981), 1–11.

[2] N. Alon and M. N. Kolountzakis, *On a problem of Erdős and Turàn and some related results*, Journal of number theory **55** (1995), 82–93.

[3] Andersson, Hagerup, Nilsson, and Raman, *Sorting in linear time?*, STOC: ACM Symposium on Theory of Computing (STOC), 1995.

[4] A. Andersson, T. Hagerup, S. Nilsson, and R. Raman, *Sorting in linear time?*, Journal of Computer and System Sciences **57** (1998), 74–93.

[5] M. D. Atkinson, N. Santoro, and J. Urrutia, *Integer sets with distinct sums and differences and carrier frequence assignments for nonlinear repeaters*, IEEE Transactions on Communications **34** (1986), 614–617.

[6] W.C. Babock, *Intermodulation interference in radio systems*, Bell Systems Technical Journal (1953), 63–73.

[7] G. S. Bloom and S. W. Golomb, *Numbered complete graphs, unusual rulers, and assorted applications*, Lecture Notes in Mathematics, Springer-Verlag New York, 1976, pp. 53–65.

[8] ———, *Applications of numbered undirected graphs*, Proceedings of IEEE **65** (1977), 562–571.

[9] E. J. Blum, F. Biraud, and J. C. Ribes, *On optimal synthetic linear arrays with applications to radioastronomy*, IEEE Transactions on Antennas and Propagation **22** (1974), 108–109.

[10] E. J. Blum, J. C. Ribes, and F. Biraud, *Some new possibilitied of optimal synthetic linear arrays for radioastronomy*, Astronomy and Astrophysics **41** (1975), 409–411.

[11] R.C. Bose, *An affine analogue of Singer's theorem*, Journal of the Indian Mathematical Society **6** (1942), 1–15.

[12] R.C. Bose and S. Chowla, *Theorems in the additive theory of numbers*, Commentarii Mathematici Helvetici **37** (1962-63), 141–147.

[13] Zhi Chen, *Further results on difference triangle sets*, IEEE Transactions on Information Theory **40** (1994), 1268–1270.

[14] S. Chowla, *Solution of a problem of Erdős in and Turàn in additive number theory*, Proceedings of the National Academy of Sciences, India **14** (1944), 1–2.

[15] J. Cilleruelo, *Gaps in dense sidon sets*, The Electronic Journal of Combinatorial Number Theory **0** (2000).

[16] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*, 2nd ed., MIT Press, 2001.

[17] A. K. Dewdney, *Computer recreations*, Scientific American **253** (1985), no. 6, 16–20.

[18] ———, *Computer recreations*, Scientific American **254** (1986), no. 3, 14.

[19] A. Dollas, W. T. Rankin, and D. McCracken, *A new algorithm for Golomb ruler derivation and proof of the 19 mark ruler*, IEEE Transactions on Information Theory **44** (1998), 379–382.

[20] A. R. Eckler, *The construction of missile guidance codes resistant to random interference*, Bell System Tech. Journal (1960), 973–994.

[21] P. Erdős, *On a problem of Sidon in additive number theory*, Acta Scientiarum Mathematicarum, Univ. Szeged **15** (1953-54), 255–259.

[22] P. Erdős and R. Freud, *On sums of a sidon-sequence*, Journal of Number Theory **38** (1991), 196–205.

[23] P. Erdős, A. Sárközy, and V. Sós, *On sum sets of Sidon sets I.*, Journal of Number Theory **47** (1994), 329–347.

[24] P. Erdős and A. Rényi, *Additive properties of random sequences of positive integers*, Acta Arithmetica **6** (1960), 83–110.

[25] P. Erdős and P. Turan, *On a problem of Sidon in additive number theory and some related problems*, Journal. of the London Mathematical Society **16** (1941), 212–215, Addendum (by P. Erdős), ibid. 19(1944), 208.

[26] M. Gardner, *Mathematical games*, Scientific American (1972), no. 3, 108–112.

[27] S.W. Graham, B_h *sequences*, Proceedings of a Conference in Honor of Heine Halberstam, Birkhauser, 1996, pp. 337–355.

[28] Richard K. Guy, *Unsolved problems in number theory*, 2nd ed., Problem Books in Mathematics, Unsolved Problems in Intuitive Mathematics, I., ch. 10 Modular Difference Sets and Error Correcting Codes, Springer-Verlag, New York, 1994.

[29] H. Halberstam and K. F. Roth, *Sequences*, vol. I, Oxford University Press, 1966, (2nd ed. Springer-Verlag, New York, 1983).

[30] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, 5th ed., Oxford University Press, 1989.

[31] M. Helm, *Some remarks on the Erdős-Turàn conjecture*, Acta arithmetica **63** (1993), 373–378.

[32] X.-D. Jia, *On finite Sidon sequences*, Journal of number theory **44** (1993), 84–92.

[33] Martin Klazar, *Note on the maximum size of a Sidon set*, unpublished.

[34] E. J. Klieber, *Some difference triangles for constructing self-orthogonal codes*, IEEE Transactions on Information Theory **16** (1970), 237–238.

[35] T. Kløve, *Bounds on the size of optimal difference sets*, IEEE Transactions on Information Theory **34** (1988), 355–361.

[36] ______, *Bounds and construction for difference triangle sets*, IEEE Transactions on Information Theory **35** (1989), 879–886.

[37] D. E. Knuth, *The art of computer programming. volume 2: Seminumerical algorithms*, Addison-Wesley, 1969.

[38] Mihail N. Kolountzakis, *Probabilistic and constructive methods in harmonic analysis and additive number theory*, Ph.D. thesis, Stanford University, May 1994.

[39] ______, *On the uniform distribution in residue classes of dense sets of integers with distinct sums*, Journal of number theory **76** (1999), 147–153.

[40] A. W. Lam and D. V. Sarwate, *On optimum time-hopping patterns*, IEEE Transactions on Communications **36** (1988), 380–382.

[41] R. Lidl and H. Niederreiter, *Finite fields*, Encyclopedia of Mathematics, vol. 20, Cambridge University Press, 1997.

[42] Bernt Lindström, *An inequality for b_2-sequences*, Journal of Combinatorial Theory **6** (1969), 211–212.

[43] ______, *Finding finite B_2-sequences faster*, Mathematics of Computation **67** (1998), 1173–1178.

[44] ______, *Well distribution of Sidon sets in residue classes*, Journal of Number Theory **69** (1998), 197–200.

[45] R. Lorentzen and R. Nilsen, *Application of linear programming to the optimal difference triangle set problem*, IEEE Transactions on Information Theory **37** (1991), 1268–1270.

[46] William T. Rankin, *Optimal Golomb rulers: An exhaustive parallel search implementation*, Master's thesis, Duke University, department of Electrical Engineering, 1993.

[47] J. Robinson and A. Bernstein, *A class of binary recurrent codes with limited error propagation*, IEEE Transactions on Information Theory **13** (1967), 106–113.

[48] J. P. Robinson, *Optimal Golomb rulers*, IEEE Transactions on Computers **28** (1979), 943–944.

[49] ______, *Addendum to optimal Golomb rulers*, IEEE Transactions on Computers **32** (1983), 201.

[50] Imre Z. Ruzsa, *Solving a linear equation in a set of integers I*, Acta Arithmetica **LXV.3** (1993), 259–282.

[51] J. B. Shearer, *Some new optimum Golomb rulers*, IEEE Transactions on Information Theory **36** (1990), 183–184.

[52] S. Sidon, Mathematische Annalen **106** (1932), 539.

[53] J. Singer, *A theorem in finite projective geometry and some applications to number theory*, Transactions of the American Mathematical Society **43** (1938), 377–385.

[54] Zhenxiang Zhang, *Finding finite B_2-sequences with larger $m - a_m^{1/2}$*, Mathematics of Computation **63** (1994), 403–414.

[55] C. Zhi, F. Pingzhi, and J. Fan, *Disjoint difference sets, difference triangle sets and related codes*, IEEE Transactions on Information Theory **38** (1992), 518–522.

Appendix A: Source code

1 ruzsa.C

```
#define REVISION 1.35

/*
    This program will find the best possible Golomb ruler
    for p prime using Ruzsa construction.

    Usage:
      ./findall l p

    Variations: -DCHECK    -> check if the produced rulers
      are actually golomb.

    p must be prime. It will find all golomb rulers of sizes
    l..p-1 using the construction for p.

*/


typedef long long int64;
typedef unsigned int uint32;
typedef unsigned int myint; // datatype for the sort

#include <iostream.h>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <signal.h>
#include <unistd.h>

uint32 a[230000];

#include "common.C"

/*
  Function: generate_one_lindstrom(g,p,f)

    Compute the Sidon set using Lindstrom construction for
```

```
    parameters g,p,f

*/

void
generate_one_lindstrom(uint32 g,uint32 p,uint32 f, uint32 a[])
{
  uint32 c = p*f;
  int64 d = (p-1)*g;

  uint32 z = p*(p-1);

  for(uint32 i=1;i<=p-1;i++)
    {
      int64 m = (c + d) % z;

      a[i] = m;

      c += p*f;
      d = (d*g) % z;
    }
}

/*
  Function: generate_one(g,p)

    Compute the Sidon set using I.Ruzsa construction for
    parameters g,p
*/


void
generate_one(uint32 g,uint32 p,uint32 f, uint32 a[])
{
  uint32 c = p;
  int64 d = (p-1)*g;
  uint32 z = p*(p-1);

  for(uint32 i=1;i<=p-1;i++)
    {
      int64 m = (c + d) % z;

      a[i] = m;

      c += p;
      d = (d*g) % z;
    }
}


long long ofs[10000];
uint32 arg[10000];

/*
```

```
  Function: modular_extract(a,n,p,l)

    For sequence a[i] find all maximum offsets
    by truncating down to l elements
*/

/* DO SOME PREPROCESSING TO SPEED THINGS UP:
      a[n+1]...a[2n] <- a[1]+p ... a[n]+p
*/

long long b[150000]; /* the unwound sequence */

void modular_extract(uint32 a[], uint32 n, uint32 z, uint32 l)
{
  uint i,j,k;

  for(i=1;i<=n;i++) { b[i] = a[i]; b[n+i] = (long long)a[i]+z; }

  for(i=1;i<=n-l+1;i++) ofs[i] = 0;

  for(k=1; k<=n-l+1; k++)
  #ifdef BREAK
  if(min_length[n-k+1] > (n-k+1)*(n-k+1))
  #endif
    for(i=1; i<=n; i++)
      {
        long long dofs = b[i+k] - b[i];

        if(dofs > ofs[k])
          {
            ofs[k] = dofs;
            arg[k] = i+k;
          }
      }

  for(i=1;i<=n-l+1;i++) if(arg[i] > n) arg[i] -= n;
}

/*
  Fucntion: find_best(int b[], int a[], int p, int z)

    Print the best ruler found with (p-z) elements
    after maximum offsets has found
    the best shift. Store in b[]

*/

void find_best(long long b[], long long a[],uint32 p,uint32 z)
{
  uint32 j = arg[z];
  uint32 ofs = a[j];

  for(uint32 i=1;i<=p-z+1;i++) b[i] = a[j++] - ofs;
}
```

```
/*
  int FindAll(r,p):

  given an integer r actually find the best rulers with
  r,r+1,...,p-1 marks
  by truncating the construction for prime p after.

  Return suitable arrays max_ofs,max_mult,g
  used for the constructions.

  Do not actually find the rulers.
*/

uint32 max_ofs[1000], max_mult[1000], max_arg[1000];
uint32 g;

void ruzsa_extract(uint32 l,uint32 p)
{
  for(uint i=1;i<=p-l;i++) max_ofs[i] = 0;

  uint32 f = 1;

  /* find a primitive element */
  uint32 g1 = 1;
  while(!is_primitive(g1,p)) g1++;

  g = g1;

  // for(uint32 t=1;t<p-1;t++,g=(g*g1)%p) if(mygcd(t,p-1)==1)

  uint32 a0[230000];
  generate_one(g,p,1,a0);

  /* No need to vary prim. element since it
     produces the same rulers!! */

  for(uint m=1;m<=p-1;m++) if(mygcd(m,p-1)==1)
    {
      /* Modular-Multiply */
      for(uint i=1;i<=p-1;i++)
        a[i] = ((long long)m*a0[i])  % (p*(p-1));

      sort(a,p-1);

      modular_extract(a,p-1,p*(p-1),l);

      for(int i=1;i<=p-l;i++)
        if(ofs[i] > max_ofs[i])
          {
            max_ofs[i] = ofs[i];
            max_mult[i] = m;
```

```
            #ifdef CHECK
            long long b1[70000];
            find_best(b1,b,p-1,i);
            cout << i << ":";
            for(int l=1;l<=p-i;l++) cout << b1[l] << " ";

            cout << "(" << is_golomb(b1,p-i);

            if(is_golomb(b1,p-i)!=1)
              { cout << "error\n"; exit(1); }

            cout << ")\n";
            #endif
          }
    }
}

uint l,p;

int main(int argc, char *argv[])
{
  cerr << "RuzsaFind (fast) rev" << REVISION << "\n\n";

  if(argc != 2)
    {
      cerr << "\bUsage: ruzsa l p\n\n";
      exit(1);
    }

  #ifdef LASTPRIME
  p = atoi(argv[1]);
  l = p-1;
  while(!is_prime(l)) l--;
  #else
  l = atoi(argv[1]);
  p = atoi(argv[2]);
  #endif

  if(!is_prime(p))
    {
      cerr << p << " is not a prime number!\n";
      exit(1);
    }

  ruzsa_extract(l,p);

  fprintf(stderr,"%d %.3f sec\n",p,usertime());

  for(int n=l;n<p;n++)
   cout << n << " " << p*(p-1)-max_ofs[p-n] << " "
        << p << " " << g << " " << max_mult[p-n] << "\n";

  fflush(stdout);
```

```
  return 0;
}
```

2 bose-fast.C

```
#define REVISION 2.22

/*
  Find rulers using Bose-Chowla construction

    Usage:
      ./bose-fast l p

    p must be prime. It will find all golomb rulers of sizes
    l..p using the construction for p.

    Compile:
      g++ -O2 bose-fast.C -lLiDIA -lgmp -lm

    Needs LiDIA library for number theory
    (see http://www.informatik.tu-darmstadt.de/TI/LiDIA/)

    Variations:
         -DCHECK    -> check if the produced rulers
      are actually golomb.
         -DBREAK    -> stop serching when golomb rulers of
      size less than n^2-2n are found.
*/

#include <iostream.h>
#include <fstream.h>
#include <strstream.h>
#include <string.h>
#include <stdlib.h>
#include <signal.h>
#include <time.h>
#include <unistd.h>

#include <LiDIA/bigint.h>
#include <LiDIA/gf_element.h>
#include <LiDIA/timer.h>

typedef unsigned long int myint;
typedef unsigned long int uint32;


long long ofs[10000];
uint32 arg[10000];

/* Mininum data found */
long long min_length[100000];
long long min_mult[100000];

int currentstate;

uint m,i;
```

```
char fname[256],fieldname[256];


#include "common.C"

/*
  Function: modular_extract(a,n,p,l)

    For sequence a[i] find all maximum offsets
    by truncating down to l elements
*/

/* DO SOME PREPROCESSING TO SPEED THINGS UP:
      a[n+1]...a[2n] <- a[1]+p ... a[n]+p
*/

long long  b[250000]; /* the unwound sequence */

void modular_extract(uint32 a[], uint32 n, uint32 z, uint32 l)
{
  uint i,j,k;

  for(i=1;i<=n;i++) { b[i] = a[i]; b[n+i] = (long long)a[i]+z; }

  for(i=1; i<=n-l+1; i++) ofs[i] = 0;

  for(k=1; k<=n-l+1; k++)
  #ifdef BREAK
  if(min_length[n-k+1] > (n-k+1)*(n-k+1))
  #endif
    for(i=1; i<=n; i++)
      {
        long long dofs = b[i+k] - b[i];

        if(dofs > ofs[k])
          {
            ofs[k] = dofs;
            arg[k] = i+k;
          }
      }

  for(i=1;i<=n-l+1;i++) if(arg[i] > n) arg[i] -= n;
}



/*
  Fucntion: find_best(int b[], int a[], int p, int z)

    Print the best ruler found with (p-z) elements
    after maximum offsets has found
    the best shift. Store in b[]

*/
```

```
void find_best(myint b[], myint a[], uint n, uint z)
{
  uint j = arg[z];
  myint ofs = a[j];

  for(uint i=1;i<=n-z+1;i++) b[i] = a[j++] - ofs;
}




struct poly { long int d[3]; void print(void); };

void
poly::print(void)
{
   cout << d[2] << " x^2 + ";
   cout << d[1] << " x + ";
   cout << d[0] << " ";
   cout << "\n";
}

poly irred, prim;

int mod; /* mod = p */


/*
   Multiply two polynomials of degree at most 1 and reduce
    modulo an irreducible poly of degree 2

    The irreducible polynomial is assumed to
    have leading coeff 1
*/

poly
poly_mul(poly a, poly b)
{
  poly r;

  r.d[0] = (a.d[0]*b.d[0]) % mod ;
  r.d[1] = (a.d[1]*b.d[0] + a.d[0]*b.d[1]) % mod;
  r.d[2] = (a.d[1]*b.d[1]) % mod;

  int c = r.d[2];

  r.d[0] -= c * irred.d[0];
  r.d[1] -= c * irred.d[1];
  r.d[2] = 0;

  /* r.d[2] will be 0 */
  // r.d[2] -= c * irred.d[2];
```

```
  r.d[0] %= mod;
  r.d[1] %= mod;

  if(r.d[0]<0) r.d[0] += mod;
  if(r.d[1]<0) r.d[1] += mod;

  return r;
}


using namespace LiDIA;

Fp_polynomial min_polynom[100000];
Fp_polynomial field_poly;


myint a[120000],a0[120000];
uint n=0;

/*
    Generate the Bose-Chowla field p^2 into a0[1..n]
*/

void generate(uint p)
{
  galois_field field((long int)p, 2);

  field_poly = field.irred_polynomial();


  #ifdef DEBUG
  cout << "Generating field GF*(" << p << "^2) ...\n\n";

  cout << "This field has ";
  cout << field.number_of_elements();
  cout <<" elements.\n";
  #endif

  #ifdef DEBUG
  cout << "The defining polynomial of the field is\n";
  cout << field_poly
  cout << endl;
  #endif

  gf_element th(field), elem(field);

  elem.assign_primitive_element(field);

  cout << "A primitive element is: " << elem << "\n";
  cout << "Order of element is : " << elem.order() << "\n";

  th = elem;

  Fp_polynomial fp = field_poly;
```

```
  ((bigint)fp[0]).longify(irred.d[0]);
  ((bigint)fp[1]).longify(irred.d[1]);
  ((bigint)fp[2]).longify(irred.d[2]);

  fp = elem.polynomial_rep();

  ((bigint)fp[0]).longify(prim.d[0]);
  ((bigint)fp[1]).longify(prim.d[1]);
  ((bigint)fp[2]).longify(prim.d[2]);

  if(irred.d[2] != 1)
    {
      cout
       <<"irreducible has not so good leading coeffficient\n";

      exit(1);
    }

  currentstate = 1;

  poly theta = prim;
  mod = p;

  n=0;

  for(i=1;i<p*p;i++)
    {
        if(theta.d[1] - prim.d[1] == 0) a0[++n] = i;

        theta = poly_mul(theta,prim);
    }

  if(n != p) { cout << "error! n=" << n << "\n"; exit(1); }

  ofstream of(fieldname);

  of << "Field: " << p << "^2\n";
  of << "Irreducible polynomial: " << field_poly << "\n";
  of << "Primitive element: " << elem << "\n";
  of << "\n";

  for(int i=1;i<=p;i++) of << a0[i] << "\n";
}


/*
  Compute all rulers of sizes l...p
*/

void
compute(uint l, uint p)
{
  ifstream fieldin(fieldname);
```

```
  if(fieldin)
    {
      cout << "\nReading field from " << fieldname << "\n";

      char t[500];
      fieldin.getline(t,500);
      fieldin.getline(t,500);
      fieldin.getline(t,500);
      fieldin.getline(t,500);

      n=0;
      for(int i=1;i<=p;i++) fieldin >> a0[++n];

      cout << "Field: " << a0[1]
        << " " << a0[2] << " ... " << a0[n] << "\n";
    }
  else
    {
      generate(p);
    }

    #ifdef CHECK
      int is = is_golomb(a0,n);
      cout << "\n";
      cout << "is_golomb: " << is << "\n";

      if(is==0) { cout << "error!"; exit(1); }
    #endif

    cout << "Varying f to produce new rulers...\n";
    currentstate = 2;

    for(m=1;m<=p*p-1;m++) if( mygcd(m,p*p-1) == 1 ) {

      /* Modular-Multiply by m*/
      for(uint i=1;i<=n;i++)
        a[i] = (((long long)m)*a0[i]) % (p*p-1);

      sort(a,n);

      modular_extract(a,n,p*p-1,l);

      for(uint n=l;n<=p;n++)
        {
          uint d =
            (p*p-1-ofs[p-n+1]); /* length of ruler computed */

          #ifdef CHECK
            int j = arg[p-n+1];
            myint shift = a[j];

            myint b[120000];
            for(int i=1;i<=n;i++) b[i] = a[j++]-shift;
```

```
            //cout << n << ": ";
            //cout << "(" << is_golomb(b,n) << ")";
            //print(b,n);

            if(!is_golomb(b,n)) { cout << "error\n"; exit(1); }
          #endif

          if(d < min_length[n])
            {
              min_length[n] = d;
              min_mult[n] = m;
            }
        } // n

      /* Break the loop if we found all rulers
         with sizes less than n*n */

      #ifdef BREAK
      uint n;
      for(n=l;n<=p;n++) if(min_length[n]>=(long long)n*n) break;
      if(n>p) break;
      #endif

    } // m
}


uint l,u;


/* print results so far */
void quit(int t)
{
  {
      ofstream out(fname);

    for(long long  t=l;t<=u;t++)
      out
      << t << " "
      << min_length[t] << " "
      << min_mult[t] << " "
      << (t*t-(signed long long)min_length[t]) << "\n" ;

    out.close();
  }

    if(currentstate == 1)
      cout << "Computing field i=" << i << "/" << u*u << "\n";
    else
      {
        cout << "Modular multiply m=" <<m << "/"
             << u*(u-1) << "\n";
```

```
        for(long long t=l;t<=u;t++)
          cout
          << t << " "
          << min_length[t] << " "
          << min_mult[t] << " "
          << (t*t-(signed long long)min_length[t]) << "\n";

        cout << "\n";
      }
}


int main(int argc,char *argv[])
{
  cout << "Bose-Chowla (fast) rev" << REVISION << "\n\n";

  l = atoi(argv[1]);
  u = atoi(argv[2]);

  if(argc != 3)
    {
      cout << "usage: bose-fast l u\n\n";
      exit(1);
    }

  #ifdef LASTPRIME
  l = u-1;
  while(!is_prime(l-1)) l--;
  #endif

  if(!is_prime(u))
    {
      cerr << u << " is not a prime number!\n";
      exit(1);
    }

  sprintf(fname,"outquick%d.dat",u);
  sprintf(fieldname,"gfruler%d.dat",u);

  printf("Constructing Bose for %d");
  printf(" and truncating down to %d\n",u,l);

  for(uint i=l;i<=u;i++) min_length[i] = (long long)1<<40;

  signal(SIGINT,quit);

  compute(l,u);

  printf("%d %.3f\n",u,usertime());

  /* Write results */

    for(long long t=l;t<=u;t++)
      cout
```

```
      << t << " "
      << min_length[t] << " "
      << min_mult[t] << " "
      << (t*t-(signed long long)min_length[t]) << "\n";

    ofstream out(fname);

    for(long long  t=l;t<=u;t++)
      out
      << t << " "
      << min_length[t] << " "
      << min_mult[t] << " "
      << (t*t-(signed long long)min_length[t]) << "\n" ;

    out.close();

    cout.flush();
}
```

3 common.C

```
/*
  Functions: is_golomb(n,a[])

    Check if given set is a golomb ruler in time O(n^2)

    assume that a[i] are sorted
*/

char *dist = new char[16000000];

template<class myint>
int is_golomb(myint a[], uint n)
{
    //uint32 dist[1000000];

    dist[0] = 1;
    for(uint i=1;i<=a[n];i++) dist[i] = 0;

    for(uint i=1;i<=n;i++)
      for(uint j=1;j<i;j++)
        {
          if(a[i]<0) return 0;

          int d = a[i] - a[j];

          if(dist[d] == 1) return 0;

          dist[d] = 1;
        }

    return 1;
}

/*
  Function: is_primitive(g,p)

   Return if g is primitive in Z_p^* and also fill the values
   of g_pow with the powers of g
*/


template<class myint>
int
is_primitive(myint g, myint p)
{
  myint r = g;

  for(myint i=1;i<=(p-1)/2;i++)
    {
      // here r = g^i mod p
      if(r==1) return 0;
      r = (r*g)%p;
```

```
    }

  return 1;
}

/*
  Function: sort(a[],n)

    Sort the array a[1]..a[n] using QuickSort in time O(nlogn)
*/

#include "sort.C"

template<class myint>
void
sort(myint a[], uint n)
{
  quicksort(&a[1],n);
}


/*
  Function: is_prime(p)

    Check if p is prime by dividing with all integers <sqrt(p). Time O(sqrt(p))

*/

template<class myint>
int is_prime(myint p)
{
  if(p==1) return 0;

  for(myint i=2;i*i<=p;i++)
      if(p % i  == 0) return 0;

  return 1;
}

/*
  Function: gcd(a,b)

    Find the greatest common divisor of a and b using
    Euclid's algorithm
*/

template<class myint>
int mygcd(myint a,myint b)
{
  while(a!=0 && b!=0)
    if(a>b) a %= b; else b %= a;

  return a+b;
}
```

```
/*
  Print the sequence a[1..n]
*/

template<class myint>
void
print(myint a[], int n)
{
  for(int i=1;i<=n;i++) cout << a[i] << " "; cout << "\n";
}


double usertime()
{
  return (double)clock()/CLOCKS_PER_SEC;
}
```