ISSN 0022-0000
ELSEVIER
Journal of Computer and System Sciences
Available online at www.sciencedirect.com
SciVerse ScienceDirect

The maximal number of cubic runs in a word

M. Crochemore, C. Iliopoulos, M. Kubica, J. Radoszewski,
W. Rytter, T. Waleń

PII: S0022-0000(11)00150-4
DOI: 10.1016/j.jcss.2011.12.005
Reference: YJCSS 2561

To appear in: *Journal of Computer and System Sciences*

Received date: 19 September 2010
Revised date: 18 April 2011
Accepted date: 17 November 2011

Please cite this article in press as: M. Crochemore et al., The maximal number of cubic runs in a word, *J. Comput. System Sci.* (2011), doi:10.1016/j.jcss.2011.12.005

Journal of Computer and System Sciences ●●●●, ●●●, ●●●

The maximal number of cubic runs in a word

M. Crochemore[1,2], C. Iliopoulos[1,3], M. Kubica[4], J. Radoszewski[4,*], W. Rytter[4,5], T. Waleń[4]

[1] *King's College London, London WC2R 2LS, UK*
[2] *Université Paris-Est, France*
[3] *Digital Ecosystems & Business Intelligence Institute, Curtin University of Technology, Perth WA 6845, Australia*
[4] *Dept. of Mathematics, Informatics and Mechanics, University of Warsaw, ul. Banacha 2, 02-097 Warsaw, Poland*
[5] *Dept. of Math. and Informatics, Copernicus University, ul. Chopina 12/18, 87-100 Toruń, Poland*

Highlights

▶ We introduce the notion of a cubic run in a word, it is a variant of a run. ▶ The exact structure of cubic runs in Fibonacci strings is described. ▶ A lower and an upper bound for the maximal number of cubic runs in a word is given. ▶ An improved upper bound for binary words is derived.

The Maximal Number of Cubic Runs in a Word

M. Crochemore[a,b], C. Iliopoulos[a,c], M. Kubica[d], J. Radoszewski[d,1,*],
W. Rytter[d,e,2], T. Waleń[d]

[a] *King's College London, London WC2R 2LS, UK*
[b] *Université Paris-Est, France*
[c] *Digital Ecosystems & Business Intelligence Institute, Curtin University of Technology, Perth WA 6845, Australia*
[d] *Dept. of Mathematics, Informatics and Mechanics, University of Warsaw, ul. Banacha 2, 02-097 Warsaw, Poland*
[e] *Dept. of Math. and Informatics, Copernicus University, ul. Chopina 12/18, 87-100 Toruń, Poland*

Abstract

A run is an inclusion maximal occurrence in a word (as a subinterval) of a factor in which the period repeats at least twice. The maximal number of runs in a word of length n has been thoroughly studied, and is known to be between $0.944\,n$ and $1.029\,n$. The proofs are very technical. In this paper we investigate cubic runs, in which the period repeats at least three times. We show the upper bound on their maximal number, $\mathsf{cubic\text{-}runs}(n)$, in a word of length n: $\mathsf{cubic\text{-}runs}(n) < 0.5\,n$. The proof of linearity of $\mathsf{cubic\text{-}runs}(n)$ utilizes only simple properties of Lyndon words and is considerably simpler than the corresponding proof for general runs. For binary words, we provide a better upper bound $\mathsf{cubic\text{-}runs}_2(n) < 0.48\,n$ which requires computer-assisted verification of a large number of cases. We also construct an infinite sequence of words over a binary alphabet for which the lower bound is $0.41\,n$.

Keywords: run in a word, Lyndon word, Fibonacci word

1. Introduction

Repetitions and periodicities in words are two of the fundamental topics in combinatorics on words [2, 14]. They are also important in other areas: lossless compression, word representation, computational biology etc. Repetitions are

[*] Corresponding author. Tel.: +48-22-55-44-484, fax: +48-22-55-44-400. Some parts of this paper were written during the corresponding author's Erasmus exchange at King's College London.

Email addresses: `maxime.crochemore@kcl.ac.uk` (M. Crochemore), `csi@dcs.kcl.ac.uk` (C. Iliopoulos), `kubica@mimuw.edu.pl` (M. Kubica), `jrad@mimuw.edu.pl` (J. Radoszewski), `rytter@mimuw.edu.pl` (W. Rytter), `walen@mimuw.edu.pl` (T. Waleń)

[1] The author is supported by grant no. N206 568540 of the National Science Centre.
[2] The author is supported by grant no. N206 566740 of the National Science Centre.

studied from different points of view: classification of words not containing repetitions of a given exponent, efficient identification of factors being repetitions of different types and, finally, computing the bounds on the number of repetitions of a given exponent that a word may contain, which we consider in this paper. Both the known results in the topic and a deeper description of the motivation can be found in a survey by Crochemore et al. [5].

The concept of runs (also called maximal repetitions) has been introduced to represent all repetitions in a word in a succinct manner. The crucial property of runs is that their maximal number in a word of length n (denoted as $\mathsf{runs}(n)$) is $O(n)$, see Kolpakov & Kucherov [11]. This fact is the cornerstone of any algorithm computing all repetitions in words of length n in $O(n)$ time. Due to the work of many people, much better bounds on $\mathsf{runs}(n)$ have been obtained. The lower bound $0.927\,n$ was first proved by Franek & Yang [9]. Afterwards, it was improved by Kusano et al. [13] to $0.944565\,n$ employing computer experiments, and recently by Simpson [20] to $0.944575712\,n$. On the other hand, the first explicit upper bound $5\,n$ was settled by Rytter [17], afterwards it was systematically improved to $3.48\,n$ by Puglisi et al. [16], $3.44\,n$ by Rytter [19], $1.6\,n$ by Crochemore & Ilie [3, 4] and $1.52\,n$ by Giraud [10]. The best known result $\mathsf{runs}(n) \leq 1.029\,n$ is due to Crochemore et al. [6], but it is conjectured [11] that $\mathsf{runs}(n) < n$. The maximal number of runs was also studied for special types of words and tight bounds were established for Fibonacci words [11, 18] and more generally Sturmian words [1].

The combinatorial analysis of runs is strongly related to the problem of estimation of the maximal number of squares in a word. In the latter problem the gap between the upper and lower bound is much larger than for runs [5, 8]. However, a recent paper [12] by some of the authors shows that introduction of integer exponents larger than 2 may lead to obtaining tighter bounds for the number of corresponding repetitions.

In this paper we introduce and study the concept of cubic runs, in which the period is at least three times shorter than the run itself. We describe the structure of cubic runs in Fibonacci words (Section 3). Then we show the following bounds on their maximal number, $\mathsf{cubic\text{-}runs}(n)$, in a word of length n:

$$0.41\,n < \mathsf{cubic\text{-}runs}(n) < 0.5\,n\ .$$

The upper bound is achieved by analyzing Lyndon words (i.e., words that are primitive and minimal/maximal in the class of their cyclic equivalents) that appear as periods of cubic runs (Section 4). In Section 6 we improve this bound for *binary* words to $0.48\,n$ by examining short factors of the word. As for the lower bound, we describe an infinite family of binary words that contain more than $0.41\,n$ cubic runs (Section 5). In particular, we improve both the lower and the (binary) upper bound from the conference version of the paper [7].

2. Preliminaries

We consider *words* u over a finite alphabet Σ, $u \in \Sigma^*$; the empty word is denoted by ε; the positions in u are numbered from 1 to $|u|$. By Σ^n we denote

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
$\mathsf{cubic\text{-}runs}_2(n)$	1	1	1	2	2	2	3	3	3	4	4	5	5	5	6	7	7	7	8	8	8	9	9	10	10	10	11

Table 1: The maximum number $\mathsf{cubic\text{-}runs}_2(n)$ of cubic runs in a binary word of length n for $n = 3, \ldots, 29$. Example binary words for which the maximal number of cubic runs is attained are shown in the following Table 2.

the set of all words of length n from Σ^*. By u^R we denote the reversed word u. By $\mathsf{Alph}(u)$ we denote the set of all letters of u. For $u = u_1 u_2 \ldots u_n$, let us denote by $u[i\,..\,j]$ a *factor* of u equal to $u_i \ldots u_j$ (in particular $u[i] = u[i\,..\,i]$). Words $u[1\,..\,i]$ are called prefixes of u, and words $u[i\,..\,n]$ are called suffixes of u.

We say that a positive integer q is the (shortest) *period* of a word $u = u_1 \ldots u_n$ (notation: $q = \mathsf{per}(u)$) if q is the smallest positive number, such that $u_i = u_{i+q}$ holds for all $1 \le i \le n - q$.

If $u = w^k$ (k is a non-negative integer), that is $u = ww \ldots w$ (k times), then we say that u is the k^{th} power of the word w. A *square* is the 2^{nd} power of some non-empty word. The *primitive root* of a word u, denoted $\mathsf{root}(u)$, is the shortest word w such that $w^k = u$ for some positive integer k. We call a word u *primitive* if $\mathsf{root}(u) = u$, otherwise it is called *non-primitive*. We say that words u and v are cyclically equivalent (or that one of them is a cyclic rotation of the other) if $u = xy$ and $v = yx$ for some $x, y \in \Sigma^*$. It is a simple and well-known observation, that if u and v are cyclically equivalent then $|\mathsf{root}(u)| = |\mathsf{root}(v)|$.

A *run* (also called a maximal repetition) in a word u is an interval $[i\,..\,j]$ such that:

- the period q of the associated factor $u[i\,..\,j]$ satisfies $2q \le j - i + 1$,
- the interval cannot be extended to the left nor to the right, without violating the above property, that is, $u[i-1] \ne u[i+q-1]$ and $u[j-q+1] \ne u[j+1]$, provided that the respective letters exist.

By $\mathcal{R}(u)$ we denote the set of runs in u, additionally $\mathsf{runs}(u) = |\mathcal{R}(u)|$.

A *cubic run* is a run $[i\,..\,j]$ for which the shortest period q satisfies $3q \le j - i + 1$. By $\mathcal{CR}(u)$ we denote the set of cubic runs in u, additionally denote $\mathsf{cubic\text{-}runs}(u) = |\mathcal{CR}(u)|$. For positive integer n, by $\mathsf{cubic\text{-}runs}(n)$ we denote the maximum of $\mathsf{cubic\text{-}runs}(u)$ for all $u \in \Sigma^n$, and by $\mathsf{cubic\text{-}runs}_2(n)$ we denote the maximum over all such binary words.

For simplicity, in the rest of the text we sometimes refer to runs or cubic runs as to occurrences of corresponding factors of u.

Example. All cubic runs for an example Fibonacci word are shown in Figure 1.

n	cubic-runs$_2(n)$	u
3	1	000
6	2	000111
9	3	000111000
12	4	000100010001
14	5	00010001000111
17	6	00010001000111000
18	7	000111000111000111
21	8	000111000111000111000
24	9	000111000111000111000111
26	10	00010001000111000111000111
29	11	00010001000111000111000111000

Table 2: Lexicographically smallest binary words $u \in \{0,1\}^n$, for which cubic-runs$(u) =$ cubic-runs$_2(n)$ (see also Table 1).

3. Fibonacci Words

Let us start by analyzing the behavior of function cubic-runs for a very common benchmark in text algorithms, i.e., the Fibonacci words, defined recursively as:

$$F_0 = a, \quad F_1 = ab, \quad F_n = F_{n-1}F_{n-2} \quad \text{for} \quad n \geq 2\ .$$

Denote by $\Phi_n = |F_n|$, the n^{th} Fibonacci number (we assume that for $n < 0$, $\Phi_n = 1$) and by g_n the word F_n with the last two letters removed.

Lemma 1. *[15, 18] Each run in F_n is of the form $F_k \cdot F_k \cdot g_{k-1}$ (short runs) or $F_k \cdot F_k \cdot F_k \cdot g_{k-1}$ (long runs), and has a period Φ_k.*

Obviously, in Lemma 1 only runs of the form $F_k^3 \cdot g_{k-1}$ are cubic runs.

Denote by $\#occ(u, v)$ the number of occurrences (as a factor) of a word u in a word v.

Lemma 2. *For every $k, n \geq 0$:*

$$\#occ(F_k^3 \cdot g_{k-1},\ F_n) \ = \ \#occ(F_k^3,\ F_n)\ .$$

Proof. Each occurrence of F_k^3 within F_n must be followed by g_{k-1}, since otherwise it would form a run different from those specified in Lemma 1. □

Lemma 3. *For every $k \geq 2$ and $m \geq 0$:*

a) $\#occ(F_k^3,\ F_{m+k}) = \#occ(aaba,\ F_m)$,

b) $\#occ(aaba,\ F_m) = \Phi_{m-3} - 1$.

Proof. Recall the Fibonacci morphism φ:

$$\varphi(a) = ab, \quad \varphi(b) = a \ .$$

Recall that $F_n = \varphi^n(a)$. The following claim provides a useful tool for the proof of items (a) and (b).

Claim 4. *Assume $F_n = uvw$, where $u, v, w \in \{a, b\}^*$, $v[1] = a$ and either $w[1] = a$ or $w = \varepsilon$. Then there exist unique words u', v', w' such that:*

$$u = \varphi(u'),\ v = \varphi(v'),\ w = \varphi(w'),\ F_{n-1} = u'v'w' \ .$$

And conversely, if v' is a factor of some F_{n-1} and $v = \varphi(v')$ then v is a factor of F_n.

Proof. It is a straightforward consequence of the definition of φ and the fact that $F_n = \varphi(F_{n-1})$. □

Now we proceed to the actual proof of the lemma. We prove item (a) by induction on k. For $k = 2$ we show the following equalities:

$$\#occ(abaabaaba, F_{m+2}) = \#occ(ababaa, F_{m+1}) = \#occ(aaba, F_m) \ . \quad (1)$$

As for the first of the equalities (1), the occurrence of F_2^3 within F_{m+2} cannot be followed by the letter a (since this would imply a larger run, contradicting Lemma 1) and cannot be a suffix of F_{m+2} (since either F_4 or F_5 is a suffix of F_{m+2}). Thus:

$$\#occ(abaabaaba, F_{m+2}) = \#occ(abaabaabab, F_{m+2}) = \#occ(ababaa, F_{m+1}) \ .$$

The latter of the above equalities holds due to Claim 4, which applies here since no occurrence of $abaabaabab$ in F_{m+2} can be followed by the letter b (bb is not a factor of any Fibonacci word).

To prove the second equality (1), we apply a very similar approach: $ababaa$ is not a suffix of F_{m+1} and its occurrence cannot be followed by the letter a, since no Fibonacci word contains the factor aaa. Hence, by Claim 4:

$$\#occ(ababaa, F_{m+1}) = \#occ(ababaab, F_{m+1}) = \#occ(aaba, F_m) \ .$$

Finally, the inductive step for $k \geq 3$ also follows from Claim 4. Indeed, F_k^3 starts with the letter a and any of its occurrences in F_{m+k} is followed by the letter a, since, by Lemma 1, it is a part of a larger run $F_k^3 \cdot g_{k-1}$. Thus:

$$\#occ(F_k^3,\ F_{m+k}) = \#occ(F_{k-1}^3,\ F_{m+k-1}) \ .$$

The proof of item (b) goes by induction on m. For $m \leq 3$ one can easily check that $\#occ(aaba,\ F_m) = 0$, and there is exactly one occurrence of $aaba$ in F_4. The inductive step is a conclusion of the fact that for $m \geq 5$ the word F_m contains all occurrences of $aaba$ from F_{m-1} and F_{m-2} and one additional occurrence overlapping their concatenation:

$$\ldots ab\,\underbrace{a\,|\,aba}\,ab\ldots \qquad\qquad \ldots ab\,\underbrace{aab\,|\,a}\,ba\ldots$$

The case of $2 \nmid m$. The case of $2 \mid m$.

This concluded the proof of the lemma. □

Lemma 5. *For $n > 5$, the word F_n contains (see Fig. 1):*

- $\Phi_{n-5} - 1$ *cubic runs* $F_2^3 \cdot g_1$
- $\Phi_{n-6} - 1$ *cubic runs* $F_3^3 \cdot g_2$

  $\vdots$
- $\Phi_1 - 1$ *cubic runs* $F_{n-4}^3 \cdot g_{n-5}$.

Words $F_0, F_1, \ldots, F_5$ do not contain any cubic runs.

Proof. It is easy to check that words F_n for $n \le 5$ do not contain any cubic runs. Let $n > 5$ and $k \in \{2, 3, \ldots, n-4\}$. Denote $m = n - k$. Combining the formulas from Lemmas 2 and 3, we obtain that:

$$\begin{aligned}\#occ(F_k^3 \cdot g_{k-1},\ F_n) &= \#occ(F_k^3 \cdot g_{k-1},\ F_{m+k}) = \#occ(F_k^3,\ F_{m+k})\\ &= \#occ(aaba,\ F_m) = \Phi_{m-3} - 1\\ &= \Phi_{n-k-3} - 1\ .\end{aligned}$$

□

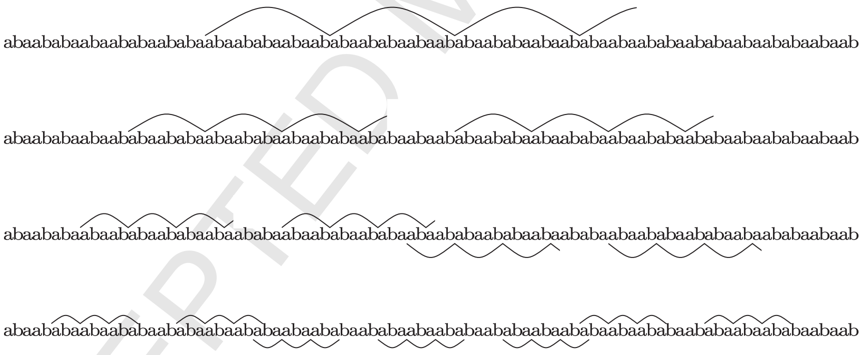
abaababaabaababaababaabaababaabaababaababaabaababaababaabaababaabaababaababaabaababaabaab
abaababaabaababaababaabaababaabaababaababaabaababaababaabaababaabaababaababaabaababaabaab
abaababaabaababaababaabaababaabaababaababaabaababaababaabaababaabaababaababaabaababaabaab
abaababaabaababaababaabaababaabaababaababaabaababaababaabaababaabaababaababaabaababaabaab

Figure 1: The structure of cubic runs in the Fibonacci word F_9. The cubic runs are distributed as follows: 1 run $F_5^3 \cdot g_4$, 2 runs $F_4^3 \cdot g_3$, 4 runs $F_3^3 \cdot g_2$, and 7 runs F_2^3.

We are now ready to describe the behaviour of the function cubic-runs(F_n). The following theorem not only provides an exact formula for it, but also shows a relationship between the number of cubic runs and the number of distinct cubes in Fibonacci words. This relationship is similar to the corresponding relationship between the number of (ordinary) runs and the number of (distinct) squares in Fibonacci words, which always differ exactly by 1, see [15, 18].

Theorem 6.

a) $\textsf{cubic-runs}(F_n) = \Phi_{n-3} - n + 2$.

b) $\lim_{n\to\infty} \frac{\textsf{cubic-runs}(F_n)}{|F_n|} = \frac{1}{\phi^3} \approx 0.2361$, *where* $\phi = \frac{1+\sqrt{5}}{2}$ *is the golden ratio.*

c) The total number of cubic runs in F_n equals the number of distinct cubes in F_n.

Proof. a) From Lemma 5 we obtain:

$$\textsf{cubic-runs}(F_n) \;=\; \sum_{i=1}^{n-5}(\Phi_i - 1) \;=\; \Phi_{n-3} - 3 - (n-5) \;=\; \Phi_{n-3} - n + 2\ .$$

b) It is a straightforward application of the formula from (a):

$$\lim_{n\to\infty} \frac{\textsf{cubic-runs}(F_n)}{|F_n|} = \lim_{n\to\infty} \frac{\Phi_{n-3} - n + 2}{\Phi_n} = \frac{1}{\phi^3}\ .$$

c) It suffices to note that the number of distinct cubes of length $3\Phi_{k+1}$ in $F_{k+1}^3 \cdot g_k$ is $|g_k| + 1 = \Phi_k - 1$, and thus the total number of distinct cubes in F_n equals:

$$\sum_{k=1}^{n-5} (\Phi_k - 1) = \Phi_{n-3} - n + 2 = \textsf{cubic-runs}(F_n).$$

□

4. Upper Bound of $0.5\,n$

Let $u \in \Sigma^n$. Let us denote by $\mathcal{I} = \{p_1, p_2, \ldots, p_{n-1}\}$ the set of inter-positions in u that are located *between* pairs of consecutive letters of u. To show the upper bound of $0.5\,n$ on the number of cubic runs in u, we will assign to each cubic run a set of interpositions from $\mathcal{I}$ (called a *handle* of the cubic run later on, formal definitions follow), so that these sets for different cubic runs are disjoint and each such set contains at least two elements. Clearly, this will imply that there are at most $\frac{n-1}{2}$ cubic runs in u.

Assume that Σ is totally ordered by $\leq$, which induces a lexicographical order on Σ^*, also denoted by $\leq$. We say that $\lambda \in \Sigma^*$ is a *Lyndon word* if it is primitive and minimal or maximal in the class of words that are cyclically equivalent to it. It is known (see [14]) that a Lyndon word has no non-trivial prefix that is also its suffix.

Definition 7. *We say that $F : \mathcal{R}(u) \to subsets(\mathcal{I})$ is a* handle function *for the runs in word u if the following conditions hold:*

$$F(v_1) \cap F(v_2) = \emptyset \quad \textit{for any } v_1 \neq v_2. \tag{2}$$

$$|F(v)| \geq 2 \quad \textit{for any } v \in \mathcal{CR}(u). \tag{3}$$

We say that $F(v)$ is the set of handles *of the run v.*

Obviously, if a word $u \in \Sigma^n$ admits a handle function then $\textsf{cubic-runs}(u) \leq \frac{n-1}{2}$.

We define a function $H : \mathcal{R}(u) \to \mathit{subsets}(\mathcal{I})$ as follows. Let v be a run with period q and let w be the prefix of v of length q. Let $w_{\min}$ and $w_{\max}$ be the minimal and maximal words (in lexicographical order) cyclically equivalent to w. $H(v)$ is defined as follows:

a) if $w_{\min} \neq w_{\max}$ then $H(v)$ contains all inter-positions in the middle of any occurrence of $w_{\min}^2$ in v, and in the middle of any occurrence of $w_{\max}^2$ in v,

b) if $w_{\min} = w_{\max}$ then $H(v)$ contains all inter-positions within v.

Example. For a cubic run $v_1 = (aabab)^3 aab$ we have $\textsf{per}(v_1) = 5$, $w = v_1[1\,..\,5] = aabab = w_{\min}$ and $w_{\max} = babaa$, see also Fig. 2a. For a cubic run $v_2 = b^4$ we have $\textsf{per}(v_2) = 1$, $w = v_2[1] = b = w_{\min} = w_{\max}$, see also Fig. 2b.

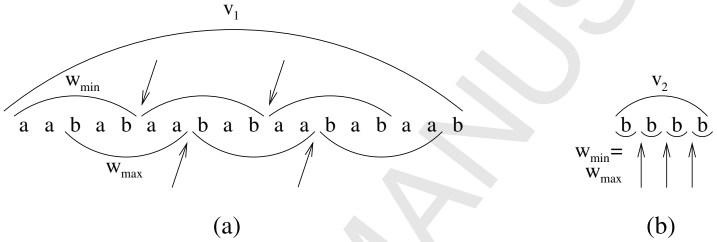

Figure 2: (a) For the cubic run v_1 with period greater than 1 we have $w_{\min} \neq w_{\max}$. (b) For the cubic run v_2 we have $w_{\min} = w_{\max} = b$ (a single-letter word). The inter-positions belonging to the sets $H(v_1)$ and $H(v_2)$ are indicated by arrows.

Lemma 8. *For any word $u \in \Sigma^*$, H is a handle function.*

Proof. Let us start by showing two simple properties of $w_{\min}$ and $w_{\max}$.

(P1) $w_{\min}$ and $w_{\max}$ are Lyndon words.

(P2) If $w_{\min} = w_{\max}$ (case (b) of the definition of $H(v)$), then $|w_{\min}| = 1$ and consequently each $p_i \in H(v)$ is located in the middle of $w_{\min}^2$.

As for the property (P1), by the definition of $w_{\min}$ and $w_{\max}$ we know that these words are lexicographically minimal and maximal respectively, hence it suffices to show that both words are primitive. This follows from the fact that, due to the minimality of q, w is primitive and that $w_{\min}$ and $w_{\max}$ are cyclically equivalent to w.

We show property (P2) by contradiction. Assume that $|w_{\min}| \geq 2$. By property (P1), $w_{\min} = w_{\max}$ is a Lyndon word. Therefore it contains at least two distinct letters, let us say: $a = w_{\min}[1]$ and $b = w_{\min}[i] \neq a$. If $b < a$ ($b > a$) then the cyclic rotation of $w_{\min} = w_{\max}$ by $i - 1$ letters is lexicographically smaller than $w_{\min}$ (greater than $w_{\max}$) and $w_{\min} \neq w_{\max}$ — a contradiction. Hence, the above assumption is false and $|w_{\min}| = 1$.

Using properties (P1) and (P2), in the following two claims we show that H satisfies conditions (2) and (3).

Claim 9. *$H(v_1) \cap H(v_2) = \emptyset$ for any two different runs v_1 and v_2 in u.*

Proof. Assume, to the contrary, that $p_i \in H(v_1) \cap H(v_2)$ is a handle of two different runs v_1 and v_2. By the definition of H and properties (P1) and (P2), p_i is located in the middle of two squares of Lyndon words: w_1^2 and w_2^2, where $|w_1| = \mathsf{per}(v_1)$ and $|w_2| = \mathsf{per}(v_2)$. Note that $w_1 \neq w_2$, since otherwise runs v_1 and v_2 would be the same. Without the loss of generality, we can assume that $|w_1| < |w_2|$. Thus the word w_1 is both a prefix and a suffix of w_2 (see Fig. 3), which contradicts the fact that w_2 is a Lyndon word. □

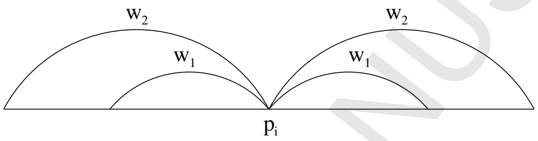

Figure 3: A situation where p_i is in the middle of two different squares w_1^2 and w_2^2.

Claim 10. *For any $v \in \mathcal{CR}(u)$, we have $|H(v)| \geq 2$.*

Proof. Let v be a cubic run. Recall that $3q \leq |v|$, where $q = \mathsf{per}(v)$. If $w_{\max} = w_{\min}$, then, by property (P2), $|w_{\min}| = 1$ and $|H(v)| = |v| - 1 \geq 2$.

If $w_{\max} \neq w_{\min}$, then it suffices to note that the first occurrences of each of the words $w_{\min}$ and $w_{\max}$ within v start no further than q positions from the beginning of v. Of course, they start at different positions. Hence, $w_{\min}^2$ and $w_{\max}^2$ are both factors of v and contribute different handles to $H(v)$. □

Thus we have showed that H satisfies both conditions of a handle function, which concludes the proof of the lemma. □

Theorem 11.

1. *$\mathsf{cubic\text{-}runs}(n) < 0.5\,n$.*
2. *For infinitely many n we have: $0.4\,n \leq \mathsf{cubic\text{-}runs}(n)$.*

Proof. The upper bound is a corollary of Lemma 8. As for the lower bound, define:

$$u = 0^3 1^3,\ v = 1^3 2^3,\ w = 2^3 0^3,\ x_k \ = \ (\ u^2\ 0^3\ v^2\ 1^3\ w^2\ 2^3\ )^k\ .$$

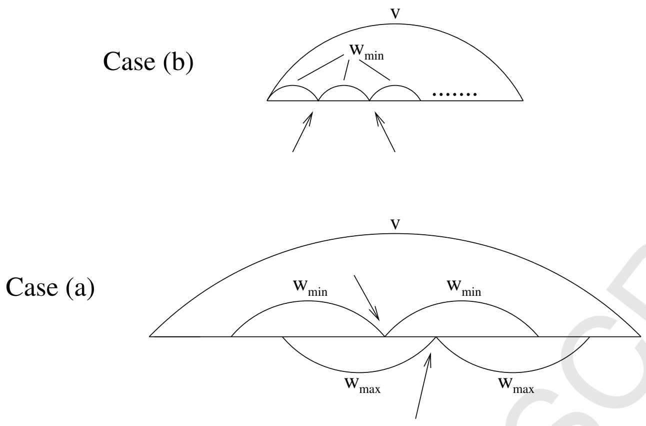

Figure 4: Illustration of the definition of H and Claim 10. The arrows in the figure point to the elements of $H(v)$ for cubic runs.

Observe that for any $k \geq 1$, the word x_k contains at least $18k - 1$ cubic runs. Indeed, we have $15k$ cubic runs with period 1, of the form 0^3, 1^3 or 2^3. Moreover, there are $3k - 1$ cubic runs with period 6: $2k$ cubic runs of the form $\left(0^3 1^3\right)^3$ or $\left(1^3 2^3\right)^3$, fully contained within each occurrence of x_1 in $x_k = (x_1)^k$, and $k - 1$ cubic runs of the form $\left(2^3 0^3\right)^3$, overlapping the concatenations of consecutive x_1's.

Note that for $k \geq 3$, the whole word x_k forms an additional cubic run. Hence, in this case the word x_k has length $45k$ and contains at least $18k$ cubic runs. Thus:

$$\textsf{cubic-runs}(x_k) \;\geq\; 0.4\,|x_k| = 0.4n \qquad \text{for } k \geq 3.$$

□

The lower bound can be improved in two ways: restricting words to be over binary alphabet and improving 0.4 to 0.41. The coefficient in the upper bound will be also slightly improved, for the case of binary alphabet (decreased by $\frac{1}{50}$). However even such small improvements require quite technical proofs.

5. Improving the Lower Bound

In this section we show an example sequence of *binary* words which gives the bound of $0.41\,n$. For this, we use the following morphism, which was found experimentally using a genetic algorithm:

$$\psi(a) = 001110, \quad \psi(b) = 0001110\ .$$

Recall that F_n is the n-th Fibonacci word.

n	$\lvert w_n \rvert$	cubic-runs$(w_n)/\lvert w_n \rvert$	w_n
0	1	0.16667	$0^2 1^3 0$
1	3	0.23077	$0^2 1^3 0^4 1^3 0$
2	5	0.26316	$0^2 1^3 0^4 1^3 0^3 1^3 0$
3	10	0.31250	$0^2 1^3 0^4 1^3 0^3 1^3 0^3 1^3 0^4 1^3 0$
4	17	0.33333	$0^2 1^3 0^4 1^3 0^3 1^3 0^3 1^3 0^4 1^3 0^3 1^3 0^4 1^3 0^3 1^3 0$
5	30	0.36145	...
6	49	0.36567	
7	83	0.38249	

Table 3: Characteristics of a few first elements of the sequence (w_n).

It appears that a sequence defined as $w_n = \psi(F_n)$ consists of cubic-run-rich words, see also Table 3. In particular, it can be checked experimentally that the word w_{20} (further denoted as w for brevity) of length 113 031 contains 46 348 cubic runs, hence cubic-runs$(w) > 0.41\,|w|$. Below we show that for infinitely many words of the form w^k, the density of cubic runs is more than 0.41.

Theorem 12 (Improved Lower Bound).
There are infinitely many binary words w^k, where $w = w_{20}$, such that:

$$\frac{r_k}{\ell_k} > 0.41 \ ,$$

where $r_k =$ cubic-runs(w^k), $\ell_k = |w^k|$.

Proof. We start the proof with the following claim, a similar property of the runs function (with different constants) was proved in [13].

Claim 13. *For any $k \geq 3$, $r_k = Ak - B$, where $A = r_4 - r_3$ and $B = 3r_4 - 4r_3$.*

Proof. We will first show that $r_{k+1} - r_k = r_4 - r_3$, i.e., that the increase of the number of cubic runs when concatenating w^k and w equals the corresponding increase when concatenating w^3 and w. Let $[i\,..\,j]$ be a cubic run in w^{k+1} ending within the last occurrence of w, that is, $j > k \cdot |w|$. In [13] it is proved (as Lemma 2) that the only run in w^{k+1} of length at least $2 \cdot |w|$ is the run equal to the word w^{k+1}. Hence, the cubic run $[i\,..\,j]$ either corresponds to the whole word w^{k+1} or satisfies $i > (k-2) \cdot |w|$. In both cases the cubic runs yield the same increase as when concatenating w to w^3. (Note that in the first case the cubic run forms only an extension of a cubic run already present in w^k, therefore it does not increase the number of cubic runs for any $k \geq 3$.)

This concludes that $r_{k+1} - r_k = r_4 - r_3$. From this formula we obtain that, for $k \geq 4$:

$$\begin{aligned} r_k &= r_{k-1} + r_4 - r_3 = r_{k-2} + 2 \cdot (r_4 - r_3) = \ldots \\ &= r_3 + (k-3) \cdot (r_4 - r_3) = k \cdot (r_4 - r_3) - (3r_4 - 4r_3) \ . \end{aligned}$$

One can easily check that the same formula holds also for $k = 3$. □

Now we complete the proof of Theorem 12. Using an extensive computer experiment one can obtain that:

$$r_3 = 139\,083 \quad \text{and} \quad r_4 = 185\,450, \quad \text{and recall that } |w| = 113\,031.$$

By Claim 13, for $k > \frac{10^4 \cdot B}{|w|}$ we obtain that:

$$\begin{aligned} \frac{r_k}{\ell_k} \;&=\; \frac{A\cdot k}{|w^k|} - \frac{B}{|w^k|} \;=\; \frac{r_4 - r_3}{|w|} - \frac{B}{|w|\cdot k} \\ &>\; \frac{185\,450 - 139\,083}{113\,031} - 0.0001 \;>\; 0.41\;. \end{aligned}$$

This concludes the proof of the theorem. □

6. Improving the Upper Bound in the Case of Binary Alphabet

Let $u \in \{0,1\}^n$. Recall that $\mathcal{I} = \{p_1, p_2, \ldots, p_{n-1}\}$ is the set of all inter-positions of u. These are all candidates for handles of cubic runs from $\mathcal{CR}(u)$.

Recall also the definition of the handle function H. We have observed that the maximal number of cubic runs would be obtained when there are $\frac{n-1}{2}$ cubic runs, and H assigns to each of them exactly two handles.

Some cubic runs can have more than two handles. Some inter-positions can be not a handle of any cubic runs, such inter-positions are called here *free* inter-positions. The key to the improvement of the upper bound is the localizations of free inter-positions and cubic runs with more than two handles.

Denote:

$$Y \;=\; \{\, 0,\; 01,\; 0001,\; 0111,\; 000111,\; 1,\; 10,\; 1000,\; 1110,\; 111000 \,\}\;.$$

By an *internal factor* of a word w we mean any factor of w having an occurrence which is neither a prefix nor a suffix of w. An internal factor can also have an occurrence at the beginning or at the end of w. For example, ab is an internal factor of $ababa$, but not of $abab$.

Let X be the set of binary words w which satisfy at least one of the properties:

(1) w has an internal factor which is a non-cubic run containing a square of a word from Y

(2) w has a factor which is a cube of a word from $Y \setminus \{0,\ 1\}$

(3) w has a factor 0000 or 1111.

The words $x \in X$ have several useful properties. For example, if $x =$ 110001000101 then the center of the square 00010001 is a free inter-position in x, since it could only be a handle of a cubic run with period 4, but the run with period 4 containing this square is not cubic. The word 1000100010 is a non-cubic run which is an internal factor of x.

On the other hand, if x contains a factor 000100010001 then it implies a cubic run with 3 handles — the centers of the squares 00010001 and 10001000 (0001 is the minimal rotation and 1000 is the maximal rotation of the period of the run).

The words in X can be checked to satisfy the following simple fact.

Observation 14. *Let $u \in \{0,1\}^n$.*

(a) If a factor $u[i\,..\,j]$ contains any factor satisfying point (1) of the definition of X then there is at least one free inter-position in u amongst $p_i, p_{i+1}, \ldots, p_{j-1}$.

(b) If a factor $u[i\,..\,j]$ contains any factor satisfying point (2) or (3) then there are at least 3 inter-positions in u amongst $p_i, p_{i+1}, \ldots, p_{j-1}$ which are handles of the same cubic run.

This implies the following result.

Theorem 15 (Improved Upper Bound).

$$\textsf{\textit{cubic-runs}}_2(n) \le 0.48\, n\ .$$

Proof. Each binary word of length 25 contains a factor from X. It has been shown experimentally by checking all binary words of size 25.

Let $u \in \{0,1\}^n$. Let us partition the word u into factors of length 25: $u[1\,..\,25], u[26\,..\,50], \ldots$ (possibly discarding at most 24 last letters of u). By Observation 14, it is possible to remove one inter-position from every one of these factors so that each cubic run in u has at least two handles in the set of remaining inter-positions.

The total number of inter-positions in u is $n-1$ and we have shown that at least $\left\lfloor \frac{n-1}{25} \right\rfloor$ of them can be removed and each cubic run will have at least two handles among remaining inter-positions. Hence:

$$\begin{aligned}
\textsf{cubic-runs}(u) &\le \frac{1}{2}\cdot\left(n-1-\left\lfloor\frac{n-1}{25}\right\rfloor\right)\\
&= \frac{1}{2}\cdot\left(\frac{24\cdot(n-1)}{25}+\frac{n-1}{25}-\left\lfloor\frac{n-1}{25}\right\rfloor\right)\\
&\le \frac{1}{2}\cdot\left(\frac{24\cdot(n-1)}{25}+\frac{24}{25}\right) \;=\; 0.48\,n\ .
\end{aligned}$$

This completes the proof. □

[1] P. Baturo, M. Piatkowski, and W. Rytter. The number of runs in Sturmian words. In O. H. Ibarra and B. Ravikumar, editors, *CIAA*, volume 5148 of *Lecture Notes in Computer Science*, pages 252–261. Springer, 2008.

[2] J. Berstel and J. Karhumäki. Combinatorics on words: a tutorial. *Bulletin of the EATCS*, 79:178–228, 2003.

[3] M. Crochemore and L. Ilie. Analysis of maximal repetitions in strings. In L. Kucera and A. Kucera, editors, *MFCS*, volume 4708 of *Lecture Notes in Computer Science*, pages 465–476. Springer, 2007.

[4] M. Crochemore and L. Ilie. Maximal repetitions in strings. *J. Comput. Syst. Sci.*, 74(5):796–807, 2008.

[5] M. Crochemore, L. Ilie, and W. Rytter. Repetitions in strings: Algorithms and combinatorics. *Theor. Comput. Sci.*, 410(50):5227–5235, 2009.

[6] M. Crochemore, L. Ilie, and L. Tinta. Towards a solution to the ”runs” conjecture. In P. Ferragina and G. M. Landau, editors, *CPM*, volume 5029 of *Lecture Notes in Computer Science*, pages 290–302. Springer, 2008.

[7] M. Crochemore, C. S. Iliopoulos, M. Kubica, J. Radoszewski, W. Rytter, and T. Waleń. On the maximal number of cubic runs in a string. In A. H. Dediu, H. Fernau, and C. Martín-Vide, editors, *LATA*, volume 6031 of *Lecture Notes in Computer Science*, pages 227–238. Springer, 2010.

[8] M. Crochemore and W. Rytter. Squares, cubes, and time-space efficient string searching. *Algorithmica*, 13(5):405–425, 1995.

[9] F. Franek and Q. Yang. An asymptotic lower bound for the maximal number of runs in a string. *Int. J. Found. Comput. Sci.*, 19(1):195–203, 2008.

[10] M. Giraud. Not so many runs in strings. In C. Martín-Vide, F. Otto, and H. Fernau, editors, *LATA*, volume 5196 of *Lecture Notes in Computer Science*, pages 232–239. Springer, 2008.

[11] R. M. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *Proceedings of the 40th Symposium on Foundations of Computer Science*, pages 596–604, 1999.

[12] M. Kubica, J. Radoszewski, W. Rytter, and T. Waleń. On the maximal number of cubic subwords in a string. In J. Fiala, J. Kratochvíl, and M. Miller, editors, *IWOCA*, volume 5874 of *Lecture Notes in Computer Science*, pages 345–355. Springer, 2009.

[13] K. Kusano, W. Matsubara, A. Ishino, H. Bannai, and A. Shinohara. New lower bounds for the maximum number of runs in a string. *CoRR*, abs/0804.1214, 2008.

[14] M. Lothaire. *Combinatorics on Words*. Addison-Wesley, Reading, MA., U.S.A., 1983.

[15] F. Mignosi and G. Pirillo. Repetitions in the Fibonacci infinite word. *ITA*, 26:199–204, 1992.

[16] S. J. Puglisi, J. Simpson, and W. F. Smyth. How many runs can a string contain? *Theor. Comput. Sci.*, 401(1-3):165–171, 2008.

[17] W. Rytter. The number of runs in a string: Improved analysis of the linear upper bound. In B. Durand and W. Thomas, editors, *STACS*, volume 3884 of *Lecture Notes in Computer Science*, pages 184–195. Springer, 2006.

[18] W. Rytter. The structure of subword graphs and suffix trees in Fibonacci words. *Theor. Comput. Sci.*, 363(2):211–223, 2006.

[19] W. Rytter. The number of runs in a string. *Inf. Comput.*, 205(9):1459–1469, 2007.

[20] J. Simpson. Modified Padovan words and the maximum number of runs in a word. *Australasian J. of Comb.*, 46:129–145, 2010.

Maxime Crochemore, Professor at King's College London and Professor Emeritus at University Paris-Est Marne-la-Vallée. He was involved in the creation of the University of Marne-la-Vallée, where he created the Computer Science research laboratory in 1991 and was the director until 2005. Prof. Crochemore's research interests are in the design and analysis of algorithms. His major achievements are on string algorithms, which includes pattern matching, text indexing, coding, and text compression. He also works on the combinatorial background of these subjects and on their applications to bioinformatics. He has co-authored several textbooks on algorithms and published more than 200 articles.

Costas Iliopoulos, Professor at King's College London and Head of the Bioinformatics and Algorithm Design Group, and Professor at Curtin University of Technology, Perth, Australia. An author/co-author of more than 100 publications. His research interests focus on the design and analysis of string algorithms, algorithms for music analysis and for biological sequences, data compression and compressed matching.

Marcin Kubica, Ph.D. in computer science, assistant professor at Institute of Informatics, Department of Mathematics, Informatics and Mechanics, University of Warsaw, lecturer at Polish-Japanese Institute of Information Technology, and scientific secretary of Polish Olympiad in Informatics. His research interests include combinatorial and text algorithms, with special focus on repetitions and periodicities, and computational biology.

Jakub Radoszewski, Ph.D. student at Department of Mathematics, Informatics and Mechanics, University of Warsaw, and chair of the Jury of Polish Olympiad in Informatics. His research interests focus on text algorithms and combinatorics, and discrete mathematics.

Wojciech Rytter, Professor at University of Warsaw, Poland, Head of the Section on Analysis of Algorithms in the Department of Mathematics, Informatics and Mechanics, University of Warsaw. Prof. Rytter is an author/co-author of more than 100 publications and a co-author of several textbooks on algorithms. His research interests focus on the design and analysis of algorithms and data structures, parallel computations, discrete mathematics, graph theory, algorithms on texts, automata theory and formal languages. Main current interests: text algorithms, algorithms for highly compressed objects (without decompression), automata and formal languages.

Tomasz Waleń, Ph.D. in computer science, assistant professor at Institute of Informatics, Department of Mathematics, Informatics and Mechanics, University of Warsaw. His research interests focus on combinatorics and text algorithms, and on their applications to bioinformatics.