Computing the Longest Common Prefix Array Based on the Burrows-Wheeler Transform

Timo Beller, Simon Gog, Enno Ohlebusch and Thomas Schnattinger

Institute of Theoretical Computer Science
Ulm University

Suffix-Array

i		$S_{SA[i]}$
1	1	annasanannas$
2	2	nnasanannas$
3	3	nasanannas$
4	4	asanannas$
5	5	sanannas$
6	6	anannas$
7	7	nannas$
8	8	annas$
9	9	nnas$
10	10	nas$
11	11	as$
12	12	s$
13	13	$
14		

Suffix-Array

i	$SA[i]$	$S_{SA[i]}$
1	13	$
2	6	anannas$
3	8	annas$
4	1	annasanannas$
5	11	as$
6	4	asanannas$
7	7	nannas$
8	10	nas$
9	3	nasanannas$
10	9	nnas$
11	2	nnasanannas$
12	12	s$
13	5	sanannas$
14		

Suffix-Array construction algorithms

Many algorithms, see survey paper of Puglisi et al. 2007:

- Time: $\mathcal{O}(n)$ to $\mathcal{O}(n^2 \log n)$
- Space: $5n$ to $18n$ bytes

DivSufSort of Yuta Mori 2008:

- Time: $\mathcal{O}(n \log n)$
- Space: $5n$ bytes

InducedSort of Nong et al. 2009:

- Time: $\mathcal{O}(n)$
- Space: $5n$ bytes

BWT (Burrows–Wheeler transform)

i	$SA[i]$	$S_{SA[i]}$
1	13	$
2	6	anannas$
3	8	annas$
4	1	annasanannas$
5	11	as$
6	4	asanannas$
7	7	nannas$
8	10	nas$
9	3	nasanannas$
10	9	nnas$
11	2	nnasanannas$
12	12	s$
13	5	sanannas$
14		

BWT (Burrows–Wheeler transform)

i	$SA[i]$	$BWT[i]$	$S_{SA[i]}$
1	13	s	$
2	6	s	anannas$
3	8	n	annas$
4	1	$	annasanannas$
5	11	n	as$
6	4	n	asanannas$
7	7	a	nannas$
8	10	n	nas$
9	3	n	nasanannas$
10	9	a	nnas$
11	2	a	nnasanannas$
12	12	a	s$
13	5	a	sanannas$
14			

BWT construction algorithms

Compute BWT from suffix array:

- Time: $\mathcal{O}(n)$
- Space: n bytes

Direct computation, e.g.:

- Lippert et al. 2005:
  - Time: $\mathcal{O}(n \log n)$
  - Space: $\frac{1}{2}(1+\sigma)(1+\epsilon)$ bits
- Okanohara and Sadakane 2009:
  - Time: $\mathcal{O}(n)$
  - Space: $\mathcal{O}(n\, log\sigma\ log(log_\sigma\ n)) \approx 2.5n$ bytes

LCP array (Longest Common Prefix array)

i	$SA[i]$	$BWT[i]$	$S_{SA[i]}$
1	13	s	$
2	6	s	anannas$
3	8	n	annas$
4	1	$	annasanannas$
5	11	n	as$
6	4	n	asanannas$
7	7	a	nannas$
8	10	n	nas$
9	3	n	nasanannas$
10	9	a	nnas$
11	2	a	nnasanannas$
12	12	a	s$
13	5	a	sanannas$
14			

LCP array (Longest Common Prefix array)

i	$SA[i]$	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	13	s	−1	$
2	6	s	0	anannas$
3	8	n	2	annas$
4	1	$	5	annasanannas$
5	11	n	1	as$
6	4	n	2	asanannas$
7	7	a	0	nannas$
8	10	n	2	nas$
9	3	n	3	nasanannas$
10	9	a	1	nnas$
11	2	a	4	nnasanannas$
12	12	a	0	s$
13	5	a	1	sanannas$
14			−1	

LCP construction algorithms from suffix array

KLAAP-algorithm of Kasai et al. 2001:

- Time: $\mathcal{O}(n)$
- Space: $13n$ bytes
- Space improvement by Manzini 2004: $9n$ bytes

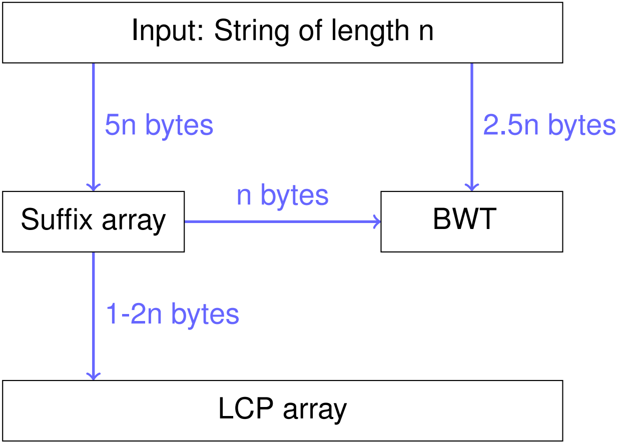
Φ-algorithm of Kärkkäinen et al. 2009:

- Time: $\mathcal{O}(n)$
- Space: $5n + \frac{4n}{k}$ bytes or $n + \frac{4n}{k}$ bytes (semi-external)

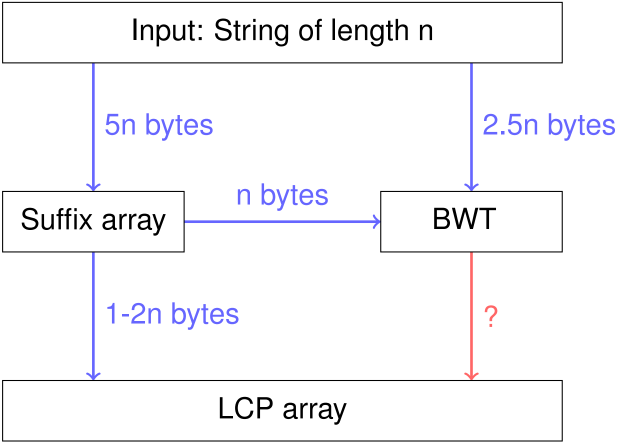
go-Φ-algorithm of Gog and Ohlebusch 2010:

- Time: $\mathcal{O}(n)$
- Space: $2n$ bytes

Overview


Input: String of length n
5n bytes
2.5n bytes
Suffix array
n bytes
BWT
1-2n bytes
LCP array

Task


Input: String of length n
5n bytes
2.5n bytes
Suffix array
n bytes
BWT
1-2n bytes
?
LCP array

Observation

Assume the string ω occurs t times in a string S:

- There are t suffixes of S that start with ω.
- These suffixes occur consecutively in the suffix array.
- Let j be the largest index, so that the corresponding suffix starts with ω.
- $LCP[j+1] < |\omega|$

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	2	annas$
4	$	5	annasanannas$
5	n	1	as$
6	n	2	asanannas$
7	a	0	nannas$
8	n	2	nas$
9	n	3	nasanannas$
10	a	1	nnas$
11	a	4	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	2	annas$
4	$	5	annasanannas$
5	n	1	as$
6	n	2	asanannas$
7	a	0	nannas$
8	n	2	nas$
9	n	3	nasanannas$
10	a	1	nnas$
11	a	4	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Idea

- Calculate all substrings of S, in the order of their length.
- Determine for each substring ω the corresponding interval $[lb \dots rb]$.
- If $LCP[rb + 1]$ wasn't set before, set $LCP[rb + 1] = |\omega| - 1$.

Pseudocode

$\text{LCP}[1] \leftarrow -1$
$\text{LCP}[i] \leftarrow \bot \quad \forall i : 2 \leq i \leq n$
$\text{LCP}[n+1] \leftarrow -1$
initialize an empty queue
enqueue(ϵ)
while not all lcp values are calculated **do**
  $\omega \leftarrow$ *dequeue*()
  for each $a \in \Sigma$ **do**
    enqueue($a\omega$)
    $[lb \ldots rb] \leftarrow$ getIntervalBounds($a\omega$)
    if $rb \neq \bot$ **and** $\text{LCP}[rb+1] = \bot$ **then**
      $\text{LCP}[rb+1] \leftarrow |a\omega| - 1$

Pseudocode

LCP[1] $\leftarrow -1$
LCP[i] $\leftarrow \perp \quad \forall i : 2 \leq i \leq n$
LCP[$n+1$] $\leftarrow -1$
initialize an empty queue
enqueue(ϵ)
while queue is not empty **do**
  $\omega \leftarrow$ *dequeue*()
  for each $a \in \Sigma$ **do**
    ~~*enqueue*($a\omega$)~~
    [$lb \ldots rb$] $\leftarrow$ getIntervalBounds($a\omega$)
    if $rb \neq \perp$ **and** LCP[$rb+1$] $= \perp$ **then**
      LCP[$rb+1$] $\leftarrow |a\omega| - 1$
      enqueue($a\omega$)

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	⊥	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	⊥	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	⊥	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	⊥	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	⊥	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	⊥	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	⊥	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	⊥	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	⊥	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	⊥	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	⊥	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	⊥	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	⊥	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	1	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	1	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	1	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	1	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	1	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Example: annasanannas$

i	$BWT[i]$	$LCP[i]$	$S_{SA[i]}$
1	s	−1	$
2	s	0	anannas$
3	n	⊥	annas$
4	$	⊥	annasanannas$
5	n	⊥	as$
6	n	⊥	asanannas$
7	a	0	nannas$
8	n	⊥	nas$
9	n	⊥	nasanannas$
10	a	1	nnas$
11	a	⊥	nnasanannas$
12	a	0	s$
13	a	1	sanannas$
14		−1	

Saving space

Store interval boundaries $[lb \dots rb]$ of ω not ω itself:

- Store two integers: $2 \log n$ bit for each substring.
- Mark lb and rb in two bit vectors B_{lb} and B_{rb} of length n: $2n$ bit for all substrings of same length.

Reserve only n byte for the LCP array:

- Use the fact, that algorithm calculates the LCP values in ascending order.
- If new LCP value cannot be stored into the LCP array, write LCP array to disk.

Calculation of the subintervals

Problem: Given the interval $[lb \ldots rb]$ of ω, if $a\omega$, $a \in \Sigma$ is a substring of S, then find the interval of $a\omega$.

Modified backward search with wavelet tree of the BWT:

- Find all subintervals by traversing the wavelet tree in a depth-first manner.
- Use Huffman-shaped wavelet trees to save time and space.
- Time: $\mathcal{O}(\sigma)$
- Space: n bytes

Runtime and space

Time complexity: $\mathcal{O}(\sigma\ n)$

Practical and space efficient implementation:

- Time: $\mathcal{O}(n\ \log n)$
- Space: $\approx 2.2n$ bytes

Experimental Results

Test cases

- Pizza&Chili Corpus
- Some DNA-files from www.ensembl.org (Release 62)

Implementation

- uses the *sdsl*-library of Simon Gog (www.uni-ulm.de/in/theo/research/sdsl.html)
- uses bit compressed arrays (i.e. log n bits, not 4 bytes or 8 bytes per integer)

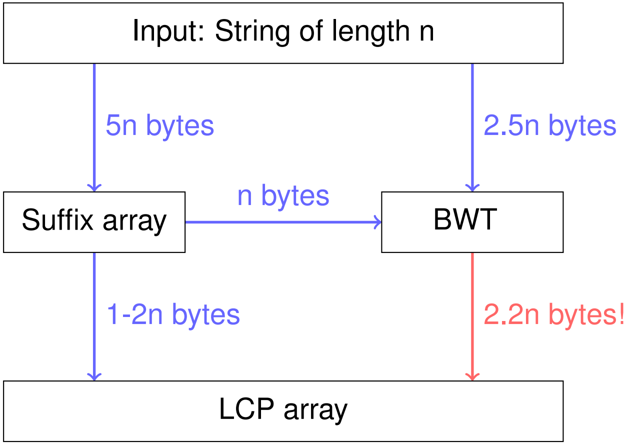
Experimental Results

	dna 200MB		english 200MB		proteins 200MB		sources 200MB		xml 200MB	
SA constr.	71	5	64	5	72	5	45	5	49	5
BWT constr.	93	1.9	109	2.2	150	2.6	87	2.2	83	2.2
KLAAP	58	9	48	9	48	9	33	9	32	9
Φ1	37	9	30	9	30	9	22	9	22	9
Φ4	83	6	74	6	78	6	60	6	63	6
Φ64	80	5.1	76	5.1	78	5.1	64	5.1	75	5.1
Φ4-Semi	78	2	72	2	72	2	59	2	63	2
Φ64-Semi	76	1.1	70	1.1	70	1.1	56	1.1	73	1.1
go-Φ	53	2	74	2	70	2	51	2	49	2
new algorithm	66	1.8	124	2	137	2	131	2.2	99	2.1
KLAAP	129	9	112	9	120	9	78	9	81	9
Φ1	108	9	94	9	102	9	67	9	71	9
Φ4	154	6	138	6	150	6	105	6	112	6
Φ64	151	5.1	140	5.1	150	5.1	109	5.1	124	5.1
Φ4-Semi	149	5	136	5	144	5	104	5	112	5
Φ64-Semi	147	5	134	5	142	5	101	5	122	5
go-Φ	124	5	138	5	142	5	96	5	98	5
new algorithm	159	1.9	233	2.2	287	2.6	218	2.2	182	2.2

Experimental Results

	Stickleback 446 MB		Chicken 1.050 MB		Sloth 2.060 MB		Orangutan 3.093 MB	
SA constr.	171	5	471	5	1.100	5	2.013	9
BWT constr.	204	2	549	1,9	1.062	1,9	1.686	1,9
KLAAP	150	9	454	9	951	9	1.527	9
Φ1	98	9	318	9	756	9	1.183	9
Φ4	187	6	534	6	1.236	6	-	-
Φ64	193	5,1	522	5,1	1.163	5,1	-	-
Φ4-Semi	182	2	523	2	1.183	2	1.786	2
Φ64-Semi	180	1,1	454	1,1	1.064	1,1	1.648	1,1
go-Φ	117	2	316	2	685	2	1.041	2
new algorithm	141	1,8	338	1,8	800	1,8	1.270	1,8
KLAAP	321	9	925	9	2.051	9	3.540	9
Φ1	269	9	789	9	1.856	9	3.196	9
Φ4	358	6	1.005	6	2.336	6	-	-
Φ64	364	5,1	993	5,1	2.263	5,1	-	-
Φ4-Semi	353	5	994	5	2.283	5	3.799	9
Φ64-Semi	351	5	925	5	2.164	5	3.661	9
go-Φ	288	5	787	5	1.785	5	3.054	9
new algorithm	345	2	887	1,9	1.862	1,9	2.956	1,9

Solution



Input: String of length n
5n bytes
2.5n bytes
Suffix array
n bytes
BWT
1-2n bytes
2.2n bytes!
LCP array

Thank you!
Any Questions?