

SACABench: Benchmarking Suffix Array Construction

A Practical Survey

Johannes Bahne Nico Bertram Marvin Böcker Jonas Bode Johannes Fischer
Hermann Foot Florian Grieskamp *Florian Kurpicz* Marvin Löbel Oliver Magiera
Rosa Pink David Piper Christopher Poeplau

1

SACABench: Benchmarking Suffix Array Construction

A Practical Survey

Johannes Bahne Nico Bertram Marvin Böcker Jonas Bode Johannes Fischer
Hermann Foot Florian Grieskamp *Florian Kurpicz* Marvin Löbel Oliver Magiera
Rosa Pink David Piper Christopher Poeplau

1

2

SACABench: Benchmarking Suffix Array Construction

A Practical Survey

Johannes Bahne Nico Bertram Marvin Böcker Jonas Bode Johannes Fischer
Hermann Foot Florian Grieskamp *Florian Kurpicz* Marvin Löbel Oliver Magiera
Rosa Pink David Piper Christopher Poeplau

3

1

2

SACABench: Benchmarking Suffix Array Construction

A Practical Survey

Johannes Bahne Nico Bertram Marvin Böcker Jonas Bode Johannes Fischer
Hermann Foot Florian Grieskamp *Florian Kurpicz* Marvin Löbel Oliver Magiera
Rosa Pink David Piper Christopher Poeplau

SUFFIX ARRAY

T=

0	1	2	3	4	5	6	7	8	9	10	11	12
c	c	e	c	e	c	e	d	c	c	e	d	\$

SA=

0	1	2	3	4	5	6	7	8	9	10	11	12
c	c	e	c	e	c	e	d	c	c	e	d	\$
c	e	c	e	c	e	d	c	c	e	d	\$	
e	c	e	c	e	d	c	c	e	d	\$		
c	e	c	e	d	c	c	e	d	\$			
e	c	e	d	c	c	e	d	\$				
c	e	d	c	c	e	d	\$					
c	c	c	e	d	\$							
c	e	d	\$									
e	d	\$										
d	\$											
\$												

SUFFIX ARRAY

T=

0	1	2	3	4	5	6	7	8	9	10	11	12
c	c	e	c	e	c	e	d	c	c	e	d	\$

SA=

12	0	8	1	3	9	5	11	7	2	4	10	6
\$	c	c	c	c	c	c	d	d	e	e	e	e
	c	c	e	e	e	e	\$	c	c	c	d	d
	e	e	c	c	d	d		c	e	e	\$	c
	c	d	e	e	\$	c		e	c	d		c
	e	\$	c	d		c		d	e	c		e
	c		e	c		e		\$	d	c		d
	e		d	c		d			c	e		\$
	d		c	e					c	d		
	c		e	d					e			
	e		d						\$			
	d											
	\$											

SUFFIX ARRAY

	0	1	2	3	4	5	6	7	8	9	10	11	12
T=	c	c	e	c	e	c	e	d	c	c	e	d	\$
SA=	12	0	8	1	3	9	5	11	7	2	4	10	6
	\$	c	c	c	c	c	c	d	d	e	e	e	e
	c	c	c	c	c	c	c	\$	c	c	c	d	d
	e	e	e	e	e	e	e	c	c	c	c	c	c
	e	d	d	d	d	d	d	c	c	c	c	c	c
	d	d	d	d	d	d	d	c	c	c	c	c	c
	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$	\$

SUFFIX ARRAY

	0	1	2	3	4	5	6	7	8	9	10	11	12
T=	c	c	e	c	e	c	e	d	c	c	e	d	\$
SA=	12	0	8	1	3	9	5	11	7	2	4	10	6
	\$	c	c	c	c	c	c	d	d	e	e	e	e
	c	c	c	c	c	c	c	\$	c	c	c	d	d
	e	e	e	e	e	e	e	c	c	e	e	c	c
	c	c	d	c	c	\$	c	c	c	d	c	c	c
	e	d	\$	c	c	c	c	e	e	c	c	d	d
	c	c	c	c	c	c	c	d	c	c	c	c	c
	d	d	c	c	c	c	c	c	c	d	c	c	c
	c	c	c	c	c	c	c	c	c	c	c	c	c
	e	d	c	c	c	c	c	c	c	d	c	c	c
	d	d	c	c	c	c	c	c	c	c	c	c	c
	c	c	c	c	c	c	c	c	c	c	c	c	c
	\$	c	c	c	c	c	c	d	d	e	e	e	e

► well-researched text index

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



SUFFIX ARRAY CONSTRUCTION

Prefix Doubling



Induced Copying

Recursion

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion

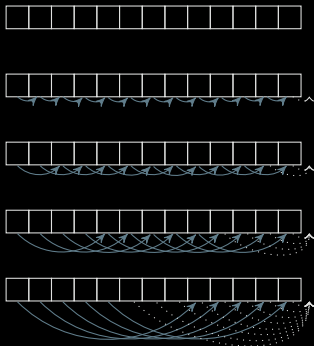


SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion

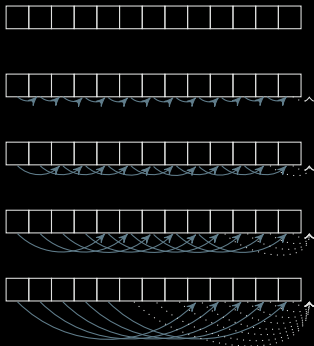


SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



Running Time

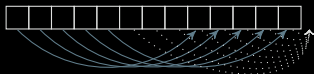
▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



Running Time

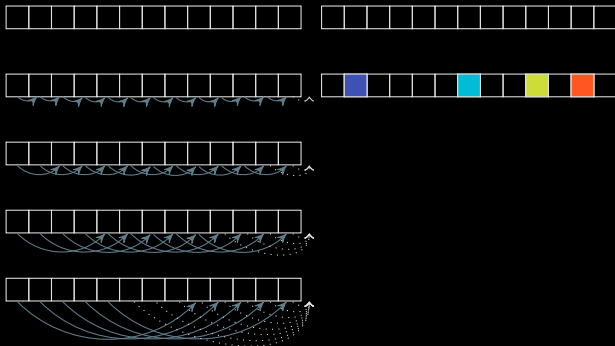
▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



Running Time

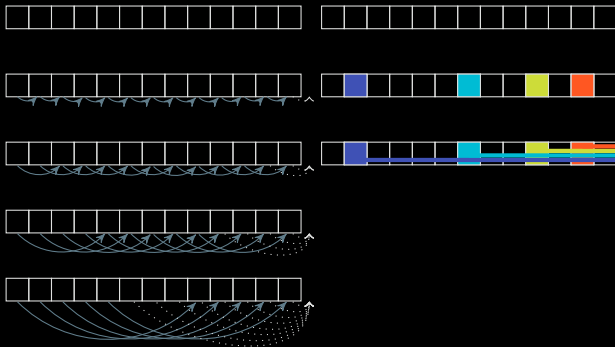
▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



Running Time

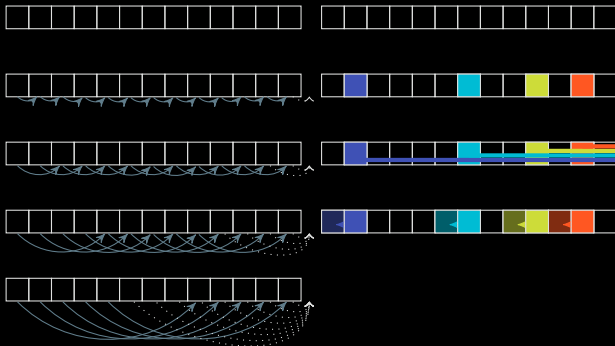
▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



Running Time

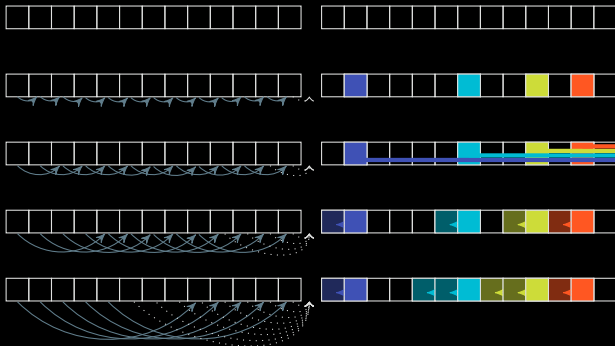
▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



Running Time

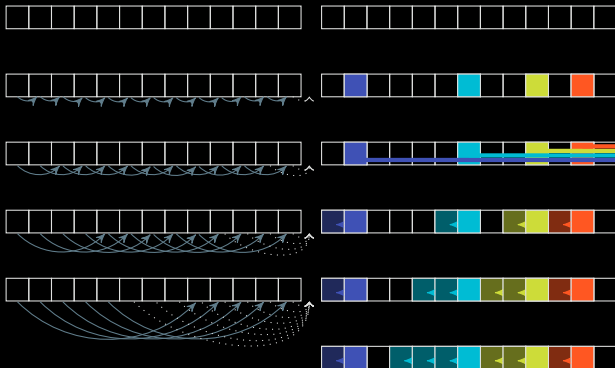
▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion



Running Time

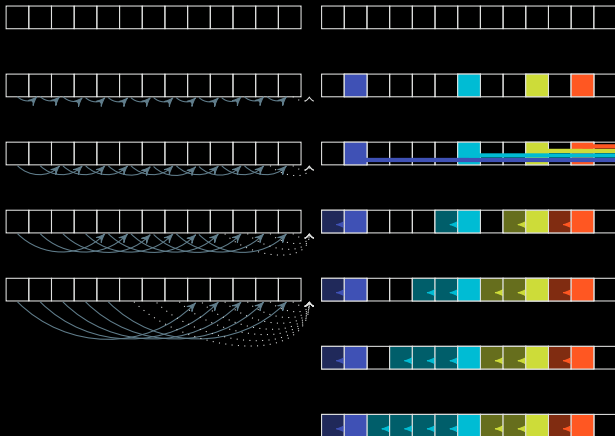
▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

Prefix Doubling

Induced Copying

Recursion

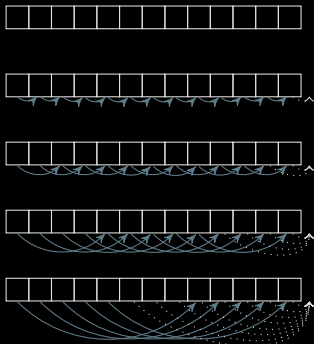


Running Time

▶ $O(n \lg n)$

SUFFIX ARRAY CONSTRUCTION

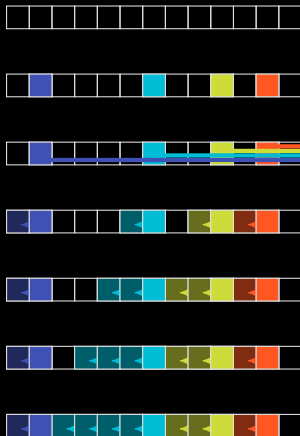
Prefix Doubling



Running Time

▶ $O(n \lg n)$

Induced Copying



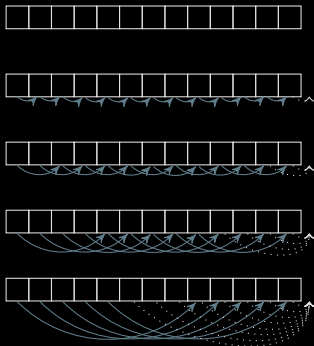
Running Time

▶ $O(n) + O(\text{sort special})$

Recursion

SUFFIX ARRAY CONSTRUCTION

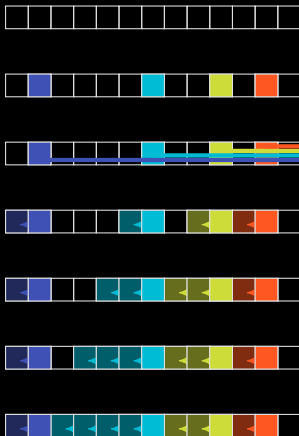
Prefix Doubling



Running Time

▶ $O(n \lg n)$

Induced Copying



Running Time

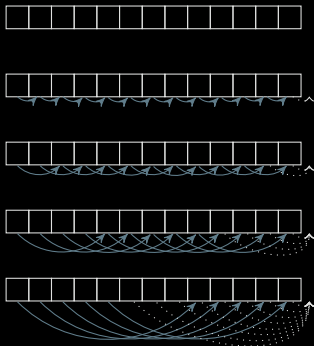
▶ $O(n) + O(\text{sort special})$

Recursion



SUFFIX ARRAY CONSTRUCTION

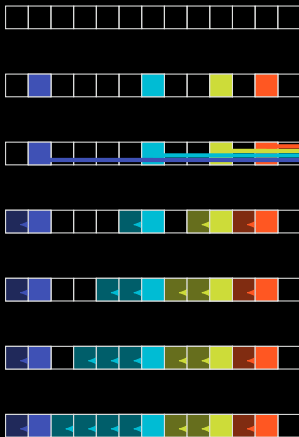
Prefix Doubling



Running Time

▶ $O(n \lg n)$

Induced Copying



Running Time

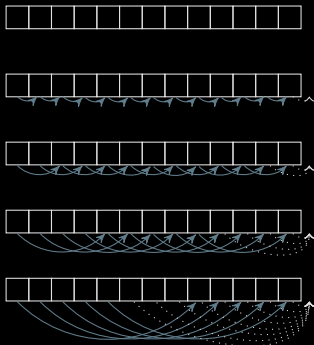
▶ $O(n) + O(\text{sort special})$

Recursion



SUFFIX ARRAY CONSTRUCTION

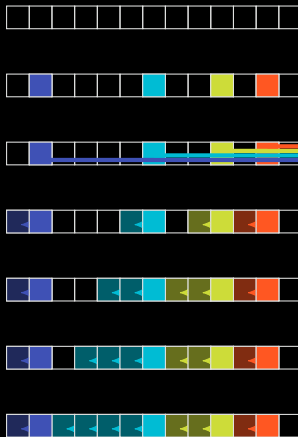
Prefix Doubling



Running Time

▶ $O(n \lg n)$

Induced Copying



Running Time

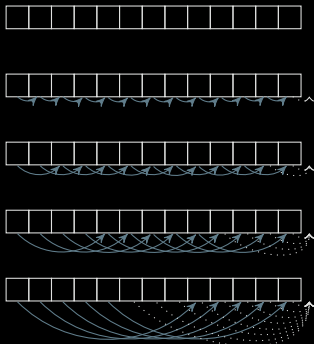
▶ $O(n) + O(\text{sort special})$

Recursion



SUFFIX ARRAY CONSTRUCTION

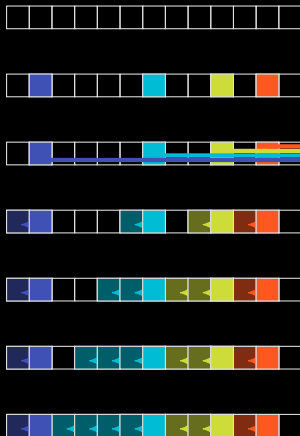
Prefix Doubling



Running Time

▶ $O(n \lg n)$

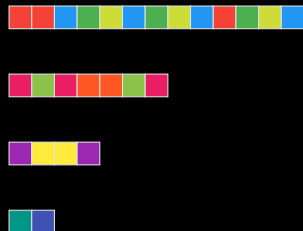
Induced Copying



Running Time

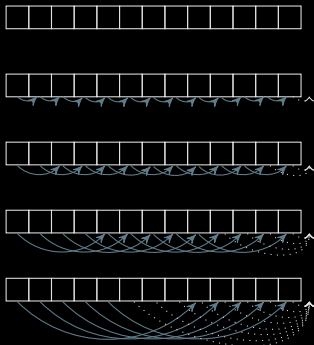
▶ $O(n) + O(\text{sort special})$

Recursion



SUFFIX ARRAY CONSTRUCTION

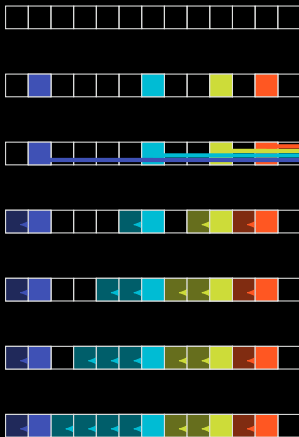
Prefix Doubling



Running Time

▶ $O(n \lg n)$

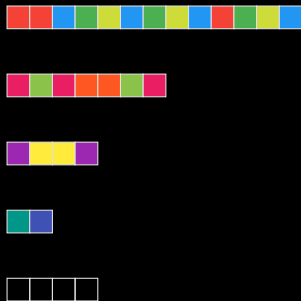
Induced Copying



Running Time

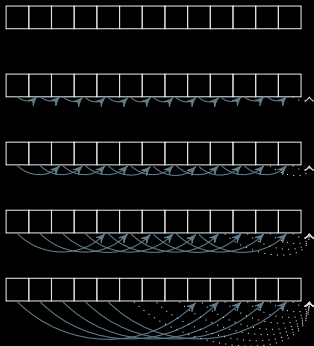
▶ $O(n) + O(\text{sort special})$

Recursion



SUFFIX ARRAY CONSTRUCTION

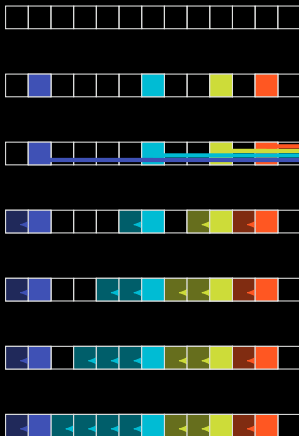
Prefix Doubling



Running Time

▶ $O(n \lg n)$

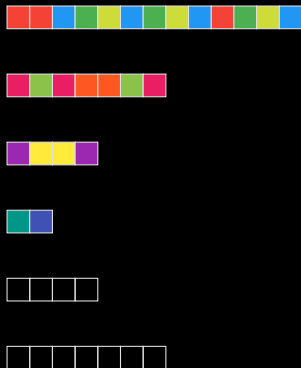
Induced Copying



Running Time

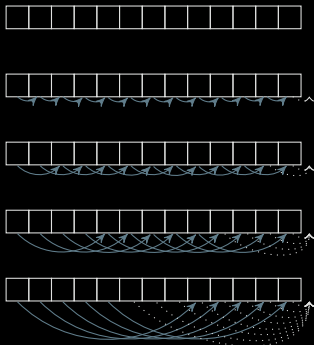
▶ $O(n) + O(\text{sort special})$

Recursion



SUFFIX ARRAY CONSTRUCTION

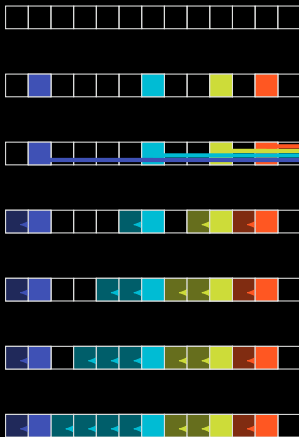
Prefix Doubling



Running Time

▶ $O(n \lg n)$

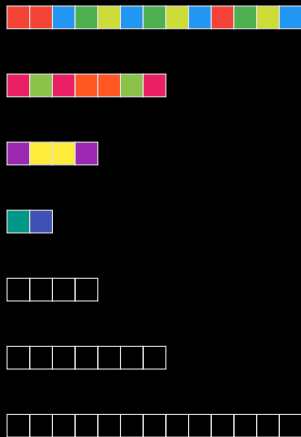
Induced Copying



Running Time

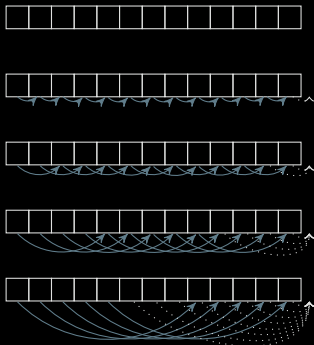
▶ $O(n) + O(\text{sort special})$

Recursion



SUFFIX ARRAY CONSTRUCTION

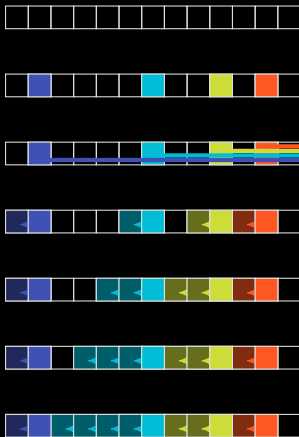
Prefix Doubling



Running Time

▶ $O(n \lg n)$

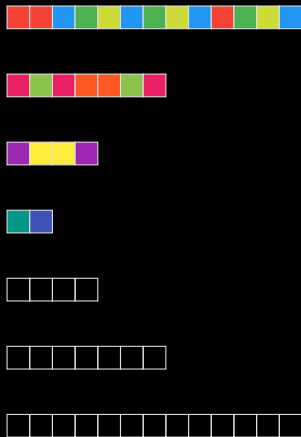
Induced Copying



Running Time

▶ $O(n) + O(\text{sort special})$

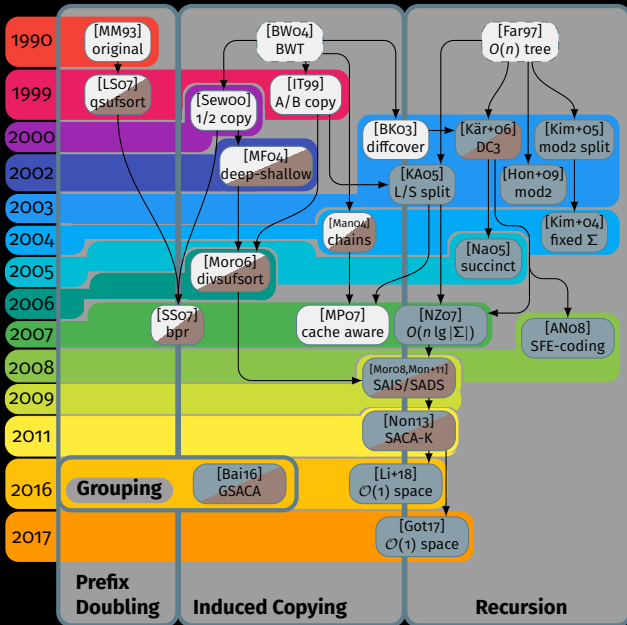
Recursion



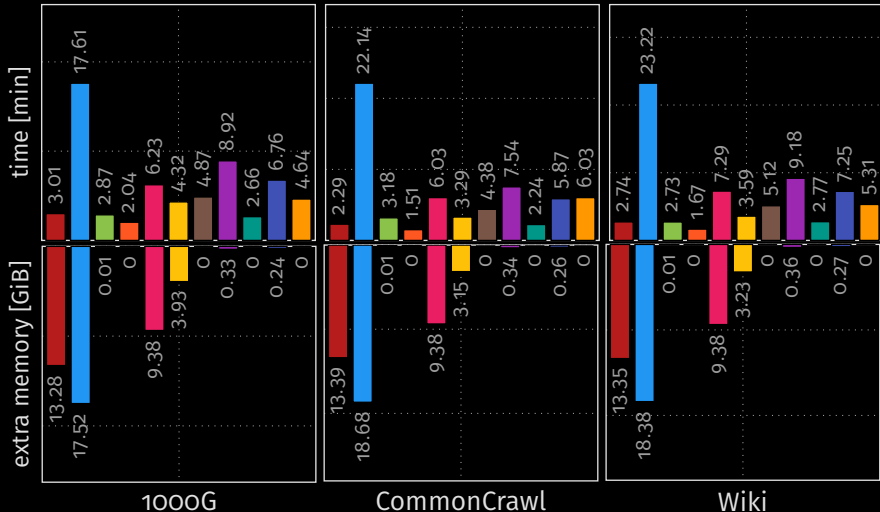
Running Time

▶ $O(n)$

SUFFIX ARRAY CONSTRUCTION

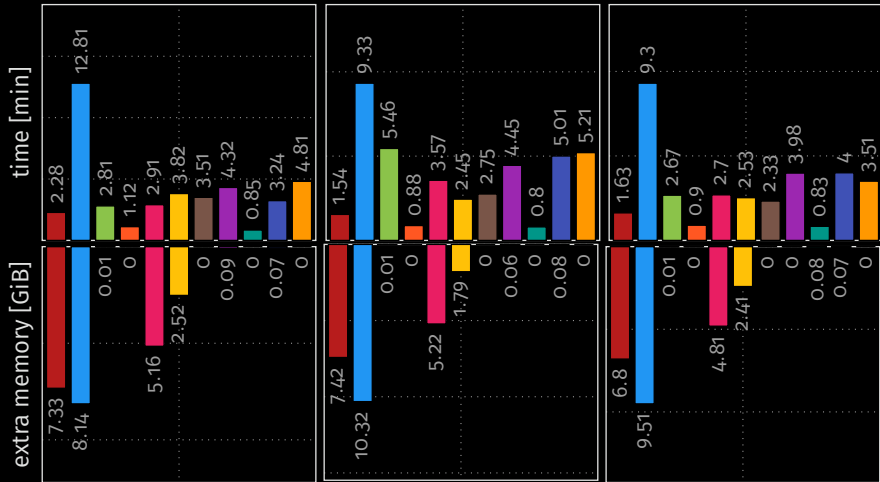


BENCHMARKING (1600 MIB)



- BPR
- DC3
- Deep-Shallow
- DivSufSort
- GSACA
- MSufSort
- SACA-K
- SADS
- SAIS-LITE
- SAIS
- qsufsort

BENCHMARKING (1600 MIB)



CONCLUSION

- ▶ Framework for benchmarking SACAs
- ▶ New implementations for many SACAs ...
- ▶ ... including fast GPU SACA
- ▶ More benchmarks in the paper

CONCLUSION

- ▶ Framework for benchmarking SACAs
- ▶ New implementations for many SACAs ...
- ▶ ... including fast GPU SACA
- ▶ More benchmarks in the paper

Thank You