

Linear Time Lempel-Ziv Factorization: Simple, Fast, Small

Juha Kärkkäinen Dominik Kempa Simon J. Puglisi

University of Helsinki, Finland

CPM 2013

Outline

1 Introduction

2 Existing solutions

3 $2n \log n$ algorithm

Example

1 2 3 4 5 6 7 8 9 10 11 12

X = b a b b a b a b b b a b

Example

1 2 3 4 5 6 7 8 9 10 11 12

X = b a b b a b a b b b a b

Example

1 2 3 4 5 6 7 8 9 10 11 12

$$X = b \boxed{a b b} a b \boxed{a b b} b a b$$

Example

$X = b \underset{p_7 = 2}{\overset{\uparrow}{a}} \underset{\ell_7 = 3}{\overset{\leftarrow \dots \rightarrow}{b}} b a b \underset{\ell_7 = 3}{\overset{\leftarrow \dots \rightarrow}{a}} b b b a b a b$

Example

1 2 3 4 5 6 7 8 9 10 11 12

X = b a b b a b a b b b a b

Example

1 2 3 4 5 6 7 8 9 10 11 12

$X = (\mathbf{b} \mathbf{a} \mathbf{b}) \mathbf{b} \mathbf{a} \mathbf{b} \mathbf{a} \mathbf{b} \mathbf{b} (\mathbf{b} \mathbf{a} \mathbf{b})$

\uparrow $\leftarrow \dots \rightarrow$

$p_{10} = 1$ $\ell_{10} = 3$

Example

$$X = b a b \boxed{b a b} a b b \boxed{b a b}$$

\uparrow $\leftarrow \dots \rightarrow$

$$p_{10} = 4 \qquad \qquad \ell_{10} = 3$$

Example

1 2 3 4 5 6 7 8 9 10 11 12

$X = b a b b a b [b a b] b [b a b]$

\uparrow

$p_{10} = 6$

$\leftarrow \dots \rightarrow$

$\ell_{10} = 3$

Example

$$X = b \underset{\uparrow}{a} b \underset{p_{10}=6}{a} b \underset{\ell_{10}=3}{a} b$$

Definition

Pairs (p_i, ℓ_i) define the LPF array: $\text{LPF}[i] = (p_i, \ell_i)$.

LPF array

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12
	p_i	ℓ_i	b a b b a b a b b a b
1	⊥	0	b a b b a b a b b a b
2	⊥	0	b a b b a b a b b a b
3	1	1	b a b a b a b a b b a b
4	1	3	b a b b a b a b a b b a b
5	2	2	b a b b a b a b a b b a b
6	1	4	b a b b a b a b a b b a b
7	2	3	b a b a b a b a b b a b
8	3	2	b a b b a b a b a b b a b
9	3	4	b a b b a b a b a b b a b
10	4	3	b a b b a b a b a b b a b
11	5	2	b a b b a b a b a b b a b
12	6	1	b a b b a b a b b a b b

Lempel-Ziv Factorization

i	LPF		1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i	b	a	b	a	b	a	b	b	a	b	b	a
1	\perp	0	b	a	b	a	b	a	b	b	a	b	b	a
2	\perp	0	b	a	b	a	b	a	b	b	a	b	b	a
3	1	1	b	a	b	a	b	a	b	b	a	b	b	a
4	1	3	b	a	b	a	b	a	b	b	b	a	b	a
5	2	2	b	a	b	a	b	a	b	b	b	a	b	a
6	1	4	b	a	b	a	b	a	b	b	b	a	b	a
7	2	3	b	a	b	a	b	a	b	b	b	a	b	a
8	3	2	b	a	b	a	b	a	b	b	b	a	b	a
9	3	4	b	a	b	a	b	a	b	b	a	b	b	a
10	4	3	b	a	b	a	b	a	b	b	b	a	b	a
11	5	2	b	a	b	a	b	a	b	b	b	a	b	a
12	6	1	b	a	b	a	b	a	b	b	b	a	b	a

LZ77:

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12										
	p_i	ℓ_i	b a b b a b a b b b a b										
1	\perp	0	b a b b a b a b b b a b										
2	\perp	0	b a b b a b a b b b a b										
3	1	1	b a b b a b a b a b b b a b										
4	1	3	b a b b a b a b a b b b a b										
5	2	2	b a b b a b a b a b b b a b										
6	1	4	b a b b a b a b a b b a b										
7	2	3	b a b b a b a b a b b a b										
8	3	2	b a b b a b a b a b b a b										
9	3	4	b a b b a b a b a b b a b										
10	4	3	b a b b a b a b a b b a b										
11	5	2	b a b b a b a b a b b a b										
12	6	1	b a b b a b a b a b b a b										

LZ77:

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12										
	p_i	ℓ_i	(b) a b b a b a b b b a b										
1	⊥	0	b a b b a b a b b b a b										
2	⊥	0	b a b b a b a b b b a b										
3	1	1	b a b a b a b a b b b a b										
4	1	3	b a b a b a b a b b b a b										
5	2	2	b a b a b a b a b b b a b										
6	1	4	b a b a b a b a b b a b										
7	2	3	b a b a b a b a b b a b										
8	3	2	b a b a b a b a b b a b										
9	3	4	b a b a b a b a b b a b										
10	4	3	b a b a b a b a b b a b										
11	5	2	b a b a b a b a b b a b										
12	6	1	b a b a b a b a b b a b										

LZ77: (b,0)

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12										
	p_i	ℓ_i	b	a	b	b	a	b	b	b	a	b	
1	\perp	0	b	a	b	b	a	b	b	b	a	b	
2	\perp	0	b	a	b	b	a	b	b	b	a	b	
3	1	1	b	a	b	b	a	b	b	b	a	b	
4	1	3	b	a	b	b	a	b	b	b	a	b	
5	2	2	b	a	b	b	a	b	b	b	a	b	
6	1	4	b	a	b	a	b	a	b	b	a	b	
7	2	3	b	a	b	a	b	a	b	b	a	b	
8	3	2	b	a	b	a	b	a	b	b	a	b	
9	3	4	b	a	b	a	b	a	b	b	a	b	
10	4	3	b	a	b	a	b	a	b	b	a	b	
11	5	2	b	a	b	a	b	a	b	b	a	b	
12	6	1	b	a	b	a	b	a	b	b	a	b	

LZ77: (b,0)

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12										
	p_i	ℓ_i	b	a	b	b	a	b	b	b	a	b	
1	\perp	0	b	a	b	b	a	b	b	b	a	b	
2	\perp	0	b	a	b	b	a	b	b	b	a	b	
3	1	1	b	a	b	b	a	b	b	b	a	b	
4	1	3	b	a	b	b	a	b	b	b	a	b	
5	2	2	b	a	b	b	a	b	b	b	a	b	
6	1	4	b	a	b	a	b	a	b	b	a	b	
7	2	3	b	a	b	a	b	a	b	b	a	b	
8	3	2	b	a	b	a	b	a	b	b	a	b	
9	3	4	b	a	b	a	b	a	b	b	a	b	
10	4	3	b	a	b	a	b	a	b	b	a	b	
11	5	2	b	a	b	a	b	a	b	b	a	b	
12	6	1	b	a	b	a	b	a	b	b	a	b	

LZ77: (b,0),(a,0)

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12
	p_i	ℓ_i	(b)(a)b b a b a b b b a b
1	\perp	0	b a b b a b a b b b a b
2	\perp	0	b a b b a b a b b b a b
3	1	1	(b)a(b)b a b a b b b a b
4	1	3	(b)a(b)b a(b)a b b b a b
5	2	2	b(a)b(a)b a(b)a b b b a b
6	1	4	(b)a(b)a(b)a(b)b a b
7	2	3	b(a)b a(b)a(b)b a b
8	3	2	b a(b)a(b)a(b)b a b
9	3	4	b a(b)a(b)a b(b)a b
10	4	3	b a b(b)a b a b(b)a b
11	5	2	b a b(b)a b a b b a b
12	6	1	b a b b a(b)a b b b a(b)

LZ77: (b,0),(a,0)

Lempel-Ziv Factorization

i	LPF		1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i	(b)	a	b	a	b	a	b	b	b	a	b	a
1	\perp	0	b	a	b	a	b	a	b	b	b	a	b	a
2	\perp	0	b	a	b	a	b	a	b	b	b	a	b	a
3	1	1	b	a	b	a	b	a	b	b	b	a	b	a
4	1	3	b	a	b	a	b	a	b	b	b	a	b	a
5	2	2	b	a	b	a	b	a	b	b	b	a	b	a
6	1	4	b	a	b	a	b	a	b	b	b	a	b	a
7	2	3	b	a	b	a	b	a	b	b	b	a	b	a
8	3	2	b	a	b	a	b	a	b	b	b	a	b	a
9	3	4	b	a	b	a	b	a	b	b	b	a	b	a
10	4	3	b	a	b	a	b	a	b	b	b	a	b	a
11	5	2	b	a	b	a	b	a	b	b	b	a	b	a
12	6	1	b	a	b	a	b	a	b	b	b	a	b	a

LZ77: (b,0),(a,0),(1,1)

Lempel-Ziv Factorization

i	LPF		1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i	b	a	b	a	b	a	b	a	b	b	b	a
1	\perp	0	b	a	b	a	b	a	b	b	b	b	b	b
2	\perp	0	b	a	b	a	b	a	b	b	b	b	b	b
3	1	1	b		a		b	a	b	a	b	b	b	b
4	1	3	b		a		b		b	a	b	b	b	b
5	2	2	b		a		b		a	b	b	b	b	b
6	1	4	b		a		b		a		b	b	b	b
7	2	3	b		a		b		a		b	b	b	b
8	3	2	b		a		b		a		b		b	b
9	3	4	b		a		b		a		b		b	b
10	4	3	b		a		b		a		b		b	b
11	5	2	b		a		b		a		b		a	b
12	6	1	b		a		b		a		b		b	

LZ77: (b,0),(a,0),(1,1)

Lempel-Ziv Factorization

i	LPF		1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i	b	a	b	a	b	a	b	a	b	b	b	a
1	\perp	0	b	a	b	a	b	a	b	b	b	b	a	b
2	\perp	0	b	a	b	a	b	a	b	b	b	b	a	b
3	1	1	b		a		b	a	b	a	b	b	b	a
4	1	3	b		a		b		b	a	b	b	b	a
5	2	2	b		a		b		a	b	b	b	b	a
6	1	4	b		a		b		a		b	b	b	a
7	2	3	b		a		b		a		b	a	b	b
8	3	2	b		a		b		a		b		b	a
9	3	4	b		a		b		a		b		b	a
10	4	3	b		a		b		a		b		b	a
11	5	2	b		a		b		a		b		a	b
12	6	1	b		a		b		a		b		b	

LZ77: (b,0),(a,0),(1,1),(1,3)

Lempel-Ziv Factorization

i	LPF		1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i	b	a	b	b	a	b	a	b	b	a	b	b
1	\perp	0	b	a	b	b	a	b	a	b	b	b	a	b
2	\perp	0	b	a	b	b	a	b	b	b	b	a	b	b
3	1	1	b	a	b	b	a	b	a	b	b	b	a	b
4	1	3	b	a	b	a	b	a	b	b	b	b	a	b
5	2	2	b	a	b	a	b	a	b	b	b	b	a	b
6	1	4	b	a	b	a	b	a	b	b	b	a	b	b
7	2	3	b	a	b	a	b	a	b	b	b	a	b	b
8	3	2	b	a	b	a	b	a	b	b	b	a	b	b
9	3	4	b	a	b	a	b	a	b	b	b	a	b	b
10	4	3	b	a	b	a	b	a	b	b	b	b	a	b
11	5	2	b	a	b	a	b	a	b	b	b	b	a	b
12	6	1	b	a	b	a	b	a	b	b	b	a	b	b

LZ77: (b,0),(a,0),(1,1),(1,3)

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12
	p_i	ℓ_i	(b)(a)(b)(b a b)(a b b)(a b a b)
1	⊥	0	b a b b a b a b b b a b
2	⊥	0	b a b b a b a b b b a b
3	1	1	(b)a(b)b a b a b b b a b
4	1	3	(b a b)b a b a b b b a b
5	2	2	b(a b)b a(b)a b b b a b
6	1	4	(b a b)a(b)a b b b a b
7	2	3	b(a b)a b(a b)b a b
8	3	2	b a(b)a b a(b)b a b
9	3	4	b a(b)a b a b(b)a b
10	4	3	b a b(b)a b a b(b)a b
11	5	2	b a b(b)a b a b b a b
12	6	1	b a b b a(b)a b b b a(b)

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3)

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12											
	p_i	ℓ_i	b	a	b	a	b	a	b	b	a	b	a	b
1	\perp	0	b	a	b	a	b	a	b	b	a	b	a	b
2	\perp	0	b	a	b	a	b	a	b	b	a	b	a	b
3	1	1	b	a	b	a	b	a	b	b	a	b	a	b
4	1	3	b	a	b	a	b	a	b	b	a	b	a	b
5	2	2	b	a	b	a	b	a	b	b	a	b	a	b
6	1	4	b	a	b	a	b	a	b	b	a	b	a	b
7	2	3	b	a	b	a	b	a	b	b	a	b	a	b
8	3	2	b	a	b	a	b	a	b	b	a	b	a	b
9	3	4	b	a	b	a	b	a	b	b	a	b	a	b
10	4	3	b	a	b	a	b	a	b	b	a	b	a	b
11	5	2	b	a	b	a	b	a	b	b	a	b	a	b
12	6	1	b	a	b	a	b	a	b	b	a	b	a	b

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3)

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12
	p_i	ℓ_i	(b)(a)(b)(b a b)(a b b)(b a b)
1	\perp	0	b a b b a b a b b b a b
2	\perp	0	b a b b a b a b b b a b
3	1	1	(b)a(b)b a b a b b b a b
4	1	3	(b a b)b a b a b b b a b
5	2	2	b(a b)b a b a b b b a b
6	1	4	(b a b)a(b a b)b a b
7	2	3	b(a b)a b(a b)b a b
8	3	2	b a(b)a b a(b)b a b
9	3	4	b a(b)a b a(b)b a(b)
10	4	3	b a b(b a b)a b(b)a b
11	5	2	b a b(b a b)a b b b(a b)
12	6	1	b a b b a(b)a b b b a(b)

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3),(4,3)

Lempel-Ziv Factorization

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12
	p_i	ℓ_i	(b)(a)(b)(b a b)(a b b)(b a b)
1	\perp	0	b a b b a b a b b b a b
2	\perp	0	b a b b a b a b b b a b
3	1	1	(b)a(b)b a b a b b b a b
4	1	3	(b a b)b a b a b b b a b
5	2	2	b(a b)b a b a b b b a b
6	1	4	(b a b)a(b a b)b a b
7	2	3	b(a b)a b(a b)b a b
8	3	2	b a(b)a b a(b)b a b
9	3	4	b a(b)a b a b(b)a b
10	4	3	b a b(b)a b a b(b)a b
11	5	2	b a b(b)a b a b b a b
12	6	1	b a b b a(b)a b b b a(b)

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3),(4,3)

Outline

1 Introduction

2 Existing solutions

3 $2n \log n$ algorithm

Existing solutions

- Space excludes the input and output (both of size $n \log \sigma$).

Algorithm	Extra space
Abouelhoda et al., 2004	$4n \log n$
Chen et al. (CPS1), 2007	$3n \log n$
Crochemore and Ilie, 2008	$(3n + \sqrt{n}) \log n$
Ohlebusch and Gog, 2011	$3n \log n$
Goto and Bannai (BGL), 2013	$3n \log n$

Our contribution

Two new linear time algorithms:

Our contribution

Two new linear time algorithms:

- ① K3: $3n \log n$ bits of extra space

Our contribution

Two new linear time algorithms:

- ① K3: $3n \log n$ bits of extra space
 - minimizes the number of cache misses

Our contribution

Two new linear time algorithms:

- ① K3: $3n \log n$ bits of extra space
 - minimizes the number of cache misses
 - fastest, when the input is not highly repetitive

Our contribution

Two new linear time algorithms:

① K3: $3n \log n$ bits of extra space

- minimizes the number of cache misses
- fastest, when the input is not highly repetitive

Algorithm K3

```
1: SA[0] ← SA[n + 1] ← top ← 0
2: for  $i \leftarrow 1$  to  $n + 1$  do
3:   while SA[top] > SA[i] do
4:     NSV[SA[top]] ← SA[i]
5:     PSV[SA[top]] ← SA[top - 1]
6:     top ← top - 1
7:   top ← top + 1
8:   SA[top] ← SA[i]
9:    $i \leftarrow 1$ 
10: while  $i \leq n$  do
11:    $i \leftarrow$  LZ-Factor( $i$ , NSV[i], PSV[i])
```

Procedure LZ-Factor(i, nsv, psv)

```
1:  $\ell_{nsv} \leftarrow \text{lcp}(i, nsv)$ 
2:  $\ell_{psv} \leftarrow \text{lcp}(i, psv)$ 
3: if  $\ell_{nsv} > \ell_{psv}$  then
4:    $(p, \ell) \leftarrow (nsv, \ell_{nsv})$ 
5: else
6:    $(p, \ell) \leftarrow (psv, \ell_{psv})$ 
7: if  $\ell = 0$  then  $p \leftarrow X[i]$ 
8: output factor ( $p, \ell$ )
9: return  $i + \max(\ell, 1)$ 
```

Our contribution

Two new linear time algorithms:

- ① K3: $3n \log n$ bits of extra space
 - minimizes the number of cache misses
 - fastest, when the input is not highly repetitive
- ② K2: $2n \log n$ bits of extra space

Our contribution

Two new linear time algorithms:

- ① K3: $3n \log n$ bits of extra space
 - minimizes the number of cache misses
 - fastest, when the input is not highly repetitive
- ② K2: $2n \log n$ bits of extra space
 - most space efficient linear algorithm for LZ77

Our contribution

Two new linear time algorithms:

- ① K3: $3n \log n$ bits of extra space
 - minimizes the number of cache misses
 - fastest, when the input is not highly repetitive
- ② K2: $2n \log n$ bits of extra space
 - most space efficient linear algorithm for LZ77
 - based on combinatorics of suffix arrays

Outline

1 Introduction

2 Existing solutions

3 $2n \log n$ algorithm

$2n \log n$ algorithm

i	LPF		1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i	b	a	b	b	a	b	a	b	b	b	a	b
1	\perp	0	b	a	b	b	a	b	a	b	b	b	a	b
2	\perp	0	b	a	b	b	a	b	b	b	a	b	b	a
3	1	1	b	a	b	b	a	b	a	b	b	a	b	b
4	1	3	b	a	b	b	a	b	b	b	a	b	b	a
5	2	2	b	a	b	b	a	b	b	b	a	b	b	a
6	1	4	b	a	b	b	a	b	a	b	b	a	b	b
7	2	3	b	a	b	b	a	b	a	b	b	a	b	b
8	3	2	b	a	b	b	a	b	a	b	b	a	b	b
9	3	4	b	a	b	b	a	b	a	b	b	a	b	b
10	4	3	b	a	b	b	a	b	a	b	b	a	b	b
11	5	2	b	a	b	b	a	b	a	b	b	a	b	b
12	6	1	b	a	b	b	a	b	a	b	b	a	b	b

LZ77:

$2n \log n$ algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12											
	p_i	ℓ_i	b	a	b	b	a	b	a	b	b	b	a	b
1	\perp	0	b	a	b	b	a	b	a	b	b	b	a	b
2	\perp	0	b	a	b	b	a	b	a	b	b	b	a	b
3	1	1	b	a	b	b	a	b	a	b	b	b	a	b
4	1	3	b	a	b	b	a	b	a	b	b	b	a	b
5	2	2	b	a	b	b	a	b	a	b	b	b	a	b
6	1	4	b	a	b	b	a	b	a	b	b	b	a	b
7	2	3	b	a	b	b	a	b	a	b	b	b	a	b
8	3	2	b	a	b	b	a	b	a	b	b	b	a	b
9	3	4	b	a	b	b	a	b	a	b	b	b	a	b
10	4	3	b	a	b	b	a	b	a	b	b	b	a	b
11	5	2	b	a	b	b	a	b	a	b	b	b	a	b
12	6	1	b	a	b	b	a	b	a	b	b	b	a	b

LZ77:

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	LPF			1 2 3 4 5 6 7 8 9 10 11 12											
	p_i	ℓ_i		b	a	b	b	a	b	a	b	b	b	a	b
1	⊥			b	a	b	b	a	b	a	b	b	b	a	b
2	⊥			b	a	b	b	a	b	a	b	b	b	a	b
3	1			b	a	(b)	b	a	b	a	b	b	b	a	b
4	1			b	a	b	(b)	a	b	a	b	b	b	a	b
5	2			b	(a)	b	b	a	b	a	b	b	b	a	b
6	1			b	a	b	b	a	(b)	a	b	b	b	a	b
7	2			b	(a)	b	a	b	a	(a)	b	b	b	a	b
8	3			b	a	b	a	b	a	b	(b)	b	b	a	b
9	3			b	a	b	(b)	a	b	a	b	(b)	b	a	b
10	4			b	a	b	b	a	b	a	b	b	(b)	a	b
11	5			b	a	b	b	a	b	a	b	b	(b)	a	b
12	6			b	a	b	b	a	b	a	b	b	b	a	b

LZ77:

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	LPF		1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i	b	a	b	a	b	a	b	b	b	a	b	
1	⊥		b	a	b	a	b	b	b	a	b	b	a	b
2	⊥		b	a	b	a	b	b	b	b	a	b	b	a
3	1		b	a	b	a	b	a	b	b	b	a	b	b
4	1		b	a	b	a	b	a	b	b	b	a	b	b
5	2		b	a	b	a	b	a	b	b	b	a	b	b
6	1		b	a	b	a	b	a	b	b	b	a	b	b
7	2		b	a	b	a	b	a	b	b	b	a	b	b
8	3		b	a	b	a	b	a	b	b	b	a	b	b
9	3		b	a	b	a	b	a	b	b	b	a	b	b
10	4		b	a	b	a	b	a	b	b	b	a	b	b
11	5		b	a	b	a	b	a	b	b	b	a	b	b
12	6		b	a	b	a	b	a	b	b	b	a	b	b

LZ77: (b,0),(a,0),(1,1),(1,3)

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	p_i	ℓ_i	LPF	1 2 3 4 5 6 \$ 8 9 10 11 12
1	⊥		b	a b b a b a b b b a b
2	⊥		b	a b b a b a b b b a b
3	1		(b)	a (b) b a b a b b b a b
4	1		(b)	a b (b) a b a b b b a b
5	2		b (a)	a b (a) b a b b b a b
6	1		(b)	a b b a b (b) a b b b a b
7	2		b (a)	a b (a) b a b b b a b
8	3		b (b)	a b (b) b a b b b a b
9	3		b (b)	a b (b) a b b b (b) a b
10	4		b (b)	a b b (b) a b b b (b) a b
11	5		b (a)	a b b (a) b a b b b (a) b
12	6		b (b)	a b b b (b) a b b b (b) a b

LZ77: (b,0),(a,0),(1,1),(1,3)

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	LPF														
	p_i	ℓ_i		1	2	3	4	5	6	7	8	9	10	11	12
1	⊥			b	a	b	a	b	a	b	b	b	b	b	b
2	⊥			b	a	b	a	b	a	b	b	b	b	b	b
3	1			b	a	b	a	b	a	b	b	b	b	b	b
4	1			b	a	b	a	b	a	b	b	b	b	b	b
5	2			b	a	b	a	b	a	b	b	b	b	b	b
6	1			b	a	b	a	b	a	b	b	b	b	b	b
7	2			b	a	b	a	b	a	b	b	b	b	b	b
8	3			b	a	b	a	b	a	b	b	b	b	b	b
9	3			b	a	b	a	b	a	b	b	b	b	b	b
10	4			b	a	b	a	b	a	b	b	b	b	b	b
11	5			b	a	b	a	b	a	b	b	b	b	b	b
12	6			b	a	b	a	b	a	b	b	b	b	b	b

LZ77: (b,0),(a,0),(1,1),(1,3)

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	LPF			Text											
	p_i	ℓ_i	Index	1	2	3	4	5	6	7	8	9	10	11	12
1	⊥		b	a	b	a	b	a	b	a	b	b	b	b	b
2	⊥		b	a	b	a	b	a	b	a	b	b	b	b	b
3	1		b	a	b	a	b	a	b	a	b	b	b	b	b
4	1		b	a	b	a	b	a	b	a	b	b	b	b	b
5	2		b	a	b	a	b	a	b	a	b	b	b	b	b
6	1		b	a	b	a	b	a	b	a	b	b	b	b	b
7	2		b	a	b	a	b	a	b	a	b	b	b	b	b
8	3		b	a	b	a	b	a	b	a	b	b	b	b	b
9	3		b	a	b	a	b	a	b	a	b	b	b	b	b
10	4		b	a	b	a	b	a	b	a	b	b	b	b	b
11	5		b	a	b	a	b	a	b	a	b	b	b	a	b
12	6		b	a	b	a	b	a	b	a	b	b	b	b	b

LZ77: (b,0),(a,0),(1,1),(1,3)

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	LPF													
	p_i	ℓ_i	1	2	3	4	5	6	\$	\$	\$	10	11	12
1	⊥		b	a	b	a	b	a	b	b	b	b	b	b
2	⊥		b	a	b	a	b	a	b	b	b	b	b	b
3	1		b	a	b	a	b	a	b	b	b	b	b	b
4	1		b	a	b	a	b	a	b	b	b	b	b	b
5	2		b	a	b	a	b	a	b	b	b	b	b	b
6	1		b	a	b	a	b	a	b	b	b	b	b	b
7	2	3	b	a	b	a	b	a	b	b	b	b	b	b
8	3		b	a	b	a	b	a	b	b	b	b	b	b
9	3		b	a	b	a	b	a	b	b	b	b	b	b
10	4		b	a	b	a	b	a	b	b	b	b	b	b
11	5		b	a	b	a	b	a	b	b	b	b	b	b
12	6		b	a	b	a	b	a	b	b	b	b	b	b

LZ77: (b,0),(a,0),(1,1),(1,3)

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing

i	LPF														
	p_i	ℓ_i		1	2	3	4	5	6	7	8	9	10	11	12
1	⊥			b	a	b	a	b	a	b	b	b	\$	\$	\$
2	⊥			b	a	b	a	b	a	b	b	b	b	b	b
3	1			b	a	(b)	b	a	b	a	b	b	b	b	b
4	1			b	a	b	(b)	a	b	a	b	b	b	a	b
5	2			b	a	b	b	(a)	b	a	b	b	b	a	b
6	1			b	a	b	b	a	(b)	a	b	b	b	a	b
7	2	3		b	a	b	a	b	a	b	(a)	b	b	a	b
8	3			b	a	b	a	b	a	b	(b)	b	b	a	b
9	3			b	a	b	a	b	a	b	b	(b)	b	a	b
10	4			b	a	b	a	b	a	b	b	a	b	(b)	a
11	5			b	a	b	b	a	b	a	b	b	b	(a)	b
12	6			b	a	b	b	a	b	a	b	b	b	a	b

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3)

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing **in linear time**.

i	LPF			1	2	3	4	5	6	7	8	9	10	11	12
	p_i	ℓ_i		b	a	b	a	b	a	b	b	a	b	b	a
1	1			b	a	b	a	b	a	b	b	a	b	b	a
2	1			b	a	b	a	b	a	b	b	a	b	b	a
3	2			b	a	b	a	b	a	b	b	a	b	b	a
4	2			b	a	b	a	b	a	b	b	a	b	b	a
5	3			b	a	b	a	b	a	b	b	a	b	b	a
6	3			b	a	b	a	b	a	b	b	a	b	b	a
7	4	3		b	a	b	a	b	a	b	b	a	b	b	a
8	4			b	a	b	a	b	a	b	b	a	b	b	a
9	5			b	a	b	a	b	a	b	b	a	b	b	a
10	5			b	a	b	a	b	a	b	b	a	b	b	a
11	6			b	a	b	a	b	a	b	b	a	b	b	a
12	6			b	a	b	a	b	a	b	b	a	b	b	a

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3)

2n log n algorithm

Observation

The p_i component of LPF array is enough to compute LZ77 parsing **in linear time**.

Goal

Space efficient computation of all p_i values.

i	p_i	ℓ_i	LPF	1 2 3 4 5 6 \$ \$ \$ 10 11 12
1	⊥		b a b b a b a b b a b	
2	⊥		b a b b a b a b b a b	
3	1		(b) a (b) b a b a b b a b	
4	1		(b) a b (b) a b a b b b a b	
5	2		b (a) b (a) b a b a b b a b	
6	1		(b) a b b a (b) a b a b b a b	
7	2	3	b a b b a b (a b b) b a b	
8	3		b a b (b) a b a b (b) b b a b	
9	3		b a b (b) a b a b (b) b a b	
10	4		b a b (b) a b a b b (b) a b	
11	5		b a b b (a) b a b a b b (a) b	
12	6		b a b b b (b) a b b b a (b)	

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3)

2n log n algorithm

Goal

Space efficient computation
of all p_i values.

$2n \log n$ algorithm

Goal

Space efficient computation
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$

SA	
10	a a b
11	a b
5	a b a b b a a b
7	a b b a a b
2	a b b a b a b b a a b
12	b
9	b a a b
4	b a b a b b a a b
6	b a b b a a b
1	b a b b a b a b b a a b
8	b b a a b
3	b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient computation
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$
- ② Locate in SA closest
smaller elements

SA	
10	a a b
11	a b
5	a b a b b a a b
7	a b b a a b
2	a b b a b a b b a a b
12	b
9	b a a b
4	b a b a b b a a b
6	b a b b a a b
1	b a b b a b a b b a a b
8	b b a a b
3	b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient computation
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$
- ② Locate in SA closest
smaller elements

SA
10 a a b
11 a b
5 a b a b b a a b
7 a b b a a b
2 a b b a b a b b a a b
12 b
9 b a a b
4 b a b a b b a a b
6 b a b b a a b
1 b a b b a b b b a a b
8 b b a a b
3 b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient computation
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$
- ② Locate in SA closest
smaller elements

SA	
10	a a b
11	a b
5	a b a b b a a b
7	a b b a a b
PSV[4] = 2	a b b a b a b b a a b
12	b
9	b a a b
4	b a b a b b a a b
6	b a b b a a b
NSV[4] = 1	b a b b a b a b b a a b
8	b b a a b
3	b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient computation
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$
- ② Locate in SA closest
smaller elements

Lemma [Crochemore, Ilie]

Either $\text{PSV}[i]$ or $\text{NSV}[i]$ is a
valid choice for p_i

SA	
10	a a b
11	a b
5	a b a b b a a b
7	a b b a a b
PSV[4] = 2	a b b a b a b b a a b
12	b
9	b a a b
4	b a b a b b a a b
6	b a b b a a b
NSV[4] = 1	b a b b a b a b b a a b
8	b b a a b
3	b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient computation
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$
- ② Locate in SA closest
smaller elements

Lemma [Crochemore, Ilie]

Either $\text{PSV}[i]$ or $\text{NSV}[i]$ is a
valid choice for p_i

Not quite what we wanted...

SA	
10	a a b
11	a b
5	a b a b b a a b
7	a b b a a b
PSV[4] = 2	a b b a b a b b a a b
12	b
9	b a a b
4	b a b a b b a a b
6	b a b b a a b
NSV[4] = 1	b a b b a b a b b a a b
8	b b a a b
3	b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient computation
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$
- ② Locate in SA closest
smaller elements

Lemma [Crochemore, Ilie]

Either $\text{PSV}[i]$ or $\text{NSV}[i]$ is a
valid choice for p_i

Not quite what we wanted...
...but sufficient for LZ77.

SA	
10	a a b
11	a b
5	a b a b b a a b
7	a b b a a b
PSV[4] = 2	a b b a b a b b a a b
12	b
9	b a a b
4	b a b a b b a a b
6	b a b b a a b
NSV[4] = 1	b a b b a b a b b a a b
8	b b a a b
3	b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

- ① Consider text position i ,
e.g., let $i = 4$
- ② Locate in SA closest
smaller elements

Lemma [Crochemore, Ilie]

Either $\text{PSV}[i]$ or $\text{NSV}[i]$ is a
valid choice for p_i

Not quite what we wanted...
...but sufficient for LZ77.

SA	
10	a a b
11	a b
5	a b a b b a a b
7	a b b a a b
PSV[4] = 2	a b b a b a b b a a b
12	b
9	b a a b
4	b a b a b b a a b
6	b a b b a a b
NSV[4] = 1	b a b b a b a b b a a b
8	b b a a b
3	b b a b a b b a a b

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	LPF		1 2 3 4 5 6 7 8 9 10 11 12											
	p_i	ℓ_i	b	a	b	b	a	b	a	b	b	b	a	b
1	\perp		b	a	b	b	a	b	a	b	b	b	a	b
2	\perp		b	a	b	b	a	b	a	b	b	b	a	b
3	1		b	a	b	b	a	b	a	b	b	b	a	b
4	1		b	a	b	b	a	b	a	b	b	b	a	b
5	2		b	a	b	b	a	b	a	b	b	b	a	b
6	1		b	a	b	b	a	b	a	b	b	b	a	b
7	2		b	a	b	b	a	b	a	b	b	b	a	b
8	3		b	a	b	b	a	b	a	b	b	b	a	b
9	3		b	a	b	b	a	b	a	b	b	b	a	b
10	4		b	a	b	b	a	b	a	b	b	b	a	b
11	5		b	a	b	b	a	b	a	b	b	b	a	b
12	6		b	a	b	b	a	b	a	b	b	b	a	b

LZ77:

2n log n algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 7 8 9 10 11 12
1	\perp	\perp	b a b b a b a b b b a b
2	\perp	1	b a b b a b a b b b a b
3	1	\perp	(b) a b (b) a b a b b b a b
4	2	1	(b) a b (b) a b a b b b a b
5	\perp	2	b (a) b (a) b a b a b b b a b
6	1	3	(b) a b (b) a b (b) a b a b b b a b
7	2	4	b (a) b (b) a b (a) b a b b b a b
8	3	\perp	b a (b) b a b a b (b) b b a b
9	4	3	b a (b) b a b a b (b) b a b
10	7	4	b a b (b) a b a b (a) b a b
11	10	5	b a b b (a) b a b b (b) a b
12	7	4	b a b b (b) a b a b b b a b

LZ77:

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 7 8 9 10 11 12
1			b
2			a
3			b
4			b
5			a
6			b
7			b
8			a
9			b
10			b
11			a
12			b

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 7 8 9 10 11 12
1			b
2			a
3			b
4			b
5			a
6			b
7	2	5	b a b b a b a b b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 7 8 9 10 11 12
1			b a b b a b a b b b a b
2			
3			
4			
5			
6			
7	2	5	b [a] b b [a] b [a] b b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 \$ 8 9 10 11 12
1			b a b b a b a b b a b
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b a b
8			
9			
10			
11			
12			

LZ77: $(b,0), (a,0), (1,1), (1,3)$

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 \$ \$ 9 10 11 12
1			b a b b a b a b b b a b
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 \$ \$ \$ 10 11 12
1			b a b b a b a b b b a b
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 \$ \$ \$ 10 11 12
1			b a b b a b a b b b a b
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6 \$ \$ \$ 10 11 12
1			b a b b a b a b b b a b
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6	10 11 12
1			b	
2			a	
3			b	
4			b	
5			a	
6			b	
7	2	5	b a b b a b	b b a b
8				
9				
10				
11				
12				

LZ77: $(b,0),(a,0),(1,1),(1,3)$

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6	10 11 12
1			b	\$
2			a	\$
3			b	\$
4			b	\$
5			a	\$
6			b	\$
7	2	5	b a b b a b	b b a b
8				≠
9				
10				
11				
12				

LZ77: (b,0),(a,0),(1,1),(1,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	1 2 3 4 5 6  10 11 12
1			b a b b a b a b b a b
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b a b
8			
9			
10			
11			
12			

LZ77: $(b,0),(a,0),(1,1),(1,3)$

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	
1			
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation**
of all p_i values.

i	PSV	NSV	
1			
2			
3			
4			
5			
6			
7	2	5	b a b b a b a b b b a b
8			
9			
10			
11			
12			

LZ77: (b,0),(a,0),(1,1),(1,3),(2,3)

$2n \log n$ algorithm

Goal

Space efficient **simulation** of all p_i values.

Fact

Given PSV and NSV arrays, LZ77 parsing can be computed in linear time.

i	PSV	NSV	1 2 3 4 5 6 10 11 12
1			b
2			a
3			b
4			b
5			a
6			b
7	2	5	b a b b a b a b b b a b
8			
9			
10			
11			
12			

LZ77: $(b,0), (a,0), (1,1), (1,3), (2,3)$

Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time

Computing NSV/PSV arrays

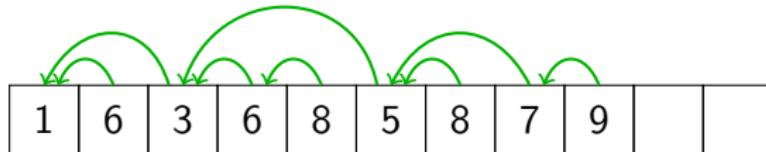
- PSV/NSV can be computed from SA in linear time
- no extra space required

Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time
- no extra space required
- computation of PSV:

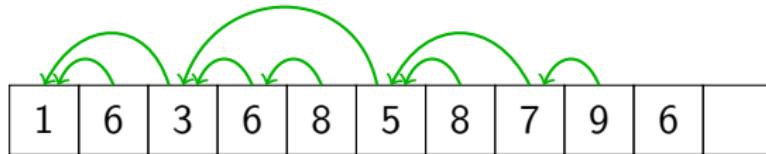
Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time
- no extra space required
- computation of PSV:



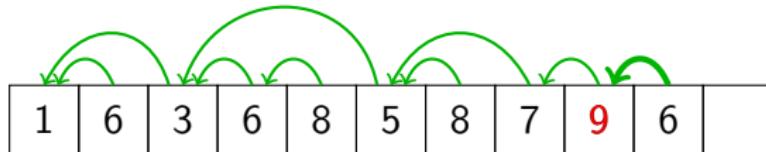
Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time
- no extra space required
- computation of PSV:



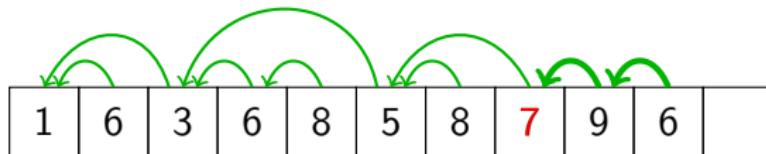
Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time
- no extra space required
- computation of PSV:



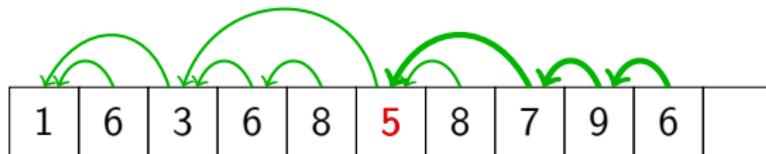
Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time
- no extra space required
- computation of PSV:



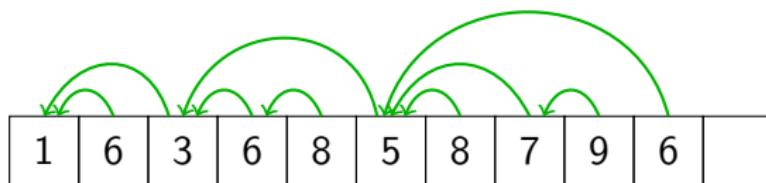
Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time
- no extra space required
- computation of PSV:



Computing NSV/PSV arrays

- PSV/NSV can be computed from SA in linear time
- no extra space required
- computation of PSV:



LZ77 with NSV/PSV arrays

Algorithm:

- ① Compute SA

LZ77 with NSV/PSV arrays

Algorithm:

- ① Compute SA
- ② Compute PSV/NSV (in place)

LZ77 with NSV/PSV arrays

Algorithm:

- ① Compute SA
- ② Compute PSV/NSV (in place)
- ③ Compute the factorization

LZ77 with NSV/PSV arrays

Algorithm:

- ① Compute SA
 - ② Compute PSV/NSV (in place)
 - ③ Compute the factorization
- $\mathcal{O}(n)$ time

LZ77 with NSV/PSV arrays

Algorithm:

- ① Compute SA
- ② Compute PSV/NSV (in place)
- ③ Compute the factorization

$\mathcal{O}(n)$ time, $3n \log n$ bits of extra space

LZ77 with NSV/PSV arrays

Algorithm:

- ① Compute SA
- ② Compute PSV/NSV (in place)
- ③ Compute the factorization

$\mathcal{O}(n)$ time, $3n \log n$ bits of extra space

Observation

We only need to access each PSV/NSV value *once*, in a left-to-right scan.

LZ77 with NSV/PSV arrays

Algorithm:

- ① Compute SA
- ② Compute PSV/NSV (in place)
- ③ Compute the factorization

$\mathcal{O}(n)$ time, $3n \log n$ bits of extra space

Observation

We only need to access each PSV/NSV value *once*, in a left-to-right scan.

Lemma

A scan of NSV and PSV can be simulated with only one of them.
It takes linear time and requires no extra space.

Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].

0	13	1	9	2	3	11	8	10	4	12	6	15	7	14	5	16	0
---	----	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---

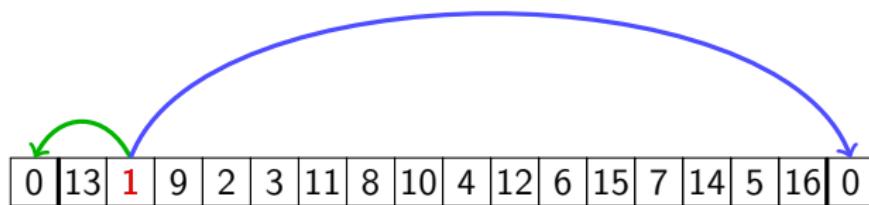
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].

0	13	1	9	2	3	11	8	10	4	12	6	15	7	14	5	16	0
---	----	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---

Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



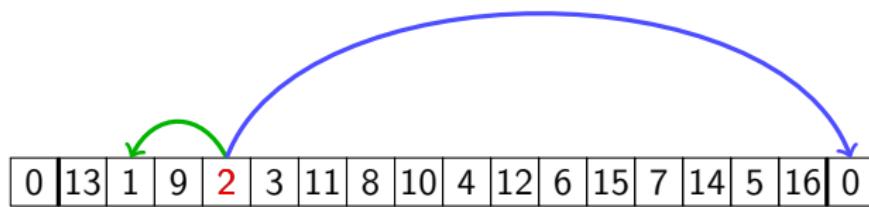
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].

0	13	1	9	2	3	11	8	10	4	12	6	15	7	14	5	16	0
---	----	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---

Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



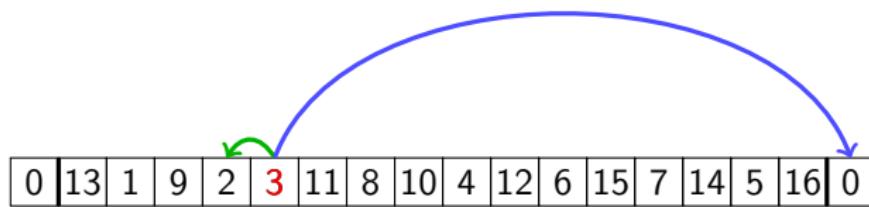
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].

0	13	1	9	2	3	11	8	10	4	12	6	15	7	14	5	16	0
---	----	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---

Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



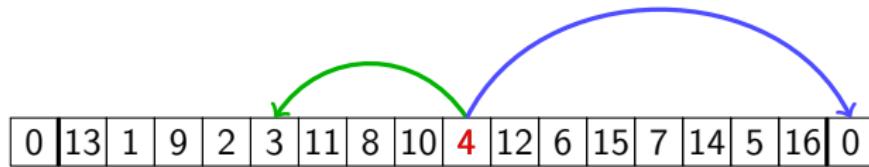
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].

0	13	1	9	2	3	11	8	10	4	12	6	15	7	14	5	16	0
---	----	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---

Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



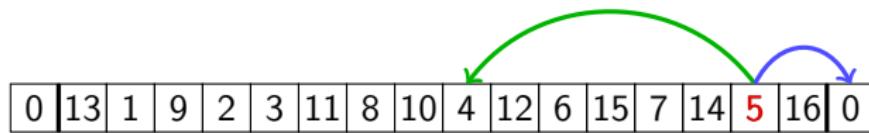
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].

0	13	1	9	2	3	11	8	10	4	12	6	15	7	14	5	16	0
---	----	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---

Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



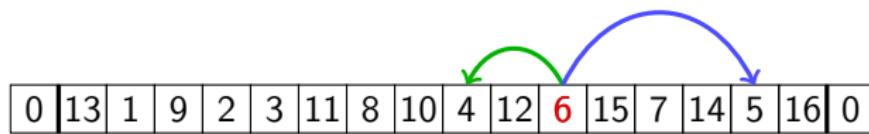
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].

0	13	1	9	2	3	11	8	10	4	12	6	15	7	14	5	16	0
---	----	---	---	---	---	----	---	----	---	----	---	----	---	----	---	----	---

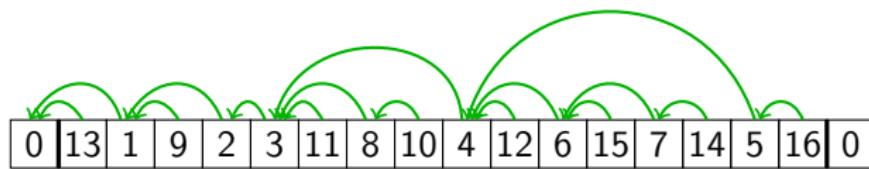
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



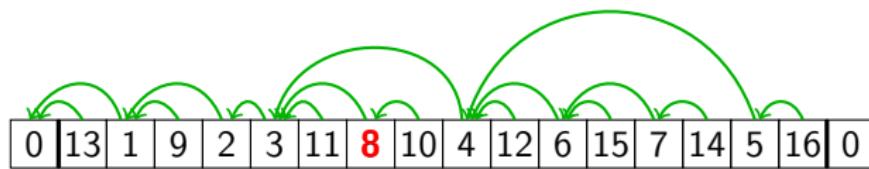
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



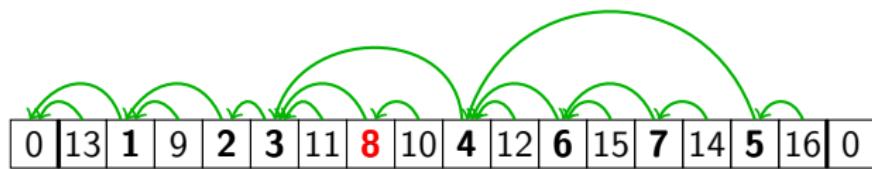
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



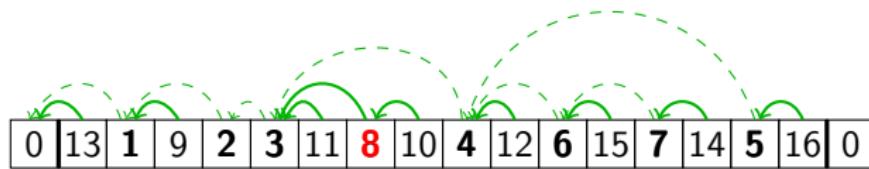
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



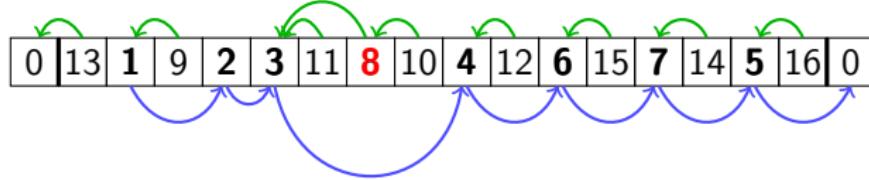
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



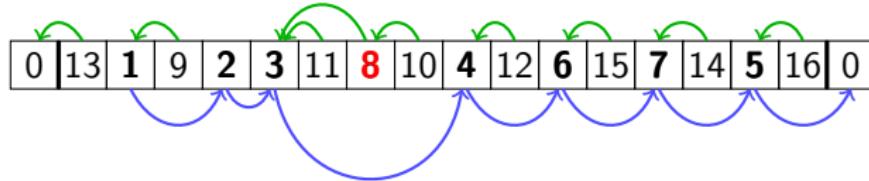
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].



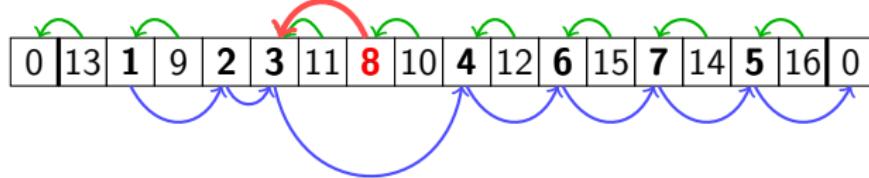
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] =



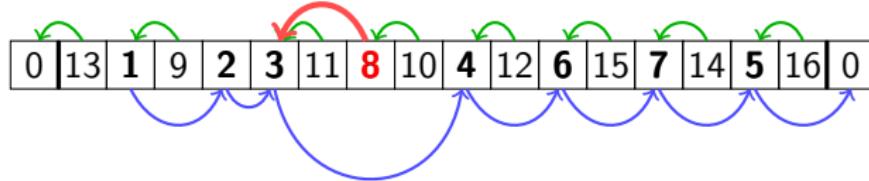
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV $[i]$ and NSV $[i]$.
- PSV $[8] = 3$



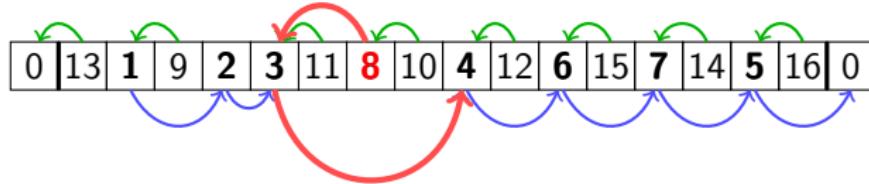
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] = 3
- NSV[8] =



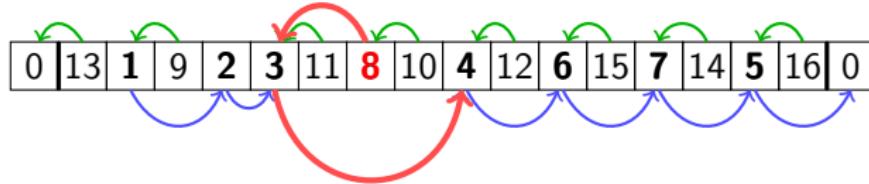
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] = 3
- NSV[8] = 4



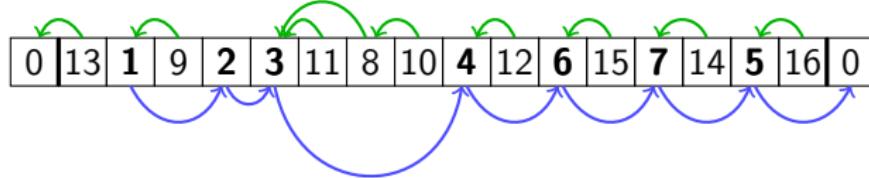
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] = 3
- NSV[8] = 4



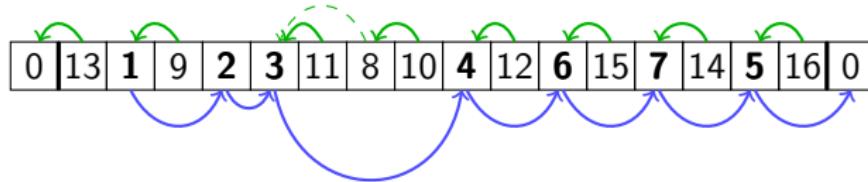
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] = 3
- NSV[8] = 4
- Updating the links:



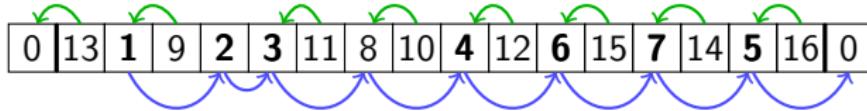
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] = 3
- NSV[8] = 4
- Updating the links:



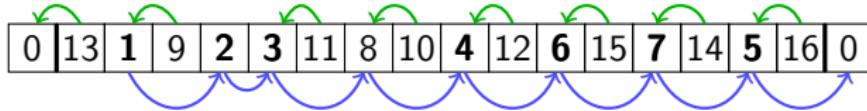
Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] = 3
- NSV[8] = 4
- Updating the links:



Simulating the scan of NSV/PSV

- Goal: at step i know the value of PSV[i] and NSV[i].
- PSV[8] = 3
- NSV[8] = 4
- Updating the links: $\mathcal{O}(1)$ time



The end

Thank you!

Experiments

- Dataset from Pizza & Chilli corpus

Experiments

- Dataset from Pizza & Chilli corpus
- Measured the time to compute LZ factorization

Experiments

- Dataset from Pizza & Chilli corpus
- Measured the time to compute LZ factorization
 - we exclude the time to compute SA

Experiments

- Dataset from Pizza & Chilli corpus
- Measured the time to compute LZ factorization
 - we exclude the time to compute SA

Alg.	Mem	pro	eng	dna	src	cor	cer	ker	ein	tm29
K3	$13n$	74.5	75.7	81.7	50.5	43.6	63.2	45.7	56.9	38.2
K2	$9n$	84.1	80.6	92.7	54.8	40.2	53.2	41.5	43.5	35.1
ISA6r	$6n$	-	-	-	-	43.3	51.8	39.2	31.1	34.2
ISA6s	$6n$	198.0	171.0	175.2	115.0	49.4	56.3	45.7	37.1	39.6
ISA9	$9n$	92.7	83.9	86.1	59.3	41.9	53.0	42.8	45.2	36.4
iBGS	$17n$	99.8	93.2	97.5	69.3	51.5	65.5	52.9	60.0	44.1
iBGL	$17n$	123.2	108.6	113.4	77.8	52.2	66.1	53.0	58.6	44.2
iBGT	$13n$	171.4	153.9	188.0	99.8	55.4	84.1	56.2	52.8	44.4