Number of occurrences of powers in strings

Maxime Crochemore, Szilard Zsolt Fazekas, Costas S. Iliopoulos, Inuka Jayasekera

To cite this version:

Maxime Crochemore, Szilard Zsolt Fazekas, Costas S. Iliopoulos, Inuka Jayasekera. Number of occurrences of powers in strings. International Journal of Foundations of Computer Science, World Scientific Publishing, 2010, 21 (4), pp.535–547. <10.1142/S0129054110007416>. <hal-00693448>

HAL Id: hal-00693448

https://hal-upec-upem.archives-ouvertes.fr/hal-00693448

Submitted on 13 Feb 2013

Number of occurrences of powers in strings

Maxime Crochemore[1,2], Szilárd Zsolt Fazekas[3*],
Costas Iliopoulos[1], Inuka Jayasekera[1**]

[1] King's College London, U.K.
[2] Université Paris-Est, France
[3] Rovira i Virgili University, Tarragona, Spain

Abstract. We show a $\Theta(n \log n)$ bound on the maximal number of occurrences of primitively-rooted k-th powers occurring in a string of length n for any integer k, $k \geq 2$. We also show a $\Theta(n^2)$ bound on the maximal number of primitively-rooted powers with fractional exponent e, $1 < e < 2$, occurring in a string of length n. This result holds obviously for their maximal number of occurrences. The first result contrasts with the linear number of occurrences of maximal repetitions of exponent at least 2.

1 Introduction

The subject of this paper is the evaluation of the number of powers in strings. This is one of the most fundamental topics in combinatorics on words not only for its own combinatorial aspects considered since the beginning of last century by the precursor A. Thue [18], but also because it is related to lossless text compression, string representation, and analysis of molecular biological sequences, to quote a few applications. These applications often require fast algorithms to locate repetitions because either the amount of data to be treated is huge or their flow is to be analysed on the fly, but their design and complexity analysis depends on the type of repetitions considered and on their bounds.

A repetition is a string composed of the concatenation of several copies of another string whose length is called a period. The exponent of a string is informally the number of copies and is defined as the ratio between the length of the string and its smallest period. This means that the repeated string, called the root, is primitive (it is not itself a nontrivial integer power). We consider two types of strings: integer powers—those having an integer exponent at least 2, and fractional powers—those having a fractional exponent between 1 and 2. For both of them we consider their maximal number in a given string as well as their maximal number of occurrences.

It is known that all the occurrences of integer powers in a string of length n can be computed in time $O(n \log n)$ (see three different methods in [2], [1], and

[*] Supported by grant no. AP2004-6969 from the Spanish Ministry of Science and Education of Spain. Partially supported by grant. no. MTM 63422 from the Ministry of Science and Education of Spain.

[**] Supported by a DTA Award from EPSRC

[14]). Indeed these algorithms are optimal because the number of occurrences of squares (powers of exponent 2) can be of the order of $n \log n$ [2].

The computation of occurrences of fractional powers with exponent at least 2 has been designed initially by Main [13] who restricted the question to the detection of their leftmost maximal occurrences only. Eventually the notion of runs—maximal occurrences of fractional powers with exponent at least 2—introduced by Iliopoulos et al. [8] for Fibonacci words, led to a linear-time algorithm for locating all of them on a fixed-sized alphabet. The algorithm, by Kolpakov and Kucherov [9, 10], is an extension of Main's algorithm but their fundamental contribution is the linear number of runs in a string. They proved that the number of runs in a string of length n is at most cn, could not provide any value for the constant c, but conjectured that $c = 1$. Rytter [16] proved that $c \leq 5$, then $c \leq 3.44$ in [17], Puglisi et al. [15] that $c \leq 3.48$, Crochemore and Ilie [3] that $c \leq 1.6$, and Giraud [7] that $c \leq 1.5$. The best value computed so far is $c = 1.029$ [4] (see the Web page `http://www.csd.uwo.ca/ ilie/runs.html`). Franek et al. showed a lower bound of $0.927...n$ in [6], which was improved to $0.944565n$ by Matsubara et al. in [11] and to 0.944575n by Puglisi and Simpson. (see the Web page `http://www.shino.ecei.tohoku.ac.jp/runs/`). These lower bounds also point in the direction of Kolpakov and Kucherov's conjecture.

Runs capture all the repetitions in a string but without discriminating among them according to their exponent. For example, the number of runs is not easily related to the number of occurrences of squares. This is why we consider an orthogonal approach here. We count and bound the maximal number of repetitions having a fixed exponent, either an integer larger than 1 or a fractional number between 1 and 2. We also bound the number of occurrences of these repetitions.

After introducing the notations and basic definitions in the next section, Section 3 deals with fractional powers with exponent between 1 and 2. It is shown that the maximum number of primitively-rooted powers with a given exponent e, $1 < e < 2$, in a string can be quadratic as well of course as their maximum number of occurrences. In Section 4, we consider primitively-rooted integer powers and show that the maximum number of occurrences of powers of a given exponent k, $k \geq 2$, is $\Theta(n \log n)$. This latter result contrasts with the linear number of such powers. We also present an efficient algorithm for constructing the strings in question.

2 Preliminaries

In this section we introduce the notation and recall some basic results that will be used throughout the paper. All results stated in this section are reported from [12]. An *alphabet* A is a finite non-empty set. We call the elements of A *letters*. The set of all finite words over A is A^*, which is a monoid with concatenation (juxtaposition), where the unit element is ϵ, the *empty word*, whereas the set of non-empty words is $A^+ = A^* - \epsilon$. The length of a word w is denoted by $|w|$; $|\epsilon| = 0$. Without loss of generality, we can assume that our alphabet is ordered and hence we have an ordering on words. The one we will use is called the *lexicographical order* and is defined by the following relation:

$$x \leq y \Leftrightarrow (x is a prefix of y \text{ or } (x = uav and y = ubw and a < b))$$

where $a, b \in A$ and $u, v, w \in A^*$.

For words $u, v, w \in A^*$, with $w = uv$, we say that u is a *prefix* and v is a *suffix* of w. For a word w and an integer $n \geq 0$, the n-th power of w is defined inductively as $w^0 = \epsilon$, $w^n = ww^{n-1}$. Extending this definition we can talk about non-integer powers too. Take $n = \frac{k}{l} > 1$ with $gcd(k, l) = 1$. We say that a word w is an n-power if both of the following conditions apply:

- $|w| = m \cdot k$ for some integer $m > 0$,
- $m \cdot l$ is a period of w.

The prefix of length $m \cdot l$ of w is a root of w.
When $w \neq \epsilon$, w^3 is called a *cube*, with *root* w. A word w is called *primitive* if there is no word u and integer $p \geq 2$ such that $w = u^p$. We say that w' is a *conjugate* of w if there exist $u, v \in A^*$ such that $w = uv$ and $w' = vu$. A *Lyndon word* is a (primitive) word which is the lexicographically smallest among its conjugates.

Let uv be a primitive word such that vu forms a Lyndon word and v is nonempty. In the cube $(uv)^3$, we call *central Lyndon* position the position $|uvu|$ For two non-empty words u and v it is known that $uv = vu$ implies $u, v \in z^+$ for some $z \in A^*$, therefore every word has a unique Lyndon position.

If a word w can be written as $w = uv = vz$, for some words $u, v, z \in A^+$, then we say that w is *bordered* (v is a border of w). If a word w is bordered, then there exists $u \in A^+, v \in A^*$ such that $w = uvu$, that is, a bordered word w always has a border of length at most half the length of w. Moreover, it is easy to see that a bordered word uvu cannot be a Lyndon word, because then either uuv (if $u < v$) or vuu (if $v < u$) is lexicographically smaller than uvu.

3 A bound on repeats with exponent e, $1 < e < 2$

In this section, we show that the maximal number of distinct repetitions with exponent e, $1 < e < 2$, is lower bounded by $\Theta(n^2)$. We do this by looking at the number of such repetitions that can start at a position in words of the form $a^k ba^{\frac{k}{e-1}-1}$, where k is any positive integer such that $c|k$, where $e = \frac{c+d}{d}$ and $gcd(c + d, d) = 1$.
First we consider an example with $e = \frac{3}{2}$ and $k = 9$, i.e. $w = a^9ba^{17}$ (see Fig 1). At the first position in this word, we can have 5 repetitions of exponent $\frac{3}{2}$, namely $a^9ba^5, a^9ba^8, a^9ba^{11}, a^9ba^{14}$ and a^9ba^{17}. Moving on to the second position, we have only 4 repetitions of exponent $\frac{3}{2}$, namely $a^8ba^6, a^8ba^9, a^8ba^{12}$ and a^8ba^{15}. In the third position also, we have the repetitions a^7ba^7, a^7ba^{10} and a^7ba^{13}. However, now we have one extra repetition as we can also have a^7ba^4. It is clear that at every other position in the word, as we get closer to the occurrence of b, we have an extra repetition. The numbers of primitively-rooted repetitions of

exponent $\frac{3}{2}$ at each position are $5, 4, 4, 3, 3, 2, 2, 1, 1$ (see Fig. 1). The total number of repetitions can now be summed up to $((5 * 6)/2) + (((5 - 1) * 5)/2) = 25$. We generalise this example in the next theorem.

Theorem 1. *The maximal number of distinct repetitions of exponent e, with $1 < e < 2$, in a word of length n is $\Theta(n^2)$.*

Proof. The upper bound is trivial because no factor of the string can be counted twice as an e-power for given e, so let us turn to proving the lower bound.
We shall count the number of repetitions starting at each position in a word. For an exponent e, $1 < e < 2$, we consider a word, w, formed as shown in Fig. 2. Here, we concatenate a repetition of exponent, e, with root $a^k b$ and $a^{\frac{k}{e-1}-1}$, where k is any positive integer such that $c|k$, where $e = \frac{c+d}{d}$ and $gcd(c+d, d) = 1$.
$(e - 1)|k$.
length of our string is $k \cdot \frac{e}{e-1}$.

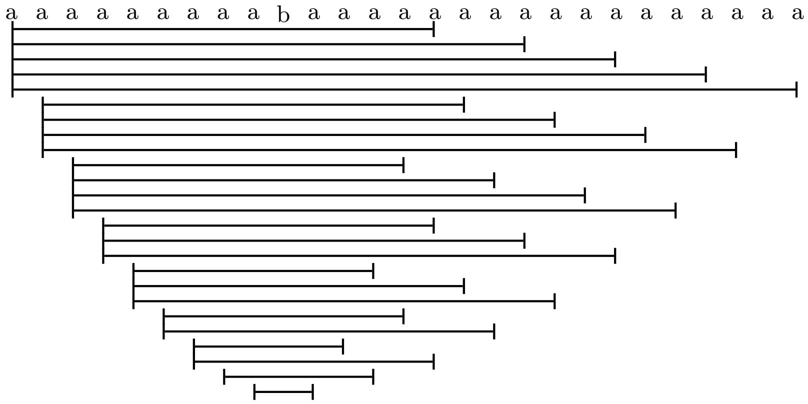


Fig. 1. Repetitions of exponent 1.5 in a^9ba^{17}

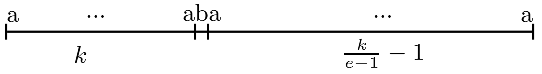


Fig. 2. Structure of word, w

For e-powers starting at the first position, the end positions can be $(k+1)(e-1)$, $(k+1)(e-1)+(c+d)$, $(k+1)(e-1)+2\cdot(c+d)$, etc.
From here we get that the number of e-powers starting at the first position is

$$\frac{|w| - (k+1)(e-1)}{c+d} + 1 = \frac{k \cdot \frac{e}{e-1} - (k+1)(e-1)}{c+d} + 1$$

Substituting $\frac{c+d}{d}$ for e in the formula above we get that the number of e-powers starting at the first position is:

$$k \cdot \frac{d-c}{d \cdot c} - \frac{1}{d} + 1$$

This formula proves useful because by substituting $k - i$ for k and taking the integer part of the result (since we are talking about the number of occurrences) we get the number of e-powers starting at position $i + 1$. Now let us sum up the number of e-power occurrences starting at any one of the first k positions:

$$\sum_{i=1}^{k} \lfloor i \cdot \frac{d-c}{d \cdot c} - \frac{1}{d} + 1 \rfloor$$

For any positive n its integer part $\lfloor n \rfloor$ is greater or equal than $n - 1$. As we are trying to give a lower bound to the number of occurrences, it is alright to subtract 1 from the formula instead of taking its integer part:

$$\sum_{i=1}^{k} \left(i \cdot \frac{d-c}{d \cdot c} - \frac{1}{d} \right) = k \cdot (k+1) \cdot \frac{d-c}{2d \cdot c} - \frac{k}{d}$$

This means that the number of e-powers in our string is quadratic in k. At the same time the length of the string, as we mentioned in the beginning, is $k \cdot \frac{e}{e-1}$, so for a given e, the number of e-powers in a string of length n is $\Theta(n^2)$.
It is easy to see that every occurrence of an e-power in this string is unique and this concludes the proof. □

4 A bound on primitively-rooted cubes

After considering powers between 1 and 2, we have look at powers greater than 2. First, we show that it is possible to construct strings of length n, which have $\Omega(n \log n)$ occurrences of cubes. We can extend the method to all integer powers greater than 2, and this, together with the $O(n \log n)$ upper bound implied by the number of squares (see [2]) leads us to the $\Theta(n \log n)$ bound. Finally, we will prove that the sum of all occurrences of powers at least 2 (including non-integer exponents) is quadratic.

Lemma 1. *The maximal number of primitively-rooted cubes in a word of length n is $\Theta(n \log n)$.*

Proof. Let us suppose there are two primitively-rooted cubes $(uv)^3$ and $(xy)^3$ in w such that their central Lyndon positions $uvu.vuv$ and $xyx.yxy$ are the same. First let us look at the case where the cubes have to be of different length. Without loss of generality we can assume $|uv| < |xy|$. In this case vu is at the same time a prefix and suffix of yx. Hence, yx is bordered and cannot be a Lyndon word contradicting the assumption that $x.y$ is a Lyndon position.

This proves that should there be more cubes which have their central Lyndon position identical, they all have to be of the same length. Naturally, the first and last position of a word cannot be central Lyndon to any cube and this gives us the bound $n-2$ if we disregard cubes of the same length which have their central Lyndon positions at the same place (see Fig. 3). It is easy to see, that because of the periodicity theorem the only string of length n, for which $n-2$ different positions are central Lyndon ones to some cube, is a^n.

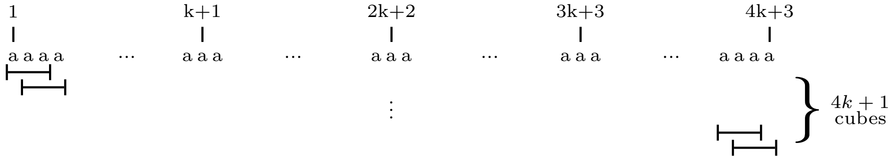


Fig. 3. Cubes of word a^{4k+3}

Now take the word a^{4k+3}. According to our previous argument it has at most $4k+1$ cubes. However, if we change a's into b's at positions $k+1$,$2k+2$ and $3k+3$ we get that the number of primitively-rooted cubes in this word is $4k+1-9+(k+1) = 5k-7$. This is because by introducing each b we lose three cubes but in the end we gain another $k+1$ cubes of the form $(a^j ba^{k-j})^3$ with $0 \leq j \leq k$ (see Fig. 4). Note that these latter cubes all have their central Lyndon position after the first b (assuming $a < b$).

We introduced three b's in the previous step but of course we can repeat the procedure for the four block of a's delimited by these b's and then in turn for the new, smaller blocks of a's that result and so on. In the second step, however, we need to introduce 12 b's - that is, 3 for each of the 4 blocks of a's - not to disrupt the cubes of length $3k+3$. This way we lose $12 \cdot 3 = 36$ cubes and we gain $(\lfloor (k-3)/4 \rfloor + 1) \cdot 4$ new ones. Performing the introduction of b's until the number of cubes we lose in a step becomes greater or equal to the ones we gain, gives us a string with the maximal possible number of cubes for its length. If k

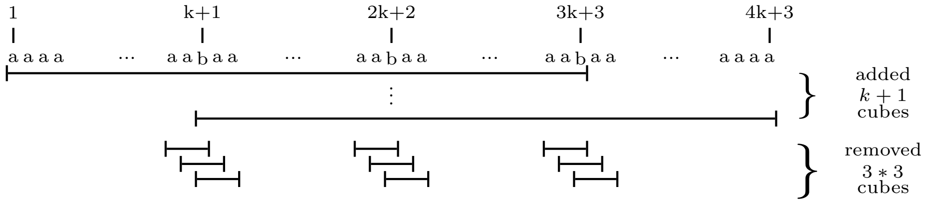

1
k+1
2k+2
3k+3
4k+3
a a a a ··· a a b a a ··· a a b a a ··· a a b a a ··· a a a a
added $k+1$ cubes
removed $3*3$ cubes

Fig. 4. Cubes of word $a^k ba^k ba^k ba^k$

equals $4j$, $4j+1$ or $4j+2$ for some j then according to the formula above the number of cubes we gain is $4j$. Note that if $k = 4j+3$ than the number of cubes we gain in the second step is $4j+4 = k+1$, i.e. the same as in the first step. However, together with the delimiting b's introduced before we would get a big cube which is not primitively-rooted anymore, so we need to move the newly introduced b's 1, 2 and 3 positions to the left, respectively. This gives us that in this case too the number of newly formed cubes will be $4j$. The smallest length at which introducing the b's does not induce less cubes is 35 that is with $k = 8$. Summarizing the points above we get that for a string of length n the maximum increase in the number of cubes for the ith ($i > 1$) consecutive application of our procedure is:

$$\frac{(n-3)}{4} - 9 \cdot 4^{i-1}$$

To be able to sum these increases we have to know the number of steps performed. This is given by solving for i the equation:

$$\frac{n-3}{4} = 9 \cdot 4^{i-1}$$

From here we get that the number of steps performed is $\#steps = \lfloor \log_4 \frac{n-3}{9} \rfloor$, where by $\lfloor x \rfloor$ we mean the integer part of x.
Hence the number of cubes for length $n \geq 39$ is:

$$n - 2 + 1 + \sum_{i=1}^{\#steps} \left(\frac{n-3}{4} - 9 \cdot 4^{i-1} \right)$$

$$= n - 1 + \frac{(n-3)\lfloor \log_4 \frac{(n-3)}{9} \rfloor}{4} - \frac{9(1 - 4^{\lfloor \log_4 \frac{n-3}{9} \rfloor})}{-3}$$

$$= n + 2 + \frac{(n-3)\lfloor \log_4 \frac{(n-3)}{9} \rfloor}{4} - 3 \cdot 4^{\lfloor \log_4 \frac{n-3}{9} \rfloor}$$

The plus one after $n-2$ comes from the first application of the insertion of b's where we get $(n-3)/4 + 1$ cubes instead of $(n-3)/4$. For strings shorter than 39 therefore the count is one less. □

Since the first paragraph of the proof is valid for any integer power, we can extend the proof by giving the construction of the strings that prove the lower bound in general for a string of length n and power k (see Fig. 4).

The algorithm above produces strings which have $O(n \log n)$ occurrences of k-th powers. Note, that if we perform the procedure the other way around, we only need $O(\log n)$ cycles and we can eliminate the recursion:

Theorem 2. *Algorithm ConstructStrings2 (see Fig. 4) produces a string of length n that has $\Omega(n \log n)$ occurrences of primitively-rooted cubes.*

```
Algorithm ConstructStrings1$(n, k)$
Input: $n \geq 0,\ k \geq 0$
Output: A string which proves the lower bound of the number of occurrences of integer
    powers.
$\ell = n$
string $= a^{\ell}$
power$(1, \ell)$
Procedure: power( start, end)
$\ell =$ end - start
if $\ell < k^3 + k^2 + k$
  then return
  else string[start $+ \lfloor \ell/(k+1) \rfloor$] $= b$
       string[start $+ 2 \cdot \lfloor \ell/(k+1) \rfloor$] $= b$
       $\ldots$
       string[start $+ k \cdot \lfloor ell/(k+1) \rfloor$] $= b$
       for $i \leftarrow 0$ to $k$
      power(start $+ i \cdot \ell/(k+1)$, start $+ (i+1) \cdot \ell/(k+1)$)
```

```
Algorithm ConstructStrings2$(n, k)$
Input: $n \geq 0,\ k \geq 0$
Output: A string which proves the lower bound of the number of occurrences of integer
    powers.
$\ell = n$
while $\ell \geq k^3 + k^2 + 3k + 2$
   do $\ell = \frac{\ell - k}{k+1}$
$string = (a^{k^2+1} + b)^k + a^{(k+1) \cdot \ell - k^3 - k}$
$delimiter = b$
while $length(string) * (k+1) + k < n$
   do $string = (string + delimiter)^k + string$
      if $delimiter = b$
         then $delimiter = a$
         else $delimiter = b$
      (* changing the delimiter is needed to stay primitive *)
$string = string + a^{n - length(string)}$
```

Proof. Before entering the second **while** loop, the length of *string* and the number of k-th power occurrences in it are both $c = (k+1) \cdot \ell + k$. Now we will show by induction on i that after the i-th iteration of the second **while** loop the length of *string* will be $(k+1)^i \cdot (c+1) - 1$ and the number of occurrences of k-th powers will be $(k+1)^i \cdot c + i \cdot (k+1)^{i-1}(c+1)$.
Note that if the length of *string* was m and the number of k-th power occurrences was p after the previous cycle, then concatenating $k+1$ copies of *string* delimited by k copies of *delimiter* we get $(k+1) \cdot p + m + 1$ powers in the new *string*, which will have length $(k+1) \cdot m + k$. Therefore, after the first cycle the length of *string* will be

$$(k+1) \cdot c + k = (k+1) \cdot c + (k+1) - 1 = (k+1)^1 \cdot (c+1) - 1$$

At the same time the number of k-th powers will be

$$(k+1) \cdot c + c + 1 = (k+1)^1 \cdot c + 1 \cdot (k+1)^0 \cdot (c+1)$$

so our statement holds for $i = 1$. Now suppose it is true for some $i \geq 1$. From here we get that for $i+1$ the length of *string* will be:

$$(k+1) \cdot ((k+1)^i \cdot (c+1) - 1) + k = (k+1)^{i+1} \cdot (c+1) - 1$$

whereas the number of k-th powers is:

$$\begin{aligned}(k+1) \cdot ((k+1)^i \cdot c + i \cdot (k+1)^{i-1} \cdot (c+1)) + ((k+1)^i \cdot (c+1) - 1) + 1 \\ = (k+1)^{i+1} \cdot c + i \cdot (k+1)^i \cdot (c+1) + (k+1)^i \cdot (c+1) \\ = (k+1)^{i+1} \cdot c + (i+1) \cdot (k+1)^i (c+1)\end{aligned}$$

Now let us look at the running time of the algorithm. In the first **while** loop we divide the actual length by $k+1$ and we do it until it becomes smaller than $k^3 + k^2 + k$ therefore we perform $O(\log n)$ cycles. The second **while** loop has the same number of cycles, with one string concatenation performed in each cycle, hence substituting $\log n$ for i in the formula above concludes the proof.

□

Corollary 1. *In a string of length n the maximal number of primitively-rooted k-th powers, for a given integer $k \geq 2$, is $\Theta(n \log n)$.*

Proof. We know from [5] that the maximal number of occurrences of primitively-rooted squares in a word of length n is $O(n \log n)$. This implies that the number of primitively-rooted greater integer powers also have an $O(n \log n)$ upper bound, while in Theorem 2 we showed the lower bound $\Omega(n \log n)$. □

Remark 1. The first part of the proof is directly applicable to runs so we have that in a string of length n the number of runs of length at least $3p - 1$, where p is the (smallest) period of the run is at most $n - 2$. Unfortunately we cannot apply the proof directly for runs shorter than that because we need the same string on both sides of the central Lyndon position.

We have seen that the number of k-th powers for a given k(≥ 2) in a string of length n is $\Theta(n \log n)$, but what happens if we sum up the occurrences of k-th powers for all $k \geq 2$?

Remark 2. The upper bound of the sum of all occurrences of k-th powers with primitive root, where $k \geq 2$, in a word w with $|w| = n$ is $\frac{n \cdot (n-1)}{2}$. Moreover, the bound is sharp.

Proof. First consider the word a^n, for some $n > 0$. Clearly, taking any substring a^k, with $2 \leq k \leq n$, we get a k-th power, so the number of powers greater or equal to two is given by the number of contiguous substrings of length at least two, that is $\frac{n \cdot (n-1)}{2}$. Now we will show that this is the upper bound. Let us suppose that any two positions i and j in the string delimit a k-th power with $k \geq 2$, just like in the example above. We need to prove that the same string cannot be considered a k_1-th power and a k_2-th power at the same time, with $k_1, k_2 \geq 2$ and $k_1 \neq k_2$. Suppose the contrary, that is there are $1 \leq m < \ell \leq \frac{j-i}{2}$ so that both m and ℓ are periods of $w[i, j]$. Since $j - i > m + \ell - gcd(m, \ell)$ the periodicity lemma tells us that $w[i, j]$ has a period p smaller than m with $p|m$ and $p|\ell$, and this, in turn, means $w[i, i + \ell]$ is not primitive. □

Theorem 3. *The number of distinct k-th powers, for a fixed integer $k \geq 3$, in a string of length n is at most $\frac{n}{k-2}$.*

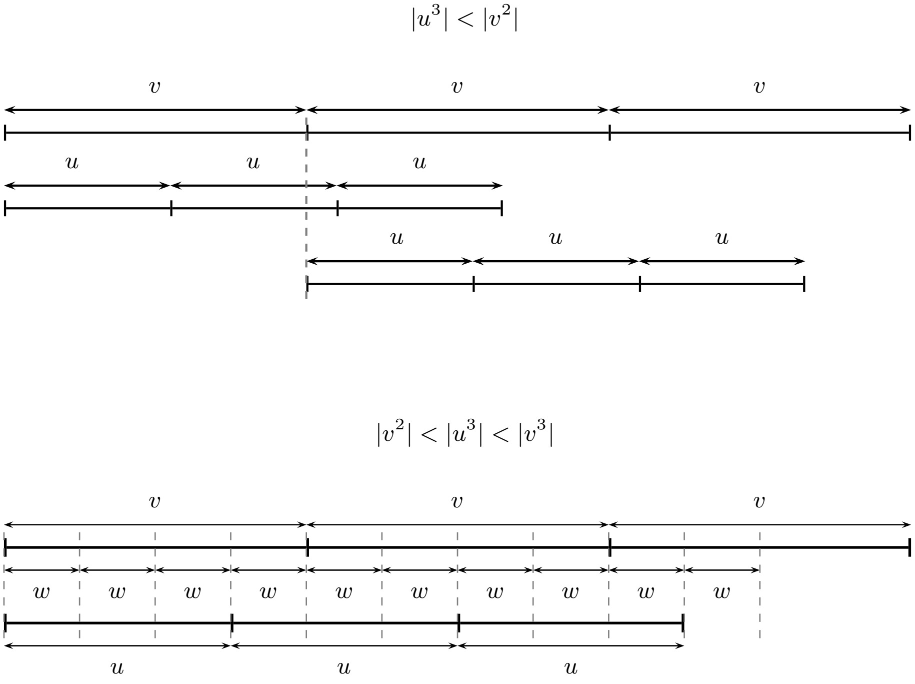
Proof. We will show the upper bound by considering the last occurrences of every k-th power. The proof is split into two parts. We will prove the statement for cubes by considering their starting positions while for higher exponents we will look at their root positions (see below).

Let us start with the case $k = 3$. Suppose the last occurrence of two different cubes u^3 and v^3 with $|u| < |v|$ start at the same position i in the string. By a simple argument we will arrive at a contradiction by looking at the two cases shown in Figure 5.

First let us look at the case when $|u^3| \leq |v^2|$. In this case there is another occurrence of u^3 starting at position $i + |v|$ contradicting our assumption of the previous occurrence being the last.

Now we are left to treat the case when $|v^2| < |u^3| < |v^3|$. The overlap between the two cubes in this case is at least $2 \cdot |v|$ which is greater than $|v| + |u|$ and from this, Fine and Wilf's theorem tells us it has a period of length $gcd(|u|, |v|)$. Therefore, there exists some w such that $u = w^m$ and $v = w^n$, for some integers $m < n$. It is easy to see then that in $|w^{3n}|$ the last occurrence of $|w^{3m}|$ starts at position $i + 3 \cdot (m - n) \cdot |w|$ contradicting our assumption again.

We showed that there can be no two different cubes which have their last occurrence starting at the same position. This implies the bound n for higher distinct powers as well. However, we can prove something stronger, as we claim in the theorem. To achieve that result, we will look at $r$oot positions. In a power u^k starting at position i, with the smallest period of u being p, we will call position $i + p$ the second root position, $i + 2p$ the third root position and so on. We show that for the last occurrences of two 4-th powers u^4 and v^4, with $|u| < |v|$, u^4



$|u^3| < |v^2|$

$|v^2| < |u^3| < |v^3|$

Fig. 5. Cubes u^3 and v^3 beginning at the same position i.

starting at position i and v^4 starting at position j, the following positions cannot coincide:

1. the second root position of u and the second root position of v:
   – if $3|u| < 2|v|$ then u^4 occurs at $i + |v|$, contradiction;
   – if $2|v| \leq 3|u|$ then according to Fine and Wilf's theorem v has period $k \cdot p$, for some k and then u^4 occurs at $i + p$, contradiction.
2. the second root position of u and the third root position of v:
   – if $3|u| \leq |v|$ then u^4 occurs at $i + |v|$, contradiction;
   – if $|v| < 3|u| \leq 2|v|$ then again Fine and Wilf's theorem gives us u^4 occurring at $i + p$, contradiction;
   – if $2|v| < 3|u|$: if this is the case then similarly as before u and v are powers of the same word and hence v^4 occurs at $j + p$, contradiction;
3. the second root position of v and the third root position of u:
   – if $3|u| < |v|$ then u^4 occurs at $i + |v|$, contradiction;
   – if $|v| \leq 3|u|$ then u^4 occurs at $i + p$, contradiction.
4. the third root position of u and the third root position of v:
   – if $2|u| \leq |v|$ then u^4 occurs at $i + |v|$, contradiction;
   – if $|v| < 2|u|$ then u^4 occurs at $i + p$.

We can apply the same argument for 5-th powers looking at the second, third and fourth root positions and so on for greater powers as well, getting the desired bound.

5 Conclusion

In conclusion, we have proven the following bounds on repetitions in words:

(i) The maximal number of distinct repetitions of exponent, e, with $1 < e < 2$, in a word of length n is $\Theta(n^2)$.
(ii) The maximal number of primitively-rooted k-th powers in a word of length n is $\Omega(n \log n)$.

We have also described an $O(m \log n)$ algorithm which can be used to construct strings to illustrate these bounds. Here $O(m)$ is the time complexity of concatenating two strings of length n.

References

1. A. Apostolico and F. P. Preparata. Optimal off-line detection of repetitions in a string. *Theoret. Comput. Sci.*, 22(3):297–315, 1983.
2. M. Crochemore. An optimal algorithm for computing the repetitions in a word. *Inf. Process. Lett.*, 12(5):244–250, 1981.
3. M. Crochemore and L. Ilie. Maximal repetitions in strings. *J. Comput. Syst. Sci.*, 2007. In press.

4. M. Crochemore, L. Ilie, and L. Tinta. Towards a solution to the "runs" conjecture. In P. Ferragina and G. M. Landau, editors, *Combinatorial Pattern Matching*, LNCS. Springer-Verlag, Berlin, 2008. In press.
5. M. Crochemore and W. Rytter. Squares, cubes and time-space efficient string-searching. *Algorithmica*, 13(5):405–425, 1995.
6. F. Franek, R. J. Simpson, and W. F. Smyth. The maximum number of runs in a string. In M. M. . K. Park, editor, *Proc. 14th Australasian Workshop on Combinatorial Algorithms*, pages 26–35, 2003.
7. M. Giraud. Not so many runs in strings. In C. Martin-Vide, editor, *2nd International Conference on Language and Automata Theory and Applications*, 2008.
8. C. S. Iliopoulos, D. Moore, and W. F. Smyth. A characterization of the squares in a Fibonacci string. *Theoret. Comput. Sci.*, 172(1–2):281–291, 1997.
9. R. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *Proceedings of the 40th IEEE Annual Symposium on Foundations of Computer Science*, pages 596–604, New York, 1999. IEEE Computer Society Press.
10. R. Kolpakov and G. Kucherov. On maximal repetitions in words. *J. Discret. Algorithms*, 1(1):159–186, 2000.
11. 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.
12. M. Lothaire. *Applied Combinatorics on Words*. Cambridge University Press, Cambridge, UK, 2005.
13. M. G. Main. Detecting leftmost maximal periodicities. *Discret. Appl. Math.*, 25:145–153, 1989.
14. M. G. Main and R. J. Lorentz. An $O(n \log n)$ algorithm for finding all repetitions in a string. *J. Algorithms*, 5(3):422–432, 1984.
15. S. J. Puglisi, J. Simpson, and W. F. Smyth. How many runs can a string contain?, 2007. Personal communication, submitted.
16. 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.
17. W. Rytter. The number of runs in a string. *Inf. Comput.*, 205(9):1459–1469, 2007.
18. A. Thue. Über unendliche Zeichenreihen. *Norske Vid. Selsk. Skr. I Math-Nat. Kl.*, 7:1–22, 1906.