Discrete Applied Mathematics 212 (2016) 81–87

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

A computational substantiation of the d-step approach to the number of distinct squares problem

Antoine Deza, Frantisek Franek *, Mei Jiang

Advanced Optimization Laboratory, Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada

ARTICLE INFO

Article history:
Received 30 September 2014
Received in revised form 14 April 2016
Accepted 19 April 2016
Available online 28 May 2016

Keywords:
String
Square
Primitively rooted square
Maximum number of distinct primitively rooted squares
Parameterized approach
$(d, n-d)$ table

ABSTRACT

Motivated by the recent validation of the d-step approach for the number of runs problem, we investigate the largest possible number $\sigma_d(n)$ of distinct primitively rooted squares over all strings of length n with exactly d distinct symbols. New properties of $\sigma_d(n)$ are presented, and the notion of s-cover is introduced with an emphasis on the recursive computational determination of $\sigma_d(n)$. In particular, we were able to determine all values of $\sigma_2(n)$ for $n \le 70$, $\sigma_3(n)$ for $n \le 45$ and $\sigma_4(n)$ for $n \le 38$. These computations reveal the unexpected existence of pairs (d, n) satisfying $\sigma_{d+1}(n+2) - \sigma_d(n) > 1$ such as $(2, 33)$ and $(2, 34)$, and of three consecutive equal values: $\sigma_2(31) = \sigma_2(32) = \sigma_2(33)$. Noticeably, we show that among all strings of length 33, the maximum number of distinct primitively rooted squares cannot be achieved by a non-ternary string.

1. Introduction

The notion of an r-cover was introduced by Baker, Deza, and Franek [1] as a means to represent the distribution of the runs in a string and thus describe the structure of the run-maximal strings. Ignoring the number of distinct symbols d in the string, a key assertion states that essentially any run-maximal string has an r-cover. This fact was used in [2] to compute values of the maximum number of runs for strings of previously intractable lengths, and to provide computational substantiation for the d-step approach to the problem of the maximum number of runs proposed by Deza and Franek [4]. Recently, Bannai et al. [3] proved that the number of runs in a string is at most its length minus 3 using the maximal Lyndon roots of runs. Considering the largest possible number $\rho_d(n)$ of runs over all strings of length n with exactly d distinct symbols, Deza and Franek [4] conjectured that $\rho_d(n) \le n-d$ and $\rho_d(n) \le n-d-1$ for $n \ge 2d+1$ which was proven by Bannai et al. [3]. The bound was slightly improved to $\rho_d(n) \le n-d-2$ for $n \ge 2d+5$ by Deza and Franek [5] and, consequently, the number of runs in a string of length at least 9 is at most its length minus 4. Fischer, Holub, I, and Lewenstein further exploited the maximal Lyndon root approach and strengthened the upper for the maximum number of runs for binary strings in [9].

In this paper, we present a method of computing square-maximal strings similar to the one used for runs in [2] and similarly based on the d-step approach. We introduce the notion of s-cover which is used to speed up computations of the maximum number of distinct primitively rooted squares allowing computing $\sigma_d(n)$ for previously intractable values of d and n. The paper is organized as follows: Section 2 gives the basic facts and notation, Section 3 discusses the computational approach to the number of distinct primitively rooted squares, Section 4 introduces a heuristic for speeding up the computation, Section 5 discusses how s-covered string can be generated, Section 6 discusses how to compute $\sigma_d(n)$ values,

* Corresponding author.
E-mail addresses: deza@mcmaster.ca (A. Deza), franek@mcmaster.ca (F. Franek), jiangm5@mcmaster.ca (M. Jiang).

http://dx.doi.org/10.1016/j.dam.2016.04.025

Table 1
$(d, n-d)$ table for $\sigma_d(n)$ with $2 \le d \le 20$ and $2 \le n-d \le 20$ where the main diagonal corresponding to $n = 2d$ is shown in bold.

		$n-d$																		
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	2	**2**	2	3	3	4	5	6	7	7	8	9	10	11	12	12	13	13	14	15
	3	2	**3**	3	4	4	5	6	7	8	8	9	10	11	12	13	13	14	14	15
	4	2	3	**4**	4	5	5	6	7	8	9	9	10	11	12	13	14	14	15	15
	5	2	3	4	**5**	5	6	6	7	8	9	10	10	11	12	13	14	15	15	16
	6	2	3	4	5	**6**	6	7	7	8	9	10	11	11	12	13	14	15	16	16
	7	2	3	4	5	6	**7**	7	8	8	9	10	11	12	12	13	14	15	16	17
	8	2	3	4	5	6	7	**8**	8	9	9	10	11	12	13	13	14	15	16	17
	9	2	3	4	5	6	7	8	**9**	9	10	10	11	12	13	14	14	15	16	17
	10	2	3	4	5	6	7	8	9	**10**	10	11	11	12	13	14	15	15	16	17
d	11	2	3	4	5	6	7	8	9	10	**11**	11	12	12	13	14	15	16	16	17
	12	2	3	4	5	6	7	8	9	10	11	**12**	12	13	13	14	15	16	17	?
	13	2	3	4	5	6	7	8	9	10	11	12	**13**	13	14	14	15	16	17	18
	14	2	3	4	5	6	7	8	9	10	11	12	13	**14**	14	15	15	16	17	18
	15	2	3	4	5	6	7	8	9	10	11	12	13	14	**15**	15	16	16	17	18
	16	2	3	4	5	6	7	8	9	10	11	12	13	14	15	**16**	16	17	17	18
	17	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	**17**	17	18	18
	18	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	**18**	18	19
	19	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	**19**	19
	20	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	**20**

Section 7 discusses how to compute $\sigma_d(2d)$ values. In Section 8 some additional theoretic properties of $\sigma_d(n)$ not presented in [6] are discussed. The computational results are summarized in Section 9.

2. Notations

We encode a square as a triple (s, e, p) where s is the starting position of the square, e is the ending position of the square, and p is its period. Note that $e = s + 2p - 1$. The *join* $x[i_1 .. i_k] \vee x[j_1 .. j_m]$ of two substrings of a string $x = x[1 .. n]$ is defined if $i_1 \le j_1 \le i_k + 1$ and then $x[i_1 .. i_k] \vee x[j_1 .. j_m] = x[i_1 .. \max\{i_k, j_m\}]$, or if $j_1 \le i_1 \le j_m + 1$ and then $x[i_1 .. i_k] \vee x[j_1 .. j_m] = x[j_1 .. max\{i_k, j_m\}]$. In other words, the join is defined when the two substrings either are adjacent or overlapping. The join $S_1 \vee S_2$ of two squares of x encoded as $S_1 = (s_1, e_1, p_1)$ and $S_2 = (s_2, e_2, p_2)$ is defined as the join $x[s_1 .. e_1] \vee x[s_2 .. e_2]$. The alphabet of x is denoted by $\mathcal{A}(x)$, (d, n)-string refers to a string of length n with exactly d distinct symbols, $\boldsymbol{s}(x)$ denotes the number of distinct primitively rooted squares in a string x, and $\sigma_d(n)$ refers to the maximum number of distinct primitively rooted squares over all (d, n)-strings. A singleton is a symbol which occurs exactly once in the string under consideration. For the empty string ε, we set $\boldsymbol{s}(\varepsilon) = 0$ and $\sigma_d(0) = 0$. In the d-step approach the main tool is the $(d, n-d)$ table of the $\sigma_d(n)$ values where the row index represents d while the column index represents $n-d$ rather than the usual n. A 20×20 fragment of the table with computed values is shown in Table 1, see [7] for a table with all currently computed values. An important aspect of the d-step approach is the fact that the bounding of $\sigma_d(n)$ is determined by the bounding on the main diagonal, i.e. $\sigma_d(2d)$. More precisely, $\sigma_d(n) \le n - d$ for any $n \ge d \ge 2$ if and only if $\sigma_d(2d) = d$ for any $d \ge 2$. Additional properties $\sigma_d(n)$ are discussed and used in the following sections, see [4,6] for details.

3. Computational approach to distinct primitively rooted squares

In the computational framework for determining $\sigma_d(n)$ we will be discussing later, we first compute a lower bound of $\sigma_d(n)$ denoted as $\sigma_d^-(n)$. It is enough to consider (d, n)-strings x that could achieve $\boldsymbol{s}(x) > \sigma_d^-(n)$ for determining $\sigma_d(n)$, thus significantly reducing the search space. The purpose of this section is to introduce the necessary conditions that guarantee that for such an x, $\boldsymbol{s}(x) > \sigma_d^-(n)$ for a given $\sigma_d^-(n)$. The necessary conditions are the existence of an *s-cover* and a sufficient *density* of the string, see Lemmas 5, 9 and 10. The s-cover is guaranteed through generation, while the density is verified incrementally during the generation at the earliest possible stages. Note that the notion of s-cover, though similar to r-cover for runs [1,2], is slightly different.

Definition 1. An *s-cover* of a string $x = x[1 .. n]$ is a sequence of primitively rooted squares $\{S_i = (s_i, e_i, p_i) \mid 1 \le i \le m\}$ so that

(1) for any $1 \le i < m$, $s_i < s_{i+1} \le e_i + 1$ and $e_i < e_{i+1}$, i.e. two consecutive squares are either adjacent or overlapping;
(2) $\bigvee_{1 \le i \le m} S_i = x$;
(3) for any occurrence of square S in x, there is $1 \le i \le m$ so that S is a substring of S_i, denoted by $S \subseteq S_i$.

Lemma 2. *The s-cover of a string is unique.*