Information Processing Letters 125 (2017) 15–19

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

# Finding the largest fixed-density necklace and Lyndon word

Joe Sawada *, Patrick Hartman

ARTICLE INFO

*Article history:*
Received 7 April 2016
Received in revised form 18 April 2017
Accepted 19 April 2017
Available online 2 May 2017
Communicated by R. Uehara

*Keywords:*
Algorithms
Necklace
Lyndon word
Fixed-density
Ranking

ABSTRACT

We present an $O(n)$ time algorithm for finding the lexicographically largest fixed-density necklace of length $n$. Then we determine whether or not a given string can be extended to a fixed-density necklace of length $n$ in $O(n^2)$ time. Finally, we give an $O(n^3)$ algorithm that finds the largest fixed-density necklace of length $n$ that is less than or equal to a given string. The efficiency of the latter algorithm is a key component to allow fixed-density necklaces to be ranked efficiently. The results are extended to find the largest fixed-density Lyndon word of length $n$ (that is less than or equal to a given string) in $O(n^3)$ time.



## 1. Introduction

A *necklace* is the lexicographically smallest string in an equivalence class of strings under rotation. A *Lyndon word* is a primitive (aperiodic) necklace. The *density* of a binary string is the number of 1s it contains. Let $\mathbf{N}(n,d)$ denote the set of all binary necklaces with length $n$ and density $d$. In this paper we present efficient algorithms for the following three problems:

1. finding the largest necklace in $\mathbf{N}(n,d)$,
2. determining if an arbitrary string is a prefix of some necklace in $\mathbf{N}(n,d)$, and
3. finding the largest necklace in $\mathbf{N}(n,d)$ that is less than or equal to a given binary string of length $n$.

The first problem can be answered in $O(n)$ time, which is applied to answer the second problem in $O(n^2)$ time, which in turn is applied to answer the third problem in $O(n^3)$ time. The third problem can also be solved for fixed-density Lyndon words in $O(n^3)$ time, which can immediately be used to find the largest fixed-density Lyndon word of a given length. Solving the third problem efficiently for both necklaces and Lyndon words is a key step in the first efficient algorithms to rank and unrank fixed-density necklaces and Lyndon words [2]. When there is no density constraint, the third problem is known to be solvable in $O(n^2)$-time; one such implementation is outlined in [10]. This problem was encountered in the first efficient algorithms to rank and unrank necklaces and Lyndon words discovered independently by Kopparty, Kumar, and Saks [5] and Kociumaka, Radoszewski and Rytter [4].

To illustrate these problems, consider the lexicographic listing of $\mathbf{N}(8,3)$:

$$00000111, 00001011, 00001101, 00010011,$$

$$00010101, 00011001, 00100101.$$

The largest necklace in this set is 00100101. The string 0010 is a prefix of a necklace in this set, however, 010 is not. Given an arbitrary string $\alpha = 00011000$, the largest necklace in this set that is less than or equal to $\alpha$ is 00010101.

Fixed-density necklaces were first studied by Savage and Wang when they provided the first Gray code listing in [11]. Since then, an algorithm to efficiently list fixed-density necklaces was given by Ruskey and Sawada [8] and another efficient algorithm to list them in cool-lex Gray

* Corresponding author.
*E-mail address:* jsawada@uoguelph.ca (J. Sawada).

http://dx.doi.org/10.1016/j.ipl.2017.04.010
0020-0190/

code order was given by Sawada and Williams [9]. The latter algorithm leads to an efficient algorithm to construct a fixed-density de Bruijn sequence by Ruskey, Sawada, and Williams [7]. When equivalence is further considered under string reversal, an algorithm for listing fixed-density bracelets is given by Karim, Alamgir and Husnine [3].

The remainder of this paper is presented as follows. In Section 2, we present some preliminary results on necklaces and related objects. In Section 3, we present an $O(n)$-time algorithm to find the largest necklace in $\mathbf{N}(n,d)$. In Section 4, we present an $O(n^2)$-time algorithm to determine whether or not a given string is a prefix of a necklace in $\mathbf{N}(n,d)$. In Section 5, we present an $O(n^3)$-time algorithm to finding the largest necklace in $\mathbf{N}(n,d)$ that is less than or equal to a given string. These results on necklaces are extended to Lyndon words in Section 6.

## 2. Preliminaries

Let $\alpha$ be a binary string and let $lyn(\alpha)$ denote the length of the longest prefix of $\alpha$ that is a Lyndon word. A *prenecklace* is a prefix of some necklace. The following theorem by Cattell et al. [1] has been called *The Fundamental Theorem of Necklaces*:

**Theorem 2.1.** *Let $\alpha = a_1a_2\cdots a_{n-1}$ be a prenecklace over the alphabet $\Sigma = \{0,1,\ldots,k-1\}$ and let $p = lyn(\alpha)$. Given $b \in \Sigma$, the string $\alpha b$ is a prenecklace if and only if $a_{n-p} \le b$. Furthermore,*

$$lyn(\alpha b) = \begin{cases} p & \text{if } b = a_{n-p} \\ n & \text{if } b > a_{n-p} \quad \text{(i.e., } \alpha b \text{ is a Lyndon word).} \end{cases}$$

**Corollary 2.2.** *If $\alpha b$ is a prenecklace then $\alpha(b+1)$ is a Lyndon word.*

**Corollary 2.3.** *If $\alpha = a_1a_2\cdots a_n$ is a necklace then $\alpha a_1$ is a prenecklace and $\alpha b$ is a Lyndon word for all $b > a_1$.*

The following is well-known property of Lyndon words by Reutenauer [6].

**Lemma 2.4.** *If $\alpha$ and $\beta$ are Lyndon words such that $\alpha < \beta$ then $\alpha\beta$ is a Lyndon word.*

**Corollary 2.5.** *If $\alpha$ is a Lyndon word and $\beta$ is a necklace such that $\alpha \le \beta$ then $\alpha\beta^t$ is a necklace for $t \ge 1$.*

**Proof.** If $\alpha = \beta$ then clearly $\alpha\beta^t$ is a (periodic) necklace. If $\beta$ is a Lyndon word, then the result follows from repeated application of Lemma 2.4. Otherwise $\beta = \delta^i$ for some Lyndon word $\delta$ and $i > 1$. Note that $\alpha \le \delta$ because otherwise $\alpha > \beta$. If $\alpha < \delta$, then repeated application of Lemma 2.4 implies that $\alpha\delta^j$ is a Lyndon word for all $j \ge 0$. If $\alpha = \delta$, then clearly $\alpha\delta^j$ is a (periodic) necklace for all $j \ge 1$. In both cases, $\alpha\beta^t$ will be a necklace for all $t \ge 1$. □

**Lemma 2.6.** *A $k$-ary string $\alpha = a_1a_2\cdots a_n$ over alphabet $\{0,1,\ldots,k-1\}$ is a necklace if and only if $0^{t-a_1}10^{t-a_2}1\cdots 0^{t-a_n}1$ is a necklace for all $t \ge k-1$.*

**Proof.** ($\Rightarrow$) Assume $\alpha$ is a necklace. Let $\beta = 0^{t-a_1}10^{t-a_2}1\cdots 0^{t-a_n}1$ for some $t \ge k-1$. If $\beta$ is not a necklace then there exists some $2 \le i \le n$ such that $0^{t-a_i}10^{t-a_{i+1}}1\cdots 0^{t-a_n}10^{t-a_1}1\cdots 0^{t-a_{i-1}}1 < \beta$. But this implies $a_ia_{i+1}\cdots a_na_1\cdots a_{i-1} < \alpha$, contradicting the assumption that $\alpha$ is a necklace. Thus $\beta$ is a necklace. ($\Leftarrow$) Assume $\beta = 0^{t-a_1}10^{t-a_2}1\cdots 0^{t-a_n}1$ is a necklace for all $t \ge k-1$. If $\alpha$ is not a necklace then there exists some $2 \le i \le n$ such that $a_ia_{i+1}\cdots a_na_1\cdots a_{i-1} < \alpha$. But this implies that $0^{t-a_i}10^{t-a_{i+1}}1\cdots 0^{t-a_n}10^{t-a_1}1\cdots 0^{t-a_{i-1}}1 < \beta$, contradicting the assumption that $\beta$ is a necklace. Thus $\alpha$ is a necklace. □

**Lemma 2.7.** *A $k$-ary string $\alpha = a_1a_2\cdots a_n$ over alphabet $\{0,1,\ldots,k-1\}$ is a necklace if and only if $01^{t+a_1}01^{t+a_2}\ldots 01^{t+a_n}$ is a necklace for all $t \ge 0$.*

**Proof.** ($\Rightarrow$) Assume $\alpha$ is a necklace. Let $\beta = 01^{t+a_1}01^{t+a_2}\cdots 01^{t+a_n}$ for some $t \ge 0$. If $\beta$ is not a necklace there exists some $2 \le i \le n$ such that $01^{t+a_i}01^{t+a_{i+1}}\ldots 01^{t+a_n}01^{t+a_1}\ldots 01^{t+a_{i-1}} < \beta$. But this implies $a_ia_{i+1}\cdots a_na_1\cdots a_{i-1} < \alpha$, contradicting the assumption that $\alpha$ is a necklace. Thus $\beta$ is a necklace. ($\Leftarrow$) Assume $\beta = 01^{t+a_1}01^{t+a_2}\ldots 01^{t+a_n}$ is a necklace for all $t \ge 0$. If $\alpha$ is not a necklace there exists some $2 \le i \le n$ such that $a_ia_{i+1}\cdots a_na_1\cdots a_{i-1} < \alpha$. But this implies that $01^{t+a_i}01^{t+a_{i+1}}\ldots 01^{t+a_n}01^{t+a_1}\ldots 01^{t+a_{i-1}} < \beta$, contradicting the assumption that $\beta$ is a necklace. Thus $\alpha$ is a necklace. □

## 3. Finding the largest necklace with a given density

Let LargestNeck$(n,d)$ denote the lexicographically largest binary necklace in $\mathbf{N}(n,d)$.

**Lemma 3.1.** *Let $0 < d \le n$ and $t = \lfloor \frac{n}{d} \rfloor$. Then*

$$\text{LargestNeck}(n,d) = 0^{t-b_1}10^{t-b_2}1\cdots 0^{t-b_d}1,$$

*where $b_1b_2\cdots b_d = \text{LargestNeck}(d, d-(n \bmod d))$.*

**Proof.** Since $d > 0$, $\alpha = \text{LargestNeck}(n,d)$ can be written as $0^{c_1}10^{c_2}1\cdots 0^{c_d}1$ where each $c_i \ge 0$. Let $x = d - (n \bmod d)$. Observe that $\alpha \ge (0^t1)^{d-x}(0^{t-1}1)^x \in \mathbf{N}(n,d)$ (it is a simple calculation to verify the length). Thus, $c_1 \le t$, and moreover each $c_i \le t$ since $\alpha$ is a necklace. Therefore $\alpha$ can be expressed as $0^{t-b_1}10^{t-b_2}1\ldots 0^{t-b_d}1$ for some string $\beta = b_1b_2\cdots b_d$ over the alphabet $\{0,1,\ldots,t\}$. By Lemma 2.6, $\beta$ is a necklace. Suppose there is some largest $1 \le i \le d$ such that $b_i > 1$. Thus, each element of $b_{i+1}\cdots b_d$ must be in $\{0,1\}$. Since $\beta$ is a necklace, each of its rotations $b_j\cdots b_db_1\cdots b_{j-1} \ge \beta$. Thus, we can deduce that if $j > i$ then $b_j\cdots b_db_1\cdots b_{j-1} > b_1b_2\cdots b_{i-1}$. This implies that $b_j\cdots b_db_1\cdots b_{i-1} > b_1b_2\cdots b_{i-1}$. Now consider $\gamma = b_1b_2\cdots b_{i-2}(b_{i-1}{+}1)b_{i+1}\cdots b_d$. Since $b_1b_2\cdots b_{i-1}$ is a prenecklace, $b_1b_2\cdots b_{i-2}(b_{i-1}{+}1)$ is a Lyndon word by Corollary 2.2. Thus any proper rotation of $\gamma$ starting before $b_{i+1}$ will be strictly greater than $\gamma$. Now consider a rotation of $\gamma$ starting from $b_j$ for $i+1 \le j \le d$. Observe that a rotation starting from $b_j$ has prefix $b_j\cdots b_db_1\cdots b_{i-2}(b_{i-1}+1)$. We have already noted that $b_j\cdots b_db_1\cdots b_{i-1} > b_1b_2\cdots b_{i-1}$, and therefore the