# Tighter Bounds for the Sum of Irreducible LCP Values

Juha Kärkkäinen[1(✉)], Dominik Kempa[1], and Marcin Piątkowski[1,2]

[1] Helsinki Institute of Information Technology (HIIT) and Department of Computer Science, University of Helsinki, Helsinki, Finland
{juha.karkkainen,dominik.kempa}@cs.helsinki.fi

[2] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Torun, Poland
marcin.piatkowski@mat.umk.pl

**Abstract.** The suffix array is frequently augmented with the longest-common-prefix (LCP) array that stores the lengths of the longest common prefixes between lexicographically adjacent suffixes of a text. While the sum of the values in the LCP array can be $\Omega(n^2)$ for a text of length $n$, the sum of so-called irreducible LCP values was shown to be $\mathcal{O}(n \lg n)$ just a few years ago. In this paper, we improve the bound to $\mathcal{O}(n \lg r)$, where $r \le n$ is the number of runs in the Burrows-Wheeler transform of the text. We also show that our bound is tight up to lower order terms (unlike the previous bound). Our results and the techniques used in proving them provide new insights into the combinatorics of text indexing and compression, and have immediate applications to LCP array construction algorithms.

## 1 Introduction

The suffix array [8], a lexicographically sorted array of the suffixes of a text, is the most important data structure in modern string processing. Modern text books spend dozens of pages in describing applications of suffix arrays, see e.g. [12]. In many of those applications, the suffix array needs to be augmented with the longest-common-prefix (LCP) array, which stores the lengths of the longest common prefixes between lexicographically adjacent suffixes (see e.g. [1,12]).

A closely related array is the Burrows–Wheeler transform (BWT) [2], which stores the characters preceding each suffix in the lexicographical order of the suffixes. The BWT was designed for text compression and is at the heart of many compressed text indexes [11]. If a text is highly repetitive (and thus highly compressible), its BWT tends to contain long runs of the same character. For example, for any string $x$ and positive integer $k$, $x$ and $x^k$ have the same number of BWT runs [7]. Thus the number of BWT runs is a rough measure of the (in)compressibility of the text.

Partially supported by the project "Enhancing Educational Potential of Nicolaus Copernicus University" (project no. POKL.04.01.01-00-081/10).

F. Cicalese et al. (Eds.): CPM 2015, LNCS 9133, pp. 316–328, 2015.
DOI: 10.1007/978-3-319-19929-0_27

An entry LCP$[i]$ in the LCP array is called reducible if BWT$[i] =$ BWT$[i-1]$, and *irreducible* otherwise. Given all the irreducible LCP values, the reducible values are easy to compute, which has been utilized in several LCP array construction algorithms [5,6,10,14]. There is also a compressed representation of the LCP array based on the fact that the number of irreducible values is one less than the number of BWT runs and thus small for repetitive texts [14].

The sum of irreducible LCP values was shown to be $\mathcal{O}(n \lg n)$ for a text of length $n$ in [6], and there are LCP array construction algorithms relying on this bound [5,6,14]. In this paper, we improve the bound to $n \lg r + \mathcal{O}(n)$, where $r$ is the number of BWT runs.[1] This immediately gives better time complexities for the algorithms in [6,14]. The tightness of our bound is shown by an infinite family of strings with the irreducible LCP sum of $n \lg r - \mathcal{O}(n)$.

Our proofs are derived in a setting where the suffix array, LCP array and BWT are defined for an arbitrary multiset of strings, closely related to the extended BWT introduced in [9]. This general setting offers cleaner combinatorics — for example, our upper and lower bounds match exactly in this setting — and could be useful for studying other topics in the combinatorics of text indexes.

## 2 Preliminaries

By $\mathcal{A}$ we denote a finite ordered set, called the *alphabet*. Elements of the alphabet are called *letters*. A finite *word* over $\mathcal{A}$ is a finite sequence of letters $w = a_0 a_1 \dots a_{n-1}$. The length of a word $w$ is defined as the number of its letters and denoted by $|w|$. An empty sequence of letters, called the *empty word*, is denoted by $\varepsilon$. The set of all finite words over $\mathcal{A}$ is denoted by $\mathcal{A}^*$ and the set of all non-empty words over $\mathcal{A}$ by $\mathcal{A}^+ = \mathcal{A}^* \setminus \{\varepsilon\}$.

For two words $x = a_0 a_1 \dots a_{m-1}$ and $y = b_0 b_1 \dots b_{n-1}$, their *concatenation* is $xy = x \cdot y = a_0 a_1 \dots a_{m-1} b_0 b_1 \dots b_{n-1}$. For a word $w$ and an integer $k \geq 1$, we use $w^k$ to denote the concatenation of $k$ copies of $w$, also called a *power* of $w$. A word $w$ is *primitive* if $w$ is not a power of some other word. The *root* of a word $w$ is defined as the shortest word $u = \mathrm{root}(w)$ such that $w = u^k$ for some $k \geq 1$.

A word $u$ is a *factor* of a word $w$ if there exist words $x$ and $y$ such that $w = xuy$. Moreover, $u$ is a *prefix* (resp. a *suffix*) of $w$ if $x = \varepsilon$ (resp. $y = \varepsilon$). By $\mathrm{lcp}(u, v)$ we denote the length of the *longest common prefix* of $u$ and $v$. For a word $w = a_0 \dots a_{n-1}$ and $i, j \in [0..n)$ by $w[i..j]$ we denote its factor of the form $a_i a_{i+1} \dots a_j$. A factor/prefix/suffix $u$ of $w$ is *proper* if $u \neq w$. A (multi)set of words $W$ is *prefix-free* if no word in $W$ is a proper prefix of another word in $W$.

The order on letters of $\mathcal{A}$ can be extended in a natural way into the *lexicographical order* of words. For any two words $x$ and $y$ we have $x < y$ if $x$ is a proper prefix of $y$ or we have $x = uav_1$ and $y = ubv_2$, where $a, b \in \mathcal{A}$ and $a < b$.

Let $a \in \mathcal{A}$ and $x \in \mathcal{A}^*$. We define a *rotation* operator $\sigma : \mathcal{A}^+ \to \mathcal{A}^+$ as $\sigma(a \cdot x) \mapsto x \cdot a$, a *first-letter* operator $\tau : \mathcal{A}^+ \to \mathcal{A}$ as $\tau(a \cdot x) \mapsto a$ and a *reverse*

[1] Throughout the paper we use lg as a shorthand for $\log_2$.

operator $\overline{\phantom{x}} : \mathcal{A}^* \to \mathcal{A}^*$ as $\overline{\varepsilon} \mapsto \varepsilon$ and $\overline{a \cdot x} \mapsto \overline{x} \cdot a$. We say that a word $w_1$ is a *conjugate* of a word $w_2$ if $w_1 = \sigma^k(w_2)$ for some $k$.

The set of *infinite periodic words* is defined as $(\mathcal{A}^+)^\omega = \{w^\omega : w \in \mathcal{A}^+\}$, where $w^\omega = w \cdot w \cdot \ldots$ is the infinite power of $w$. We extend several of the above operators to infinite periodic words: $\mathrm{root}(w^\omega) = \mathrm{root}(w)$, $\sigma(w^\omega) = \big(\sigma(w)\big)^\omega$, $\tau\big((a \cdot w)^\omega\big) = a$ and $\overline{(w^\omega)} = \big(\overline{w}\big)^\omega$. Some key properties are given below:

- The operators are well defined: If $u^\omega = v^\omega$ for two words $u$ and $v$, then $\mathrm{root}(u) = \mathrm{root}(v)$, $\sigma(u)^\omega = \sigma(v)^\omega$, $\tau(u) = \tau(v)$, and $\overline{u}^\omega = \overline{v}^\omega$.
- The rotation operator $\sigma$ is in fact a suffix operator for infinite periodic words: $w^\omega = \tau(w^\omega)\sigma(w^\omega)$ for all $w \in \mathcal{A}^+$. However, unlike a suffix operator for finite words, $\sigma$ has a well defined inverse $\sigma^{-1}$.
- The lexicographical ordering of infinite periodic words is not necessarily the same as their roots. For example, with alphabet $\{a < b\}$, $ab < aba$ but $(ab)^\omega > (aba)^\omega$. However, for two infinite periodic words with the roots $u$ and $v$, either $u^\omega = v^\omega$ (and $\mathrm{lcp}(u^\omega, v^\omega) = \omega$) or $\mathrm{lcp}(u^\omega, v^\omega) \le |u| + |v| - \gcd(|u|, |v|)$ due to properties of periodicity [3].

A *rooted tree* $T$ is a directed graph that contains no undirected cycles and where every vertex is reachable from a single vertex called the *root*. If $(u, v)$ is an edge in $T$, $u$ is the *parent* of $v$ and $v$ is a *child* of $u$. If there is a directed path from a vertex $u$ to vertex $v$, $u$ is an *ancestor* of $v$ and $v$ is a *descendant* of $u$. A subgraph of $T$ induced by the set of vertices that are reachable from a vertex $u$ is called the *subtree* rooted at $u$.

A *compact trie* is a rooted tree, where the edges are labelled by non-empty words so that, for any vertex $u$ with two outgoing edges $(u, v_1)$ and $(u, v_2)$, $\mathrm{lcp}(\mathrm{label}(u, v_1), \mathrm{label}(u, v_2)) = 0$. The edge labelling induces a vertex labelling: the label of a vertex $u$ is the concatenation of edge labels on the path from the root to $u$. The compact trie $\mathrm{CTrie}(W)$ for a set $W$ of words is the smallest compact trie that contains a vertex labelled by $w$ for every $w \in W$. If $W$ is prefix-free, a vertex $v$ in $\mathrm{CTrie}(W)$ is labelled by a word in $W$ if and only if $v$ is a leaf. For $W \subseteq (\mathcal{A}^+)^\omega$, the leafs and the leaf edges in $\mathrm{CTrie}(W)$ are labelled by infinite periodic words, but other edges and vertices have finite labels.

## 3 Cyclic Suffixes

In this section, we define a generalization of the suffix array and related data structures based on the concept of cyclic suffixes.

Let $W = \{\!\{w_i\}\!\}_{i=1}^s$ be a multiset[2] of words and $n = \sum_{i=1}^s |w_i|$. The set of positions of $W$ is defined as the set of integer pairs $\mathrm{pos}(W) := \big\{\langle i, p\rangle \ : \ i \in [1..s],\ p \in [0..|w_i|)\big\}$. For a position $\langle i, p\rangle \in \mathrm{pos}(W)$ we define a *cyclic suffix* $W_{\langle i,p\rangle} := \big(\sigma^p(w_i)\big)^\omega \in (\mathcal{A}^+)^\omega$. The multiset of all cyclic suffixes of $W$ is defined as $\mathrm{suf}(W) := \{\!\{W_{\langle i,p\rangle} \ : \ \langle i, p\rangle \in \mathrm{pos}(W)\}\!\}$.

[2] We use the double brace notation $\{\!\{\cdot\}\!\}$ to denote a multiset as opposed to a set.

We define two multisets $V$ and $W$ to be *cyclically equivalent* if $\mathrm{suf}(V) = \mathrm{suf}(W)$. It is easy to see that the corresponding equivalence classes are closed under conjugation of words in the multiset. Indeed, the restriction of cyclic equivalency to multisets of primitive words is the multiset conjugacy relation defined in [9]. The following lemma illustrates some further properties of our extension.

**Lemma 1.** *For any multiset of words* $W = \{\!\{w_i\}\!\}_{i=1}^{s}$ *there exists a multiset of primitive words* $V = \{\!\{v_i\}\!\}_{i=1}^{t}$, $t \geq s$, *such that* $\mathrm{suf}(W) = \mathrm{suf}(V)$, *and a set (not a multiset) of words* $U = \{u_i\}_{i=1}^{q}$, $q \leq s$, *such that* $\mathrm{suf}(W) = \mathrm{suf}(U)$.

*Proof.* To obtain $V$ we replace each non-primitive word $w = v^k \in W$, where $v = root(w)$, with $k$ occurrences of the primitive word $v$. To obtain $U$ we replace each word $w$ having $k$ occurrences in $W$ with a single word $u = w^k$. □

Over the multiset $\mathrm{pos}(W)$, we define a total order $\preceq_W$. We say that $\langle i, p\rangle \preceq_W \langle i', p'\rangle$ if $W_{\langle i,p\rangle} < W_{\langle i',p'\rangle}$, or $W_{\langle i,p\rangle} = W_{\langle i',p'\rangle}$ and $\langle i, p\rangle \leq \langle i', p'\rangle$, where the last comparison is the usual integer pair comparison, i.e. $(i_1, j_1) < (i_2, j_2)$ if $i_1 < i_2$ or $i_1 = i_2$ and $j_1 < j_2$.

The *(cyclic) suffix array* of a multiset of words $W$ is defined as an array $\mathrm{SA}_W[j] = \langle i_j, p_j\rangle$, where $\langle i_j, p_j\rangle \in \mathrm{pos}(W)$ for all $j \in [0..n)$ and $\langle i_{j-1}, p_{j-1}\rangle \prec_W \langle i_j, p_j\rangle$ for all $j \in [1..n)$. Note that for two cyclically equivalent multisets $V$ and $W$, we may have $\mathrm{SA}_V \neq \mathrm{SA}_W$ but always $V_{\mathrm{SA}_V[j]} = W_{\mathrm{SA}_W[j]}$ for all $j$.

The *longest-common-prefix array* $\mathrm{LCP}_W[1..n)$ is defined as $\mathrm{LCP}_W[j] = \mathrm{lcp}\big(W_{\mathrm{SA}[j-1]}, W_{\mathrm{SA}[j]}\big)$. The *distinguishing prefix array* $\mathrm{DP}_W[1..n)$ is defined as $\mathrm{DP}_W[i] = \mathrm{LCP}_W[i] + 1$. Note that we can have $\mathrm{LCP}_W[i] = \omega = \mathrm{DP}_W[i]$.

The *Burrows-Wheeler transform* $\mathrm{BWT}_W[0..n)$ (also denoted $\mathrm{BWT}(W)$) is defined as $\mathrm{BWT}_W[j] = \tau\big(\sigma^{-1}(W_{\mathrm{SA}[j]})\big)$. This definition is a natural generalization of the original one [2] defined for a single (not necessarily primitive) word and the one in [9] defined for a multiset of primitive words. It is easy to see that if multisets $V$ and $W$ are cyclically equivalent, then $\mathrm{LCP}_V = \mathrm{LCP}_W$, $\mathrm{DP}_V = \mathrm{DP}_W$ and $\mathrm{BWT}_V = \mathrm{BWT}_W$.

Let $v$ be a word of length $n$ and $\widehat{v}$ be obtained from $v$ by sorting its letters. The *standard permutation* [4] of $v$ is the permutation corresponding to the *stable* sorting of the letters, i.e., it is the mapping $\Psi_v : [0..n) \to [0..n)$ such that: for each $i \in [0..n)$ we have $\widehat{v}[i] = v[\Psi_v(i)]$ and for $\widehat{v}[i] = \widehat{v}[j]$ the relation $i < j$ implies $\Psi_v(i) < \Psi_v(j)$. Let IBWT be the mapping that maps a word $v$ into a multiset of (primitive) words $W$ as follows. Let $\Psi_v$ be a standard permutation of $v$ and $C = \{c_i\}_{i=1}^{s}$ its disjoint cycle decomposition. Then $W = \{\!\{w_i\}\!\}_{i=1}^{s}$ and for each $i \in [1..s]$ and $j \in [0..|c_i|)$ we define $w_i[j] = v[\Psi_v(c_i[j])]$. The mapping IBWT is the inverse of BWT in the sense that $\mathrm{BWT}(\mathrm{IBWT}(v)) = v$ for every word $v$. Thus the mapping from a word $v$ to the cyclical equivalence class of $\mathrm{IBWT}(v)$ is a bijection (see [9]).

*Example 1.* Let $W = \{\!\{ab, abaaba\}\!\}$. We have $v = \mathrm{BWT}(W) = bbaabaaa$ (having $r = 4$ runs) and $\Psi_v = (0,2,5)(1,3,6)(4,7)$. Then $\mathrm{IBWT}(v) = \{\!\{aab, aab, ab\}\!\}$, which is cyclically equivalent to $W$.

The *suffix tree* of $W$, denoted by $\mathrm{STree}(W)$, is the compact trie of suffixes $\mathrm{CTrie}(\mathrm{suf}(W))$. If $\mathrm{suf}(W)$ is a multiset, a single vertex in $\mathrm{STree}(W)$ represents all copies of a suffix, and the number of leaves in $\mathrm{STree}(W)$ is the number of *distinct* words in $\mathrm{suf}(W)$.

## 4 Irreducible Sums

For a multiset of words $W$, we say that a value $\mathrm{LCP}_W[i]$ is *reducible* if $\mathrm{BWT}_W[i-1] = \mathrm{BWT}_W[i]$ and *irreducible* otherwise. Observe that if $\mathrm{LCP}_W[i] = \omega$, then this value is obviously reducible. We say that a value $\mathrm{DP}_W[i]$ is irreducible if the corresponding value $\mathrm{LCP}_W[i]$ is irreducible. Let $\Sigma\mathrm{lcp}(W)$ denote the sum of all $\mathrm{LCP}_W$ values, $\Sigma\mathrm{ilcp}(W)$ the sum of all irreducible $\mathrm{LCP}_W$ values, and $\Sigma\mathrm{idp}(W)$ the sum of irreducible $\mathrm{DP}_W$ values. Note, that $\Sigma\mathrm{idp} = \Sigma\mathrm{ilcp} + r - 1$, where $r$ is the number of runs in the BWT. For technical reasons we analyze $\Sigma\mathrm{idp}$ rather than $\Sigma\mathrm{ilcp}$.

We define a *lexicographically adjacent repeat (LAR)* in a multiset $W$ as a tuple $(\langle i,p\rangle, \langle j,q\rangle, \ell)$ such that $\langle i,p\rangle, \langle j,q\rangle \in \mathrm{pos}(W)$, $\ell$ is a non-negative integer, $\mathrm{lcp}(W_{\langle i,p\rangle}, W_{\langle j,q\rangle}) \geq \ell$, $\langle i,p\rangle \prec_W \langle j,q\rangle$ and there exists no $\langle i',p'\rangle$ such that $\langle i,p\rangle \prec_W \langle i',p'\rangle \prec_W \langle j,q\rangle$ i.e., $W_{\langle i,p\rangle}$ and $W_{\langle j,q\rangle}$ are lexicographically adjacent suffixes with a common prefix of length (at least) $\ell$. A LAR $(\langle i,p\rangle, \langle j,q\rangle, \ell)$ is *left-maximal* if $(\langle i,p-1\rangle, \langle j,q-1\rangle, \ell+1)$ is not a LAR.

**Lemma 2.** *The number of left-maximal LARs in $W$ equals $\Sigma\mathrm{idp}(W)$.*

*Proof.* Clearly, the set of all LARs is exactly $\{(\mathrm{SA}[i-1], \mathrm{SA}[i], \ell) : i \in [1..n), \ell \in [0..\mathrm{DP}[i])\}$, and a LAR $(\mathrm{SA}[i-1], \mathrm{SA}[i], \ell)$ is left-maximal if and only if $\mathrm{DP}[i]$ is irreducible. $\square$

Let $T$ be a rooted tree and $\leq$ a total order over the leaves of $T$. Let $u$ and $v$ be leaves of $T$, and let $x$ be the nearest common ancestor of $u$ and $v$. The pair $(u, v)$ is called a *dispersal pair* if $u < v$ and the subtree rooted at $x$ contains no leaf $w$ such that $u < w < v$. Let $D_x(T, \leq)$ denote the set of dispersal pairs with $x$ as the nearest common ancestor. The *dispersal value* of $T$ with respect to $\leq$, denoted by $d(T, \leq)$, is the number of dispersal pairs in $T$.

Let $\overline{\mathrm{suf}}(W) = \{\!\{\overline{w} : w \in \mathrm{suf}(W)\}\!\}$ be the multiset of *reverse suffixes* of a multiset $W$. The reverse suffix tree $\overline{\mathrm{STree}}(W)$ of $W$ is $\mathrm{CTrie}(\overline{\mathrm{suf}}(W))$. Define a total order $\leq_W$ over the leaves of $\overline{\mathrm{STree}}(W)$ by $u \leq_W v \iff \langle i,p\rangle \preceq_W \langle j,q\rangle$, where $\overline{W_{\langle i,p\rangle}}$ is the label of $u$ and $\overline{W_{\langle j,q\rangle}}$ is the label of $v$. If $\overline{\mathrm{suf}}(W)$ contains duplicates, any of the identical reverse suffixes can be used as the representative of a vertex.

**Lemma 3.** $d(\overline{\mathrm{STree}}(W), \leq_W) = \Sigma\mathrm{idp}(W)$.

*Proof.* Let $(\mathrm{SA}[i-1], \mathrm{SA}[i], \ell)$ be a left-maximal LAR, i.e., $\mathrm{DP}[i] > \ell$ is irreducible. Let $x$ be the length $\ell$ prefix of $W_{\mathrm{SA}[i]}$, and let $y$ and $y'$ be infinite periodic words such that $xy = W_{\mathrm{SA}[i-1]}$ and $xy' = W_{\mathrm{SA}[i]}$. Then $\overline{x}$ is the longest common prefix of $\overline{y}$ and $\overline{y'}$. Let $u$, $v$ and $v'$ be the vertices of $\overline{\text{STree}}(W)$ that are labelled by $\overline{x}$, $\overline{y}$ and $\overline{y'}$, respectively. Then $v <_W v'$ and we will show that $(v, v')$ is a dispersal pair. Suppose $(v, v')$ is not a dispersal pair. Then there exists a leaf $v''$ descendant to $u$ such that $v <_W v'' <_W v'$. If $\overline{y''}$ is the label of $v''$, then $xy'' \in \text{suf}(W)$ and $xy < xy'' < xy'$, which contradicts $xy$ and $xy'$ being adjacent in SA. Thus $(v, v')$ is a dispersal pair. This mapping from left-maximal LARs to dispersal pairs is clearly injective, and thus $d(\overline{\text{STree}}(W), \leq_W) \geq \Sigma\text{idp}(W)$.

Let $(v, v')$ be a dispersal pair in $\overline{\text{STree}}(W)$, and let $u$ be the nearest common ancestor of $v$ and $v'$. Let $y$, $y'$ and $x$ be words such that $\overline{x}$, $\overline{y}$ and $\overline{y'}$ are the labels of $u$, $v$ and $v'$, respectively. Then $xy, xy' \in \text{suf}(W)$, $xy < xy'$ and $\tau(\sigma^{-1}(xy)) \neq \tau(\sigma^{-1}(xy'))$. Let $i$ be the largest integer such that $W_{\text{SA}[i]} = xy$ and $i'$ the smallest integer such that $W_{\text{SA}[i']} = xy'$. Then $i < i'$ and we will show that $i = i' - 1$. Suppose $i < i' - 1$ and let $i'' = i' - 1$. Then we must have $W_{\text{SA}[i]} < W_{\text{SA}[i'']} < W_{\text{SA}[i']}$ and $x$ is a prefix of $W_{\text{SA}[i'']}$. If $y''$ is the word such that $xy'' = W_{\text{SA}[i'']}$, then $\overline{y''} \in \overline{\text{suf}}(W)$ has $\overline{x}$ as a prefix. If $v''$ is the leaf in $\overline{\text{STree}}(W)$ labelled by $\overline{y''}$, then $v <_W v'' <_W v'$, which contradicts $(v, v')$ being a dispersal pair. Thus $i = i' - 1$ and $(\text{SA}[i-1], \text{SA}[i], |x|)$ is a left-maximal LAR. This mapping from dispersal pairs to left-maximal LARs is clearly injective and thus $d(\overline{\text{STree}}(W), \leq_W) \leq \Sigma\text{idp}(W)$. □

## 5 $n \lg n$ Upper Bound

We will now derive upper bounds on the maximum dispersal value of any tree with $n$ leaves. By Lemma 3, these bounds are upper bounds for $\Sigma$idp, too.

Define, for $n > 0$ and $k \in [1..\lfloor n/2 \rfloor]$,

$$\begin{aligned} d(1) &= 0 \\ d(n) &= \max_{i \in [1..\lfloor n/2 \rfloor]} d(n, i) \quad \text{when } n > 1 \\ d(n, k) &= d(k) + d(n-k) + \min\{2k, n-1\} \end{aligned}$$

**Lemma 4.** $d(n) = \max\{d(T, \leq)\}$, *where the maximum is taken over any rooted tree $T$ with $n$ leaves and any total order $\leq$ on the leaves of $T$.*

*Proof.* We will first prove that we can restrict ourselves to proper binary trees, where every non-leaf vertex has exactly two children. Let $T$ be a tree with a leaf order $\leq$, and let $u$ be a vertex with at least three children $v_1$, $v_2$ and $v_3$. Let $T'$ be the tree obtained from $T$ by adding a vertex $u'$ and replacing the edges $(u, v_1)$ and $(u, v_2)$ with $(u, u')$, $(u', v_1)$ and $(u', v_2)$. Let $w_1$, $w_2$ and $w_3$ be leaves in the subtrees rooted at $v_1$, $v_2$ and $v_3$, respectively. Then, $(w_1, w_2)$ could be a dispersal pair in $T'$ but not in $T$ if $w_1 < w_3 < w_2$. However, any dispersal pair in $T$ is a dispersal pair in $T'$ too. Thus $d(T, \leq) \leq d(T', \leq)$. The above procedure can be repeated as long as the tree contains vertices with more than two children to obtain a binary tree. Furthermore, one can similarly show that unary vertices can be removed without removing any dispersal pairs to obtain a proper binary tree.

Let then $T$ be a proper binary tree of size (number of leaves) $n \geq 2$ with a leaf order $\leq$. Let $T_L$ and $T_R$ be the left and right subtree of $T$ of sizes $k$ and $n-k$, respectively. W.l.o.g., assume that $k \leq n-k$. Let $\leq_L$ ($\leq_R$) be the leaf order $\leq$ restricted to the left (right) subtree. Let $D_{\mathrm{root}}(T, \leq)$ be the set of dispersal pairs with the root of $T$ as the nearest common ancestor. Then, clearly,

$$d(T, \leq) = d(T_L, \leq_L) + d(T_R, \leq_R) + |D_{\mathrm{root}}(T, \leq)| \,.$$

If $(u, v) \in D_{\mathrm{root}}(T, \leq)$, then $u$ and $v$ are adjacent in the order $\leq$, and one of $u$ and $v$ is in $T_L$ and the other is in $T_R$. A leaf $u$ can be involved with at most two pairs in $D_{\mathrm{root}}(T, \leq)$, once with its immediate predecessor in $\leq$ and once with its immediate successor. Thus $|D_{\mathrm{root}}(T, \leq)| \leq 2k$. Furthermore, if $k = n-k$, at least one of the leaves in $T_L$ is the first in $\leq$, the last in $\leq$ or adjacent to another leaf in $T_L$, and thus involved in at most one pair in $D_{\mathrm{root}}(T, \leq)$. Then $|D_{\mathrm{root}}(T, \leq)| \leq 2k-1 = n-1$. It is now easy to see that $d(n)$ is an upper bound on the dispersal value over trees of size $n$, by induction on $n$:

$$\begin{aligned} d(T, \leq) &= d(T_L, \leq_L) + d(T_R, \leq_R) + |D_{\mathrm{root}}(T, \leq)| \\ &\leq d(k) + d(n-k) + \min\{2k, n-1\} = d(n,k) \leq d(n) \,. \end{aligned}$$

We still need to show that, for every $n$, there exists a tree $T_n$ and its leaf order $\leq_n$ such that $d(T_n, \leq_n) = d(n)$. The case $n = 1$ is trivial. For $n > 1$ and $k \in [1..\lfloor n/2 \rfloor]$, let $T_{n,k}$ be a tree with $T_k$ and $T_{n-k}$ as the two subtrees. Define the leaf order $\leq_{n,k}$ so that the leaves at positions $2, 4, 6, \ldots, 2k$ come from $T_k$ consistent with the order $\leq_k$ and the leaves at positions $1, 3, 5, \ldots, 2k-1, 2k+1, 2k+2, \ldots, n$ come from $T_{n-k}$ consistent with the order $\leq_{n-k}$. Then, it is easy to see that

$$\begin{aligned} d(T_{n,k}, \leq_{n,k}) &= d(T_k, \leq_k) + d(T_{n-k}, \leq_{n-k}) + |D_{\mathrm{root}}(T_{n,k}, \leq_{n,k})| \\ &= d(k) + d(n-k) + \min\{2k, n-1\} = d(n,k) \end{aligned}$$

Finally, set $T_n = T_{n,k}$ and $\leq_n = \leq_{n,k}$, for $k = \operatorname{argmax}_i d(n,i)$. Then $d(T_n, \leq_n) = d(n)$. □

Basic properties and closed form equations for $d(n)$ are given in the following lemmas. The proofs are omitted due to lack of space.

**Lemma 5.** *For any* $2 \leq 2k \leq n$,

*(i)* $d(n,k) \leq d(n, \lfloor n/2 \rfloor)$
*(ii)* $d(n) - d(n-1) = \lceil \lg n \rceil$ .

**Lemma 6.** $d(n) = n\lceil \lg n \rceil - 2^{\lceil \lg n \rceil} + 1$.

**Lemma 7.** $d(n) = n \lg n - (1 - \alpha(n))n + 1$, *where*
$0 \leq \alpha(n) := 1 - 2^{\lceil \lg n \rceil}/n + \lg(2^{\lceil \lg n \rceil}/n) < (1 - \lg e + \lg \lg e) < 0.0861$.

Thus we obtain an $n \lg n$ bound on the irreducible sums.

**Theorem 1.** *For any multiset* $W$ *of words of total length* $n > 0$, *we have*

$$\Sigma\mathrm{ilcp}(W) \leq \Sigma\mathrm{idp}(W) \leq d(n) \leq n \lg n \,.$$

## 6 $n \lg r$ Upper Bound

We will now use the above machinery to improve the upper bound on $\Sigma\mathrm{idp}(W)$ when the number $r$ of runs in $\mathrm{BWT}(W)$ is given.

**Lemma 8.** *If* BWT$(W)$ *has $r$ runs, then* $|D_u(\overline{\mathrm{STree}}(W), \leq_W)| < r$ *for every vertex $u$ in* $\overline{\mathrm{STree}}(W)$.

*Proof.* Let $u$ be a vertex in $\overline{\mathrm{STree}}(W)$ labelled by $\overline{x}$. The bijection defined in the proof of Lemma 3 maps $D_u(\overline{\mathrm{STree}}(W), \leq_W)$ into

$$\{(\mathrm{SA}[i-1], \mathrm{SA}[i], |x|) : x \text{ is prefix of } W_{\mathrm{SA}[i-1]} \text{and } W_{\mathrm{SA}[i]}, \text{and } \mathrm{DP}[i] \text{ is irreducible}\}.$$

Since the total number of irreducible distinguishing prefixes is $r-1$, the size of this set cannot be more than $r-1$, and thus $|D_u(\overline{\mathrm{STree}}(W), \leq_W)| < r$. □

Define, for $r > 0$, $n > 0$ and $k \in [1..\lfloor n/2 \rfloor]$,

$$\begin{aligned} d_r(1) &= 0 \\ d_r(n) &= \max_{i \in [1..\lfloor n/2 \rfloor]} d_r(n,i) \quad \text{when } n > 1 \\ d_r(n,k) &= d_r(k) + d_r(n-k) + \min\{2k, n-1, r-1\} \end{aligned}$$

**Lemma 9.** $d_{r(n)} = max\{d(T, \leq)\}$*, where the maximum is taken over any rooted tree $T$ with $n$ leaves and any total order $\leq$ on the leaves of $T$ such that* $|D_u(T, \leq)| < r$ *for every vertex $u$ in $T$.*

*Proof.* We will prove the claim by modifying the construction utilized in the proof of Lemma 4. First note that in the transformation from a non-binary tree to a binary tree, the new vertex $u'$ might have $|D_{u'}(T', \leq)| \geq r$. However, we can then replace $\leq$ with $\leq'$ such that $|D_{u'}(T', \leq')| = r-1$ and no other vertex dispersal value is changed. Then we must have $|D_u(T', \leq')| + |D_{u'}(T', \leq')| \geq |D_u(T, \leq)|$ and thus $d(T, \leq) \leq d(T' \leq')$.

The inequality $d(T, \leq) \leq d_r(n)$ follows immediately from using the bound $|D_{\mathrm{root}}(T, \leq)| \leq \min\{2k, n-1, r-1\}$ in place of $|D_{\mathrm{root}}(T, \leq)| \leq \min\{2k, n-1\}$. In the construction of $T_n$ and $\leq_n$, the only difference is in constructing $\leq_{n,k}$ when $r-1 < \min\{2k, n-1\}$. In that case, the interleaving of $\leq_k$ and $\leq_{n-k}$ to obtain $\leq_{n,k}$ is then chosen so that $|D_{\mathrm{root}}(T_{n,k}, \leq_{n,k})| = r-1$. With this change, the construction shows that $d(T_n, \leq_n) = d_r(n)$. □

Basic properties and closed form equations for $d_r(n)$ are given in the following lemmas. Again, the proofs are omitted due to lack of space.

**Lemma 10.** *For any* $r \geq 2$,

*(i)* $d_r(n) = d(n)$ *if* $n \leq r$
*(ii)* $d_r(n,k) \leq d_r(n, \lfloor r/2 \rfloor)$ *if* $n \geq r$ *and* $k \leq n/2$
*(iii) for* $n > \lceil r/2 \rceil$,

$$d_r(n) - d_r(n-1) = \begin{cases} \lfloor \lg r \rfloor & \textit{if } n - \lceil r/2 \rceil - 1 \bmod \lfloor r/2 \rfloor \in [0..q) \\ \lceil \lg r \rceil & \textit{if } n - \lceil r/2 \rceil - 1 \bmod \lfloor r/2 \rfloor \in [q..\lfloor r/2 \rfloor) \end{cases}$$

*where* $q = 2^{\lceil \lg r \rceil - 1} - \lceil r/2 \rceil$.

**Lemma 11.** *For any* $2 \le r \le n$,

$$\begin{aligned} d_r(n) &= n\lceil \lg r \rceil - 2^{\lceil \lg r \rceil} + 1 - q(n-r-p)/\lfloor r/2 \rfloor - \min\{q,p\} \\ &\le n\lceil \lg r \rceil - 2^{\lceil \lg r \rceil} + 1 \end{aligned}$$

*where* $q = 2^{\lceil \lg r \rceil - 1} - \lceil r/2 \rceil$ *and* $p = (n-r) \bmod \lfloor r/2 \rfloor$.

**Lemma 12.** *For any* $2 \le r \le n$,

$$\begin{aligned} d_r(n) &= n \lg r + n(\alpha(r) + \beta(r)) - r(1+\beta(r)) - \gamma(p,q) + 1 \\ &\le n \lg r + n\alpha(r) - r + \begin{cases} 1 & \textit{if } r \textit{ is even} \\ \frac{n}{r} & \textit{if } r \textit{ is odd} \end{cases}, \end{aligned}$$

*where* $p$ *and* $q$ *are as in Lemma 11,* $\alpha(r) \in [0, 0.0861)$ *is as in Lemma 7,*

$$\beta(r) = \begin{cases} 0 & \textit{if } r \textit{ is even} \\ \frac{2r - 2^{\lceil \lg r \rceil}}{r(r-1)} & \textit{if } r \textit{ is odd} \end{cases} \in [0, 1/r]$$

*and* $\gamma(p,q) = \min\{p,q\} - pq/\lfloor r/2 \rfloor \in [0, r/8)$.

Thus we obtain the following upper bound on the irreducible sums.

**Theorem 2.** *For any multiset* $W$ *of words of total length* $n > 0$ *such that* $\mathrm{BWT}(W)$ *has* $r$ *runs, we have*

$$\Sigma\mathrm{ilcp}(W) + r - 1 = \Sigma\mathrm{idp}(W) \le d_r(n) < n \lg r + 0.0861 \cdot n + n/r - r = n \lg r + \mathcal{O}(n)\,.$$

## 7 $n \lg n$ Lower Bound

In this and the next section, we will show the tightness of the above upper bounds by constructing words and sets of words with matching irreducible sums. We will deal only with sets (not multisets) over the binary alphabet $\{a, b\}$, which allows a useful characterization of the LCP array.

**Lemma 13.** *For any set of words* $W$ *of total length least two, such that* $\mathrm{suf}(W)$ *contains no duplicates, the sequence of the depths of internal (non-leaf) vertices in* $\mathrm{STree}(W)$ *listed in inorder is exactly* $\mathrm{LCP}_W$.

*Proof.* Consider constructing $\mathrm{STree}(W)$ by inserting the suffixes into a compact trie one at a time in the lexicographical order. When inserting $W_{\mathrm{SA}[i]}$, $i > 0$, we add exactly two vertices, the leaf $v_i$ labelled $W_{\mathrm{SA}[i]}$ and the parent $u_i$ of $v_i$. Note that $u_i$ could not have existed (or was the root and unary) before, since the tree is binary. Since the depth of $u_i$ must be $\mathrm{LCP}[i]$, and the internal vertices are inserted in inorder, the claim follows. □

A word set $W \subseteq \{a, b\}^*$ is a de Bruijn set [4] of order $k \geq 1$ if every $v \in \{a, b\}^k$ is a prefix of exactly one word in $\mathrm{suf}(W)$ (and thus $\mathrm{suf}(W)$ is a set).

**Lemma 14.** *For any de Bruijn set $W$ of order $k$, $\Sigma\mathrm{lcp}(W) = k2^k - 2^{k+1} + 2$.*

*Proof.* $\mathrm{STree}(W)$ has $2^i$ vertices at depth $i \in [0..k-1]$ and no internal vertices at levels $i \geq k$. By Lemma 13, $\Sigma\mathrm{lcp}(W) = \sum_{i=1}^{k-1} i2^i = k2^k - 2^{k+1} + 2$. □

Higgins [4] showed the following characterization of the de Bruijn sets.

**Lemma 15** *([4]). For $k \geq 1$, and any $u \in U_k = \{ab, ba\}^{2^{k-1}}$, $W = \mathrm{IBWT}(u)$ is a de Bruijn set.*

In particular, $W_k = \mathrm{IBWT}((ab)^{2^{k-1}})$ is a de Bruijn set. Since every entry in $\mathrm{LCP}_{W_k}$ is irreducible, we obtain the following result.

**Theorem 3.** *For any $k \geq 1$, $\Sigma\mathrm{ilcp}(W_k) = k2^k - 2^{k+1} + 2 = n \lg n - 2n + 2$ and $\Sigma\mathrm{idp}(W_k) = \Sigma\mathrm{ilcp}(W_k) + n - 1 = n \lg n - n + 1$, where $n = 2^k$ is the total length of the words in $W_k$.*

Thus $\Sigma\mathrm{idp}(W_k)$ matches the upper bound from Sect. 5 exactly. We still want to show that, for any $k \geq 1$, there exist a de Bruijn word, i.e., a de Bruijn set of size one, that matches the upper bound within $\mathcal{O}(n)$. First we need a bound on the size of $W_k$.

**Lemma 16.** *$|W_k| \leq (2^k + (k-1)2^{k/2})/k$.*

*Proof.* From [4, Theorem 3.8], the size of $W_k$ is equal to the number of Lyndon words of length dividing $k$. Thus the claim follows by combining Eqs. (7.10) and (7.13) from [13].

**Lemma 17.** *Starting with $u = u_k = (ab)^{2^{k-1}}$, there exists a sequence of $|W_k| - 1$ swaps of the form $u[2i] \leftrightarrow u[2i+1]$ resulting in $u \in U_k$ such that $|\mathrm{IBWT}(u)| = 1$.*

*Proof.* We will show that the following invariant is maintained during the sequence of swaps until $|\mathrm{IBWT}(u)| = 1$: there exists $i$ such that every value in $[0..2i]$ belongs to the same cycle in $\Psi_u$, $2i+1$ belongs to a different cycle, and there has been no swaps affecting $u[2i..2^k)$. Then the next swap is $u[2i] \leftrightarrow u[2i+1]$. The invariant is clearly true when $u = u_k$.

Let $j = \Psi_u^{-1}(2i)$ and $j' = \Psi_u^{-1}(2i+1)$. Since $u[2i] = a$ and $u[2i+1] = b$, after the swap we have $\Psi_u(j) = 2i+1$ and $\Psi_u(j') = 2i$, i.e., the two cycles were merged. The values of $\Psi_u$ are not affected elsewhere. Now consider the smallest $j \in [2i+2..2^k)$ that is not in the same cycle as 0. We must have $u[j] = b$, since otherwise $\Psi_u^{-1}(j) < j$ and $j$ would be in the same cycle. Thus $j$ is odd, and we can choose $i = (j-1)/2$ to satisfy the invariant. □

**Theorem 4.** *For any $k \geq 1$, there exists a word $w$ of length $n = 2^k$ such that $\Sigma\mathrm{idp}(w) = n \lg n - \mathcal{O}(n)$.*

*Proof.* Let $u \in U_k$ be the result of Lemma 17 and $w = \mathrm{IBWT}(u)$. From the proof of Lemma 14, $\max \mathrm{LCP}_w = k - 1$. Each swap reduces the number of irreducible LCP values by at most two, thus the initial $\Sigma\mathrm{idp}(W_k) = n \lg n - \mathcal{O}(n)$ is reduced by at most $(\max \mathrm{LCP}_w + 1)|W_k| \leq k(2^k + (k-1)2^{k/2})/k = \mathcal{O}(n)$. □

## 8 $n \lg r$ Lower Bound

We will now extend the above lower bound results to cases where $r \ll n$. The following lemma shows the key idea of the construction.

**Lemma 18.** *Let $u \in \{a, b\}^{2k}$, $k \geq 1$ be a word containing exactly $k$ a's and $k$ b's. Then, for any $w = ua^{jk}$, $j \geq 0$, it holds* $|\mathrm{IBWT}(w)| = |\mathrm{IBWT}(u)|$. *Furthermore,* $\mathrm{IBWT}(w)$ *can be obtained by replacing every occurrence of $b$ in all* $\mathrm{IBWT}(u)$ *with $a^j b$.*

*Proof.* The standard permutation of $w$ can be expressed by the following formula:

$$\Psi_w(i) = \begin{cases} \Psi_u(i) & 0 \leq i < k \\ i + k & k \leq i < |w| - k \\ \Psi_u(i - jk) & |w| - k \leq i < |w| \end{cases}$$

Let $j > 0$ (since the case $j = 0$ is trivial), and compare the reconstructions of $\mathrm{IBWT}(u)$ and $\mathrm{IBWT}(w)$ by following $\Psi_u$ and $\Psi_w$. Whenever we visit $i \in [0..k)$, $\Psi_w(i) = \Psi_u(i)$ and we append letter $a$ to the currently decoded word in both cycles. When visiting $i \in [k..2k)$ we append $b$ to $\mathrm{IBWT}(u)$ but $a$ to $\mathrm{IBWT}(w)$ and $\Psi_w(i) \neq \Psi_u(i)$. However, for any such $i$, and any $p \in [1..j]$, we have $\Psi_w^p(i) = i + pk$, and thus $\Psi_w^{j+1}(i) = \Psi_w(i + jk) = \Psi_u(i)$. Therefore, after a detour of $j$ extra steps in $\Psi_w$ the cycles meet again, and where a single $b$ was appended $\mathrm{IBWT}(u)$, $a^j b$ was appended to $\mathrm{IBWT}(w)$. $\square$

Define $U_{k,j} = \{ab, ba\}^{2^{k-1}} a^{j2^{k-1}}$ for $k \geq 1$ and $j \geq 0$. Consider arbitrary $u \in U_{k,j}$ for some $k$ and $j$, and let $W = \mathrm{IBWT}(u)$. Let $S_{k,j} = S_{0,k,j} \cup S_{1,k,j} \cup \ldots \cup S_{j+1,k,j}$, where

$$S_{i,k,j} = \begin{cases} a^i ba^j \{a, ba^j\}^{k-1} & \text{if } i \leq j \\ a^{j+1} \{a, ba^j\}^{k-1} & \text{if } i > j \end{cases} .$$

**Lemma 19.** *Every word in $S_{k,j}$ is a prefix of exactly one word in* $\mathrm{suf}(W)$.

*Proof.* First observe that, since $S_{k,j}$ is prefix-free and $|S_{k,j}| = (j + 2)2^{k-1} = |\mathrm{suf}(W)|$, it is sufficient to show that every $w \in S_{k,j}$ is a prefix of at least one word in $\mathrm{suf}(W)$.

Let $u' \in \{ab, ba\}^{2^{k-1}}$ be the word such that $u = u'a^{j2^{k-1}}$, and let $W' = \mathrm{IBWT}(u')$. Since $W'$ is a de Bruijn set, for every $v' \in \{a, b\}^k$ either $v'a^j$ or $v'a^h b$ for some $h < j$ is a prefix of a word in $\mathrm{suf}(W')$. Thus, by Lemma 18, every $v \in \{a, a^j b\}^k a^j = a^j \{a, ba^j\}^k$ is a prefix of a word in $\mathrm{suf}(W)$. Since every word $w \in S_{k,j}$ is a factor of a word $v \in a^j \{a, ba^j\}^k$, $w$ must be a prefix of a word in $\mathrm{suf}(W)$ too. $\square$

It is easy to see from the definition of $S_{i,k,j}$, that $\mathrm{CTrie}(S_{i,k,j})$, consists of a full binary tree of height $k - 1$ connected to the root with a single edge, and that $\mathrm{CTrie}(S_{k,j})$ consists of $j + 2$ such full binary subtrees connected to the main branch labelled $a^j$, see Fig. 1 for examples.

Define $u_{k,j} = (ab)^{2^{k-1}} a^{j2^{k-1}} \in U_{k,j}$ and let $W_{k,j} = \mathrm{IBWT}(u_{k,j})$. Since $u_{k,j}$, $j \geq 1$ has $2^k + 1$ runs, $\Sigma\mathrm{idp}(W_{k,j}) = \Sigma\mathrm{ilcp}(W_{k,j}) + 2^k$.

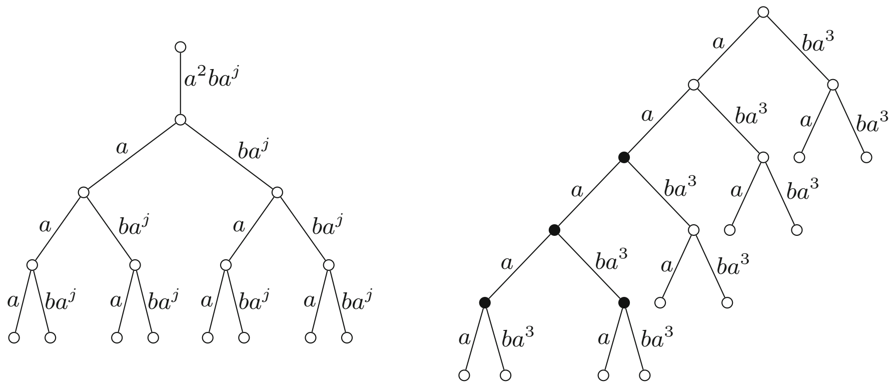

**Fig. 1.** Left: CTrie($S_{2,4,j}$). Right: CTrie($S_{2,3}$). Dark vertices correspond to irreducible LCP values of $W_{2,3}$.

**Lemma 20.** *For* $j \geq 1$, $\Sigma\mathrm{ilcp}(W_{k,j}) = (j+2)k2^{k-1} - 2^{k+1} + j + 1$.

*Proof.* By Lemma 19, STree($W_{k,j}$) is the same as CTrie($S_{k,j}$) except leaf labels have been extended to infinite words. Clearly, all LCP values in $\mathrm{LCP}_{W_{k,j}}[1..2^k]$ (and only these) are irreducible, and by Lemma 13, they correspond to the depths of the first $2^k$ internal vertices in CTrie($S_{k,j}$) by inorder. These are exactly the $2^k - 1$ vertices in the subtree CTrie($\{a, ba^j\}^k$) rooted in the vertex labelled $a^j$ plus one more vertex labelled $a^{j-1}$ (see Fig. 1). The sum of internal vertex depths of CTrie($\{a, ba^j\}^k$) is $(j+2)(2^{k-1}(k-2)+1)$, thus $\Sigma\mathrm{ilcp}(W_{k,j}) = (j+2)(2^{k-1}(k-2)+1) + j(2^k-1) + j - 1 = (j+2)k2^{k-1} - 2^{k+1} + j + 1$. □

Thus we obtain the following bounds for a set of words (with the proof omitted due to lack of space) and for a single word.

**Theorem 5.** *For any* $k \geq 1$ *and* $j \geq 1$, $\Sigma\mathrm{idp}(W_{k,j}) = d_r(n)$ *matching the upper bound in Sect. 6, where* $n = (j+2)2^{k-1}$ *is the total length of the words in* $W_{k,j}$ *and* $r = 2^k + 1$ *is the number of runs in* BWT($W_{k,j}$).

**Theorem 6.** *For any* $r = 2^k + 1$, $k \geq 1$, *and* $n \geq r$ *such that* $2^{k-1} | n$, *there exists a word* $w$ *of length* $n$ *such that* BWT($w$) *contains* $r - o(r)$ *runs and* $\Sigma\mathrm{idp}(w) = n \lg r - \mathcal{O}(n)$.

*Proof.* Let $u = u_{k,j}$, where $j = n/2^{k-1} - 2 \geq 1$. From definition, $u$ has $2^k + 1 = r$ runs. Lemmas 16 and 18 give $|\mathrm{IBWT}(u)| = |W_{k,j}| = |W_k| = \mathcal{O}(r/\lg r)$. Note, that Lemma 17 applies also for $u \in U_{k,j}$. Let $u' \in U_{k,j}$ be the result and $w = \mathrm{IBWT}(u')$. From Lemma 19, $\mathrm{LCP}_w = \mathrm{LCP}_{W_{k,j}}$. Furthermore, $\max \mathrm{LCP}_w$ corresponds to the deepest internal vertex in CTrie($S_{k,j}$), i.e., $\max \mathrm{LCP}_w = k(j+1) - 1 = \mathcal{O}((n \lg r)/r)$. Each swap in Lemma 17 reduces the number of irreducible LCP values by at most two, thus $u' = \mathrm{BWT}(w)$ contains $r - \mathcal{O}(r/\lg r) = r - o(r)$ runs. Altogether, the initial $\Sigma\mathrm{idp}$ is reduced by at most $|W_{k,j}|(\max \mathrm{LCP}_w + 1) = \mathcal{O}(n)$, which gives $\Sigma\mathrm{idp}(w) = \Sigma\mathrm{idp}(W_{k,j}) - \mathcal{O}(n) = n \lg r - \mathcal{O}(n)$ (see Theorem 5). □

## References

1. Abouelhoda, M.I., Kurtz, S., Ohlebusch, E.: Replacing suffix trees with enhanced suffix arrays. J. Discrete Algorithms **2**(1), 53–86 (2004)
2. Burrows, M., Wheeler, D.J.: A block sorting lossless data compression algorithm. Technical report 124, Digital Equipment Corporation, Palo Alto, California (1994)
3. Fine, N.J., Wilf, H.S.: Uniqueness theorems for periodic functions. Proc. Amer. Math. Soc. **16**(1), 109–114 (1965)
4. Higgins, P.M.: Burrows-Wheeler transformations and de Bruijn words. Theor. Comput. Sci. **457**, 128–136 (2012)
5. Kärkkäinen, J., Kempa, D.: LCP array construction in external memory. In: Gudmundsson, J., Katajainen, J. (eds.) SEA 2014. LNCS, vol. 8504, pp. 412–423. Springer, Heidelberg (2014)
6. Kärkkäinen, J., Manzini, G., Puglisi, S.J.: Permuted longest-common-prefix array. In: Kucherov, G., Ukkonen, E. (eds.) CPM 2009. LNCS, vol. 5577, pp. 181–192. Springer, Heidelberg (2009)
7. Mäkinen, V., Navarro, G., Sirén, J., Välimäki, N.: Storage and retrieval of highly repetitive sequence collections. J. Comp. Biol. **17**(3), 281–308 (2010)
8. Manber, U., Myers, G.W.: Suffix arrays: a new method for on-line string searches. SIAM J. Comp. **22**(5), 935–948 (1993)
9. Mantaci, S., Restivo, A., Rosone, G., Sciortino, M.: An extension of the Burrows-Wheeler transform. Theor. Comput. Sci. **387**(3), 298–312 (2007)
10. Manzini, G.: Two space saving tricks for linear time LCP array computation. In: Hagerup, T., Katajainen, J. (eds.) SWAT 2004. LNCS, vol. 3111, pp. 372–383. Springer, Heidelberg (2004)
11. Navarro, G., Mäkinen, V.: Compressed full-text indexes. ACM Comput. Surv. **39**(1), 1–61 (2007)
12. Bioinformatics Algorithms: Sequence Analysis, Genome Rearrangements, and Phylogenetic Reconstruction. Oldenbusch Verlag, Bremen, Germany (2013)
13. Ruskey, F.: Combinatorial generation, working version (1j-CSC 425/520) (2003)
14. Sirén, J.: Sampled longest common prefix array. In: Amir, A., Parida, L. (eds.) CPM 2010. LNCS, vol. 6129, pp. 227–237. Springer, Heidelberg (2010)