Wavelet Trees Meet Suffix Trees

Maxim Babenko[1] Paweł Gawrychowski[2→3]
Tomasz Kociumaka[3] Tatiana Starikovskaya[1→4]

[1]National Research University Higher School of Economics, Moscow, Russia

[2]Max-Planck-Institut für Informatik, Saarbrücken, Germany

[3]University of Warsaw, Poland

[4]University of Bristol, UK

SODA 2015
San Diego, California, USA
January 4, 2015

Rank and Select Queries

General Setting

Universe	totally ordered universe U (integers, strings,...)
Input	multiset A over universe U
$rank_A(u)$	count elements in A not exceeding u
$select_A(k)$	return the k-th smallest element in A

Rank and Select Queries

General Setting

Universe	totally ordered universe U (integers, strings,...)
Input	multiset A over universe U
$rank_A(u)$	count elements in A not exceeding u
$select_A(k)$	return the k-th smallest element in A

Example: an integer multiset

$$A = \{0, 1, 3, 4, 5, 5, 8, 8, 9, 9\}$$

Rank and Select Queries

General Setting

Universe	totally ordered universe U (integers, strings,...)
Input	multiset A over universe U
$rank_A(u)$	count elements in A not exceeding u
$select_A(k)$	return the k-th smallest element in A

Example: an integer multiset

$$A = \{0, 1, 3, 4, 5, 5, 8, 8, 9, 9\}$$

$$rank_A(7) = 6$$

Rank and Select Queries

General Setting

Universe	totally ordered universe U (integers, strings,...)
Input	multiset A over universe U
$rank_A(u)$	count elements in A not exceeding u
$select_A(k)$	return the k-th smallest element in A

Example: an integer multiset

$$A = \{0, 1, 3, 4, 5, 5, 8, 8, 9, 9\}$$

$$rank_A(7) = 6 \qquad select_A(7) = 8$$

Rank and Select Queries

General Setting

Universe	totally ordered universe U (integers, strings, . . .)
Input	multiset A over universe U
$rank_A(u)$	count elements in A not exceeding u
$select_A(k)$	return the k-th smallest element in A

Example: an integer multiset

$$A = \{0, 1, 3, 4, 5, 5, 8, 8, 9, 9\}$$

$$rank_A(7) = 6 \qquad select_A(7) = 8$$

Example: a set encoded as bitmask

$$A = \{2, 3, 5, 9, 10, 11, 13, 14, 15, 16, 17, 18, 20\}$$

01101000111011111101

Rank and Select Queries

General Setting

Universe	totally ordered universe U (integers, strings,...)
Input	multiset A over universe U
$rank_A(u)$	count elements in A not exceeding u
$select_A(k)$	return the k-th smallest element in A

Example: an integer multiset

$$A = \{0, 1, 3, 4, 5, 5, 8, 8, 9, 9\}$$

$$rank_A(7) = 6 \qquad select_A(7) = 8$$

Example: a set encoded as bitmask

$$A = \{2, 3, 5, 9, 10, 11, 13, 14, 15, 16, 17, 18, 20\}$$

01101000111011111101

$rank_1(7) = 3$

Rank and Select Queries

General Setting

Universe	totally ordered universe U (integers, strings,...)
Input	multiset A over universe U
$rank_A(u)$	count elements in A not exceeding u
$select_A(k)$	return the k-th smallest element in A

Example: an integer multiset

$$A = \{0, 1, 3, 4, 5, 5, 8, 8, 9, 9\}$$

$$rank_A(7) = 6 \qquad select_A(7) = 8$$

Example: a set encoded as bitmask

$$A = \{2, 3, 5, 9, 10, 11, 13, 14, 15, 16, 17, 18, 20\}$$

011010001110111111101

$$rank_1(7) = 3 \qquad select_1(7) = 13$$

RANGE RANK and SELECTION

Range Rank and Select Queries

Input	An integer sequence $S = (S_1, \ldots, S_n)$.
Queries	Rank and select on a range $S_{\ell..r} = \{S_\ell, S_{\ell+1}, \ldots, S_r\}$.

RANGE RANK and SELECTION

Range Rank and Select Queries

Input An integer sequence $S = (S_1, \ldots, S_n)$.

Queries Rank and select on a range $S_{\ell..r} = \{S_\ell, S_{\ell+1}, \ldots, S_r\}$.

Special case:

- Range Minimum Queries (RMQ).

RANGE RANK and SELECTION

Range Rank and Select Queries

Input	An integer sequence $S = (S_1, \ldots, S_n)$.
Queries	Rank and select on a range $S_{\ell..r} = \{S_\ell, S_{\ell+1}, \ldots, S_r\}$.

Special case:

- Range Minimum Queries (RMQ).

Example:

S: 2 6 0 7 0 0 2 1 4 1 8 1 6 7 4 2 9 3 0 5

1 20

RANGE RANK and SELECTION

Range Rank and Select Queries

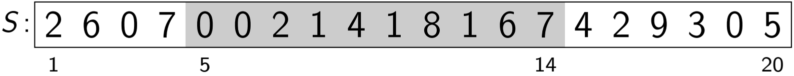
Input An integer sequence $S = (S_1, \ldots, S_n)$.

Queries Rank and select on a range $S_{\ell..r} = \{S_\ell, S_{\ell+1}, \ldots, S_r\}$.

Special case:

- Range Minimum Queries (RMQ).

Example:



S:	2	6	0	7	0	0	2	1	4	1	8	1	6	7	4	2	9	3	0	5
	1				5									14						20

$$rank_{5..14}(6) =$$

RANGE RANK and SELECTION

Range Rank and Select Queries

Input	An integer sequence $S = (S_1, \ldots, S_n)$.
Queries	Rank and select on a range $S_{\ell..r} = \{S_\ell, S_{\ell+1}, \ldots, S_r\}$.

Special case:

- Range Minimum Queries (RMQ).

Example:

S:	2	6	0	7	0	0	2	1	4	1	8	1	6	7	4	2	9	3	0	5
	1				5									14						20

$$rank_{5..14}(6) = 8$$

Range Rank and Selection

Range Rank and Select Queries

Input An integer sequence $S = (S_1, \ldots, S_n)$.

Queries Rank and select on a range $S_{\ell..r} = \{S_\ell, S_{\ell+1}, \ldots, S_r\}$.

Special case:

- Range Minimum Queries (RMQ).

Example:

S:	2	6	0	7	0	0	2	1	4	1	8	1	6	7	4	2	9	3	0	5
	1								9								17			20

$rank_{5..14}(6) = 8$ $\qquad$ $select_{9..17}(5) =$

RANGE RANK and SELECTION

Range Rank and Select Queries

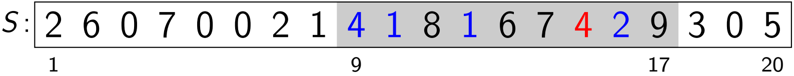
Input An integer sequence $S = (S_1, \ldots, S_n)$.

Queries Rank and select on a range $S_{\ell..r} = \{S_\ell, S_{\ell+1}, \ldots, S_r\}$.

Special case:

- Range Minimum Queries (RMQ).

Example:



S:	2	6	0	7	0	0	2	1	4	1	8	1	6	7	4	2	9	3	0	5
	1								9								17			20

$$rank_{5..14}(6) = 8 \qquad select_{9..17}(5) = 4.$$

Range Rank and Selection: Results

Problem	Space	Query	Construction	Reference
Rank	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log\log n})$	$\mathcal{O}(n\sqrt{\log n})$	Chan & Pătrașcu SODA 2010
(negative)	$n\log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log\log n})$	–	Pătrașcu STOC 2007

Range Rank and Selection: Results

Problem	Space	Query	Construction	Reference
Rank	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log \log n})$	$\mathcal{O}(n\sqrt{\log n})$	Chan & Pătrașcu SODA 2010
(negative)	$n \log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log \log n})$	–	Pătrașcu STOC 2007
Select	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(n\sqrt{\log n})$	Chan & Pătrașcu SODA 2010
	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log \log n})$	$\mathcal{O}(n \log n)$	Brodal et al. ICALP 2009
(negative)	$n \log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log \log n})$	–	Jørgensen & Larsen SODA 2011

Range Rank and Selection: Results

Problem	Space	Query	Construction	Reference
Rank	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log\log n})$	$\mathcal{O}(n\sqrt{\log n})$	Chan & Pătrașcu SODA 2010
(negative)	$n\log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log\log n})$	–	Pătrașcu STOC 2007
Select	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(n\sqrt{\log n})$	Chan & Pătrașcu SODA 2010
	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log\log n})$	$\mathcal{O}(n\log n)$	Brodal et al. ICALP 2009
(negative)	$n\log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log\log n})$	–	Jørgensen & Larsen SODA 2011
Select	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log\log n})$	$\mathcal{O}(n\sqrt{\log n})$	this work

Range Rank and Selection: Results

Problem	Space	Query	Construction	Reference
Rank	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log\log n})$	$\mathcal{O}(n\sqrt{\log n})$	Chan & Pătrașcu SODA 2010
(negative)	$n\log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log\log n})$	–	Pătrașcu STOC 2007
Select	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(n\sqrt{\log n})$	Chan & Pătrașcu SODA 2010
	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log\log n})$	$\mathcal{O}(n\log n)$	Brodal et al. ICALP 2009
(negative)	$n\log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log\log n})$	–	Jørgensen & Larsen SODA 2011
Select	$\mathcal{O}(n)$	$\mathcal{O}(\frac{\log n}{\log\log n})$	$\mathcal{O}(n\sqrt{\log n})$	this work

Theorem (independently: Munro, Nekrich, Vitter; SPIRE 2014)

Wavelet trees can be constructed in $\mathcal{O}(n\sqrt{\log n})$ *time.*

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on the set of suffixes of T ($Suf(T)$).

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe strings over Σ (denoted Σ^*)

Input a string T of length n

Queries rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11
ISA	6	10	3	7	11	5	9	2	4	8	1

i	SA	
1	11	a
2	8	aaba
3	3	aababaaba
4	9	aba
5	6	abaaba
6	1	abaababaaba
7	4	ababaaba
8	10	ba
9	7	baaba
10	2	baababaaba
11	5	babaaba

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11
ISA	6	10	3	7	11	5	9	2	4	8	1

$select_T(5) =$

i	SA	
1	11	a
2	8	aaba
3	3	aabababa
4	9	aba
5	6	abaaba
6	1	abaababaaba
7	4	ababaaba
8	10	ba
9	7	baaba
10	2	baababaaba
11	5	babaaba

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11
ISA	6	10	3	7	11	5	9	2	4	8	1

$$select_T(5) = T[\mathrm{SA}[5]..] = T[6..] = \texttt{abaaba}$$

i	SA	
1	11	a
2	8	aaba
3	3	aabababa
4	9	aba
5	6	abaaba
6	1	abaababaaba
7	4	ababaaba
8	10	ba
9	7	baaba
10	2	baababaaba
11	5	babaaba

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11
ISA	6	10	3	7	11	5	9	2	4	8	1

$select_T(5) = T[\mathrm{SA}[5]..] = T[6..] =$ `abaaba`

$rank_T(T[4..]) =$

i	SA	
1	11	a
2	8	aaba
3	3	aababaaba
4	9	aba
5	6	abaaba
6	1	abaababaaba
7	4	ababaaba
8	10	ba
9	7	baaba
10	2	baababaaba
11	5	babaaba

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11
ISA	6	10	3	7	11	5	9	2	4	8	1

$$select_T(5) = T[\mathrm{SA}[5]..] = T[6..] = \texttt{abaaba}$$

$$rank_T(T[4..]) = \mathrm{ISA}[4] = 7$$

i	SA	
1	11	a
2	8	aaba
3	3	aababaaba
4	9	aba
5	6	abaaba
6	1	abaababaaba
7	4	ababaaba
8	10	ba
9	7	baaba
10	2	baababaaba
11	5	babaaba

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe strings over Σ (denoted Σ^*)

Input a string T of length n

Queries rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11
ISA	6	10	3	7	11	5	9	2	4	8	1

$$select_T(5) = T[\mathrm{SA}[5]..] = T[6..] = \texttt{abaaba}$$

$$rank_T(T[4..]) = \mathrm{ISA}[4] = 7$$

$$rank_T(\texttt{aabb}) =$$

i	SA	
1	11	a
2	8	aaba
3	3	aababaaba
4	9	aba
5	6	abaaba
6	1	abaababaaba
7	4	ababaaba
8	10	ba
9	7	baaba
10	2	baababaaba
11	5	babaaba

Suffix Rank and Selection

Suffix Rank and Selection Queries

Universe strings over Σ (denoted Σ^*)

Input a string T of length n

Queries rank and select on the set of suffixes of T ($Suf(T)$).

T	a	b	a	a	b	a	b	a	a	b	a
i	1	2	3	4	5	6	7	8	9	10	11
ISA	6	10	3	7	11	5	9	2	4	8	1

$$select_T(5) = T[\mathrm{SA}[5]..] = T[6..] = \texttt{abaaba}$$

$$rank_T(T[4..]) = \mathrm{ISA}[4] = 7$$

$$rank_T(\texttt{aabb}) = 3$$

i	SA	
1	11	a
2	8	aaba
3	3	aababaaba
4	9	aba
5	6	abaaba
6	1	abaababaaba
7	4	ababaaba
8	10	ba
9	7	baaba
10	2	baababaaba
11	5	babaaba

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe strings over Σ (denoted Σ^*)

Input a string T of length n

Queries rank and select on suffixes of subwords $T[\ell..r]$ of T.

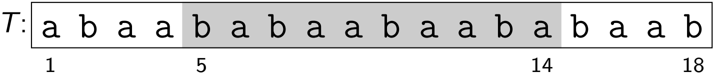
T: `a b a a b a b a a b a a b a b a a b`

1 … 18

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.

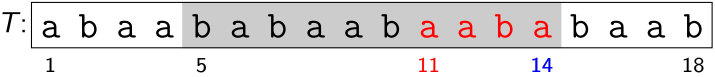


$$select_{T[5..14]}(2) =$$

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.



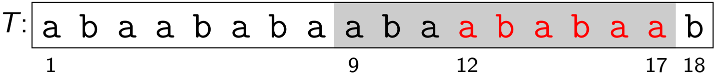
T:	a	b	a	a	b	a	b	a	a	b	a	a	b	a	b	a	a	b
	1				5						11			14				18

$$select_{T[5..14]}(2) = T[11..14] = \texttt{aaba}$$

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.



T:	a	b	a	a	b	a	b	a	a	b	a	a	b	a	b	a	a	b
	1								9			12					17	18

$$select_{T[5..14]}(2) = T[11..14] = \text{aaba}$$

$$rank_{T[9..17]}(T[12..17]) =$$

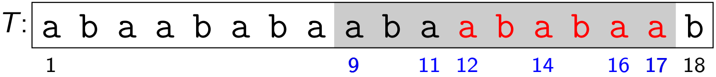
Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe strings over Σ (denoted Σ^*)

Input a string T of length n

Queries rank and select on suffixes of subwords $T[\ell..r]$ of T.



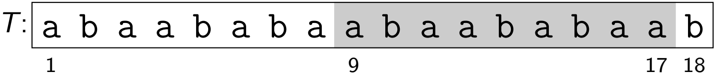
$$select_{T[5..14]}(2) = T[11..14] = \mathtt{aaba}$$

$$rank_{T[9..17]}(T[12..17]) = 6$$

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.



T:	a	b	a	a	b	a	b	a	a	b	a	a	b	a	b	a	a	b
	1								9								17	18

$$select_{T[5..14]}(2) = T[11..14] = \texttt{aaba}$$

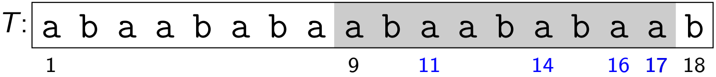
$$rank_{T[9..17]}(T[12..17]) = 6$$

$$rank_{T[9..17]}(\texttt{abaab}) =$$

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.



T:	a	b	a	a	b	a	b	a	a	b	a	a	b	a	b	a	a	b
	1								9		11			14		16	17	18

$$select_{T[5..14]}(2) = T[11..14] = \texttt{aaba}$$

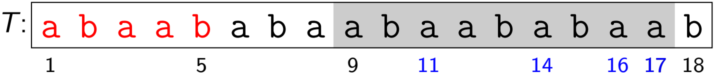
$$rank_{T[9..17]}(T[12..17]) = 6$$

$$rank_{T[9..17]}(\texttt{abaab}) = 4$$

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.



T:	a	b	a	a	b	a	b	a	a	b	a	a	b	a	b	a	a	b
	1				5				9		11			14		16	17	18

$$select_{T[5..14]}(2) = T[11..14] = \text{aaba}$$

$$rank_{T[9..17]}(T[12..17]) = 6$$

$$rank_{T[9..17]}(\text{abaab}) = 4$$

$$rank_{T[9..17]}(T[1..5]) = 4$$

Substring Suffix Rank and Selection

Substring Suffix Rank and Selection Queries

Universe	strings over Σ (denoted Σ^*)
Input	a string T of length n
Queries	rank and select on suffixes of subwords $T[\ell..r]$ of T.

T: `a b a a b a b a a b a a b a b a a b`

1 18

$$select_{T[5..14]}(2) = T[11..14] = \texttt{aaba}$$

$$rank_{T[9..17]}(T[12..17]) = 6$$

~~$rank_{T[9..17]}(\texttt{abaab}) = 4$~~

$$rank_{T[9..17]}(T[1..5]) = 4$$

Subword Suffix Rank and Selection: Results

Problem	Space	Query	Construction	Reference
Maximum	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	B.G.K.S. CPM 2014
Minimum	$\mathcal{O}(n)$	$\mathcal{O}(1)$ $\mathcal{O}(\log n)$	$\mathcal{O}(n \log n)$ $\mathcal{O}(n)$	B.G.K.S. CPM 2014

Subword Suffix Rank and Selection: Results

Problem	Space	Query	Construction	Reference
Maximum	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	B.G.K.S. CPM 2014
Minimum	$\mathcal{O}(n)$	$\mathcal{O}(1)$ $\mathcal{O}(\log n)$	$\mathcal{O}(n \log n)$ $\mathcal{O}(n)$	B.G.K.S. CPM 2014
Rank & Select	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(n\sqrt{\log n})^*$	this work
(negative)	$n \log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log \log n})$	—	

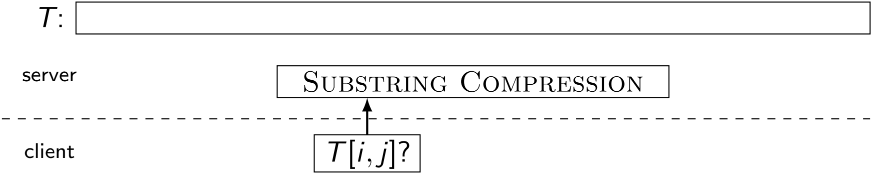
Subword Suffix Rank and Selection: Results

Problem	Space	Query	Construction	Reference
Maximum	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	B.G.K.S. CPM 2014
Minimum	$\mathcal{O}(n)$	$\mathcal{O}(1)$ $\mathcal{O}(\log n)$	$\mathcal{O}(n \log n)$ $\mathcal{O}(n)$	B.G.K.S. CPM 2014
Rank & Select	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(n\sqrt{\log n})^*$	this work
(negative)	$n \log^{\mathcal{O}(1)} n$	$\Omega(\frac{\log n}{\log \log n})$	–	

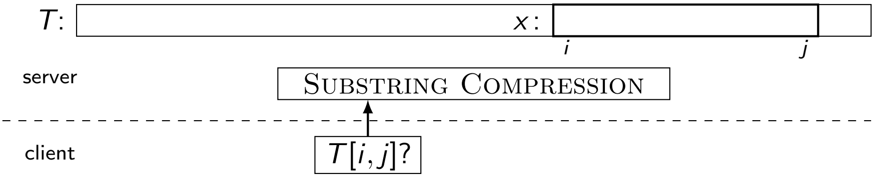
* Wavelet suffix trees: randomized construction: $\mathcal{O}(n\sqrt{\log n})$ deterministic + $\mathcal{O}(n)$ expected.

Applications to Substring Compression

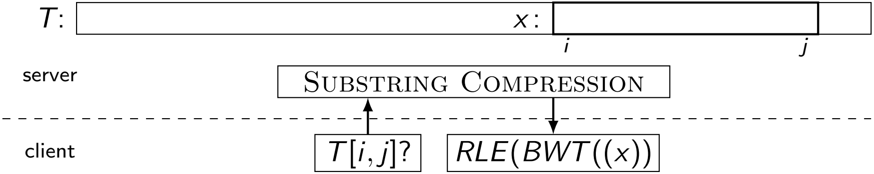
Applications to Substring Compression



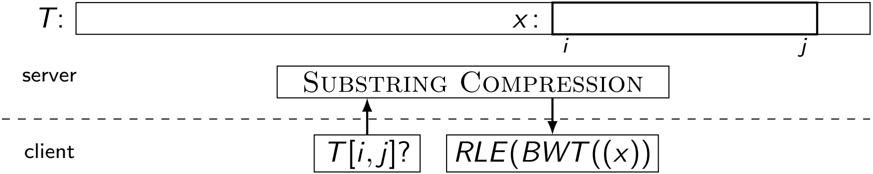
Applications to Substring Compression



Applications to Substring Compression



Applications to Substring Compression



T: x: i j

server

SUBSTRING COMPRESSION

client

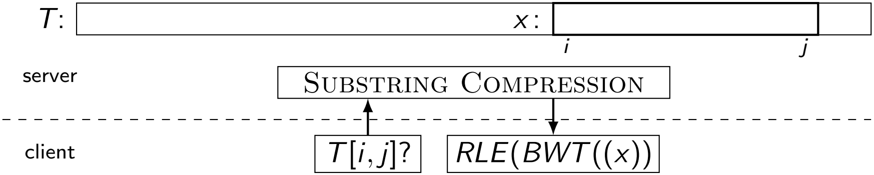
$T[i,j]$?

$RLE(BWT((x))$

Burrows-Wheeler transform:

- often makes data easier to compress with simple methods:
    - run-length encoding.
- $RLE(BWT(\texttt{banana\$})) = RLE(annb\$aa) = a^1n^2b^1\$^1a^2$.

Applications to Substring Compression



T:

x:

i

j

server

SUBSTRING COMPRESSION

client

$T[i,j]?$

$RLE(BWT((x))$

Burrows-Wheeler transform:

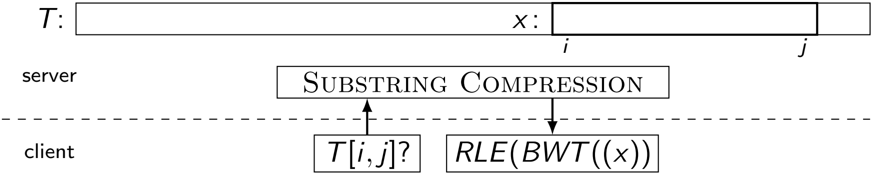
- often makes data easier to compress with simple methods:
  - run-length encoding.
- $RLE(BWT(\texttt{banana\$})) = RLE(annb\$aa) = a^1 n^2 b^1 \$^1 a^2$.

Substring compression: [Cormode & Muthukrishnan; SODA 2005)]

LZ77 $\mathcal{O}(s \log^{\varepsilon} n)$ [Keller et al.; Theor. Comp. Sci., 2014]

BWT+RLE $\mathcal{O}(s \log n)$ [this work]

Applications to Substring Compression



T: x: i j

server

SUBSTRING COMPRESSION

client

$T[i,j]?$

$RLE(BWT((x))$

Burrows-Wheeler transform:

- often makes data easier to compress with simple methods:
  - run-length encoding.
- $RLE(BWT(\texttt{banana\$})) = RLE(annb\$aa) = a^1n^2b^1\$^1a^2$.

Substring compression: [Cormode & Muthukrishnan; SODA 2005)]

LZ77 $\mathcal{O}(s \log^{\varepsilon} n)$ [Keller et al.; Theor. Comp. Sci., 2014]

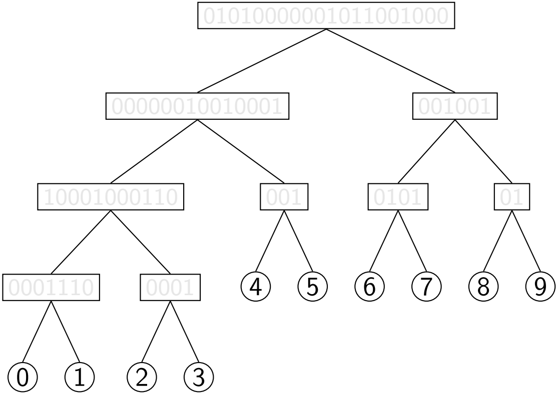
BWT+RLE $\mathcal{O}(s \log n)$ [this work]

Acknowledgement

Thanks to Djamal Belazzougui, Travis Gagie, and Simon Puglisi (University of Helsinki) for suggesting the of study BWT queries.

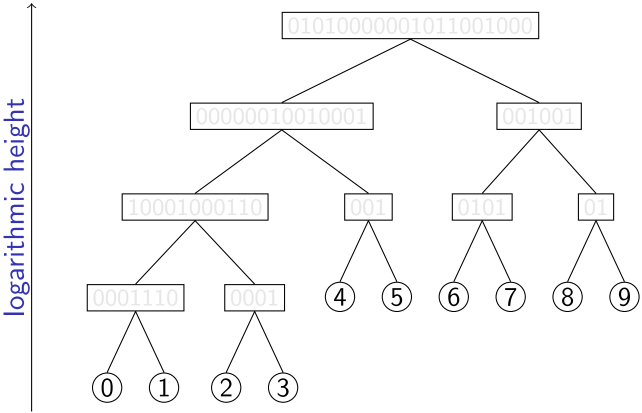
Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



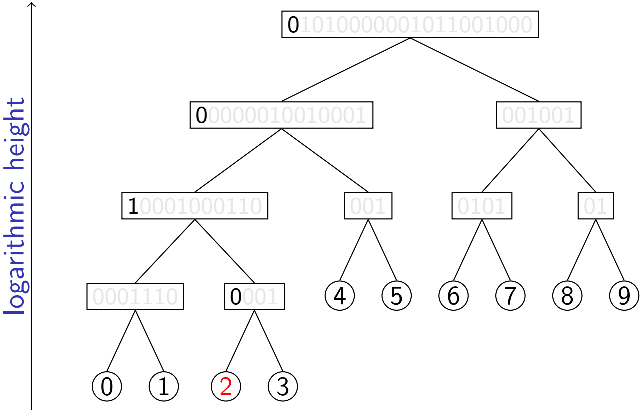
Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $S = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



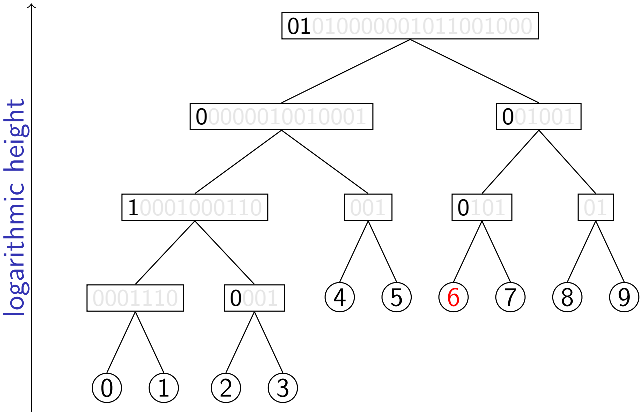
Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



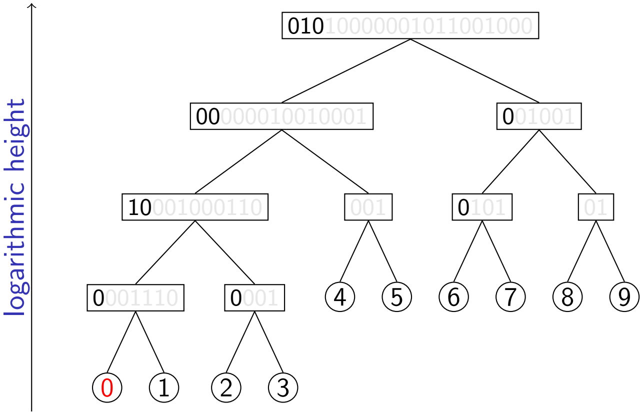
Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



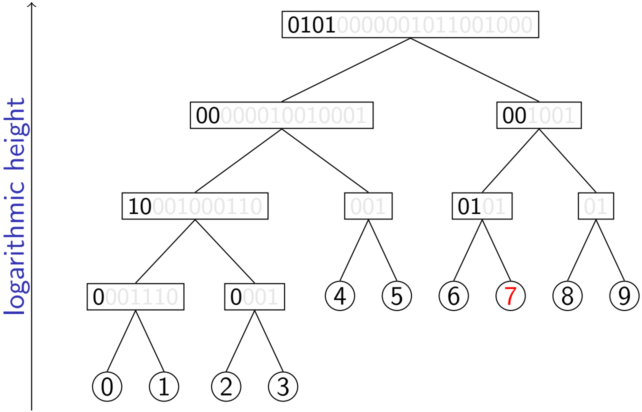
Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



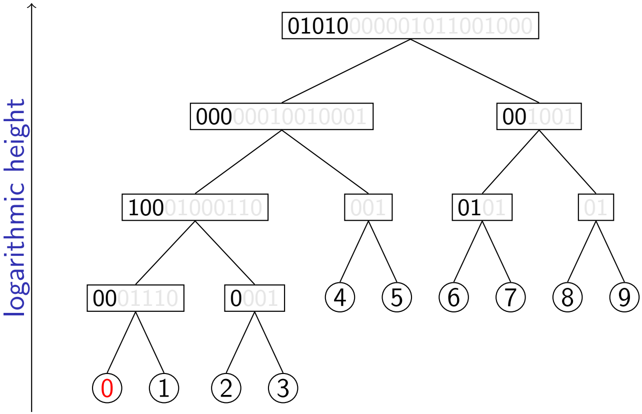
Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



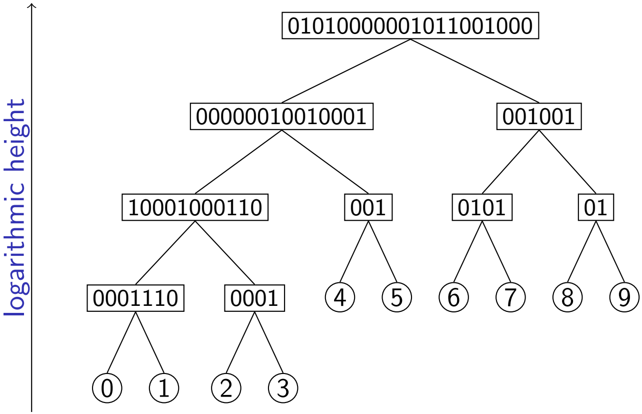
Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



Wavelet Trees [Grossi, Gupta, Vitter; SODA 2003]

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$

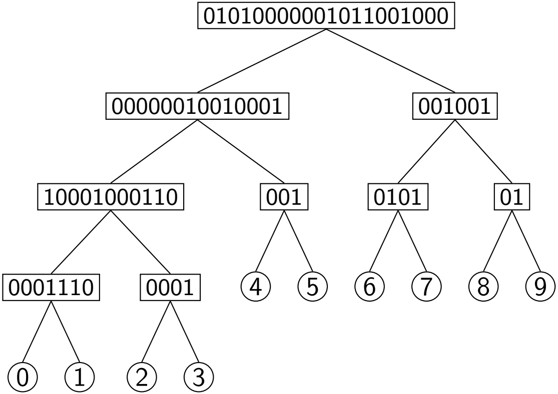


Size $\mathcal{O}(n)$

Construction $\mathcal{O}(n\sqrt{\log n})$

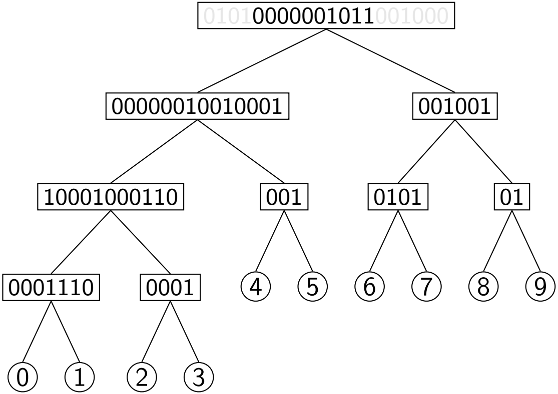
Range Queries with Wavelet Trees

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



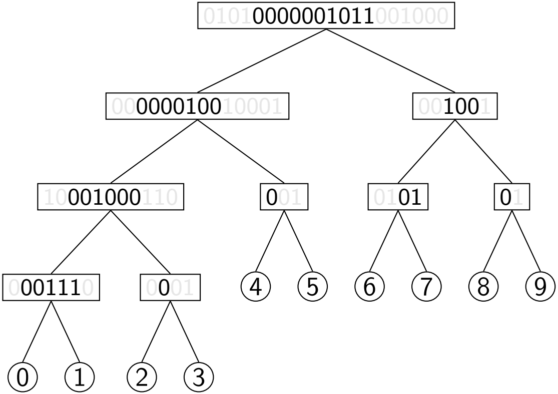
Range Queries with Wavelet Trees

Wavelet tree of $S = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



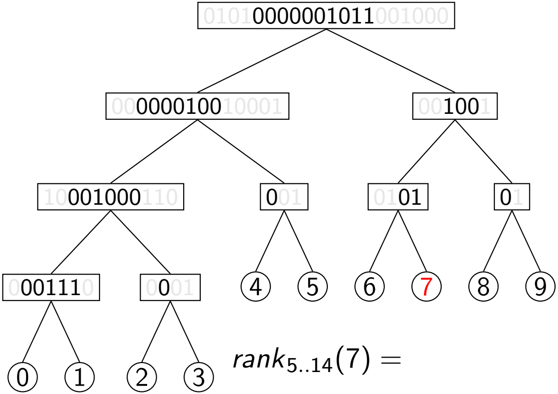
Range Queries with Wavelet Trees

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



Range Queries with Wavelet Trees

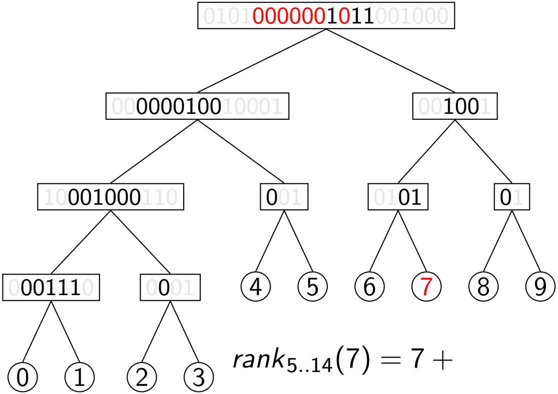
Wavelet tree of $S = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$



$rank_{5..14}(7) =$

Range Queries with Wavelet Trees

Wavelet tree of $\mathrm{S} = \underset{1}{2}\ \underset{2}{6}\ \underset{3}{0}\ \underset{4}{7}\ \underset{5}{0}\ \underset{6}{0}\ \underset{7}{2}\ \underset{8}{1}\ \underset{9}{4}\ \underset{10}{1}\ \underset{11}{8}\ \underset{12}{1}\ \underset{13}{6}\ \underset{14}{7}\ \underset{15}{4}\ \underset{16}{2}\ \underset{17}{9}\ \underset{18}{3}\ \underset{19}{0}\ \underset{20}{5}$

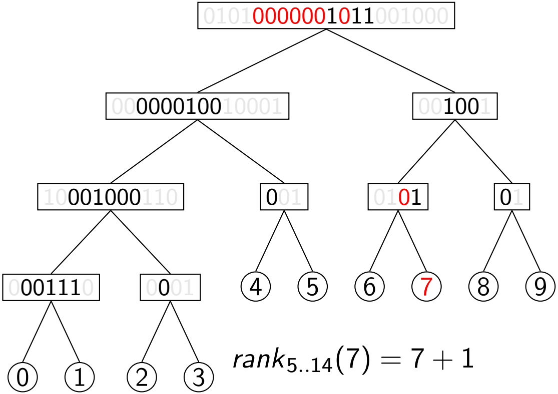


$rank_{5..14}(7) = 7 +$

Range Queries with Wavelet Trees

Wavelet tree of $\mathrm{S} =$

2	6	0	7	0	0	2	1	4	1	8	1	6	7	4	2	9	3	0	5
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

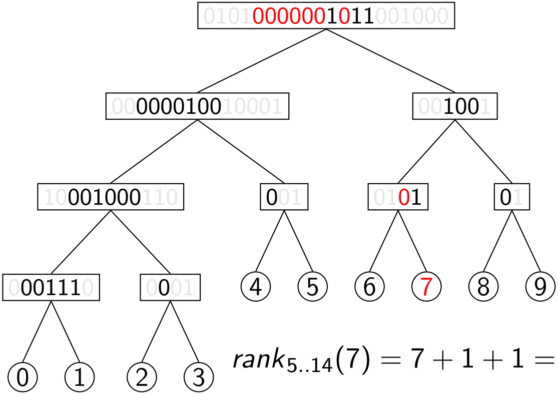


$rank_{5..14}(7) = 7 + 1$

Range Queries with Wavelet Trees

Wavelet tree of S =

2	6	0	7	0	0	2	1	4	1	8	1	6	7	4	2	9	3	0	5
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

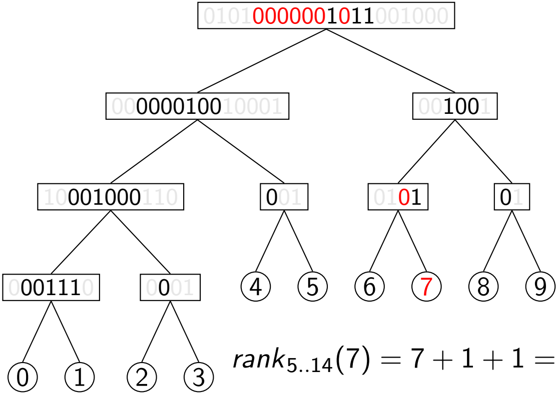


$rank_{5..14}(7) = 7 + 1 + 1 = 9$

Range Queries with Wavelet Trees

Wavelet tree of $\mathrm{S} =$

	2	6	0	7	0	0	2	1	4	1	8	1	6	7	4	2	9	3	0	5
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20



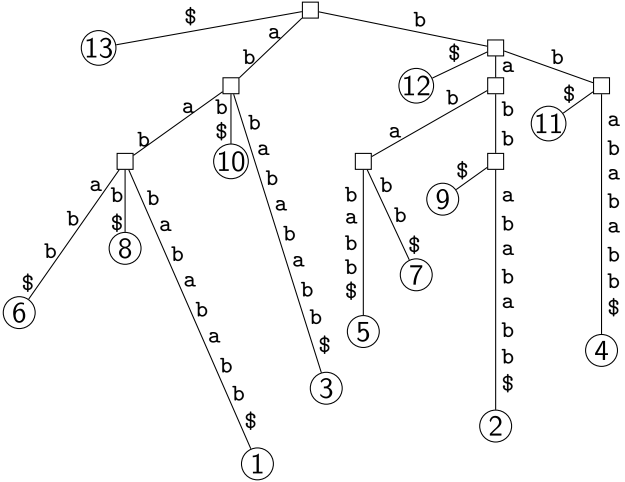
$rank_{5..14}(7) = 7 + 1 + 1 = 9$

Rank $\mathcal{O}(\log n)$ ($\mathcal{O}(\frac{\log n}{\log\log n})$: higher arity)

Select $\mathcal{O}(\log n)$ ($\mathcal{O}(\frac{\log n}{\log\log n})$: higher arity + extra tools)

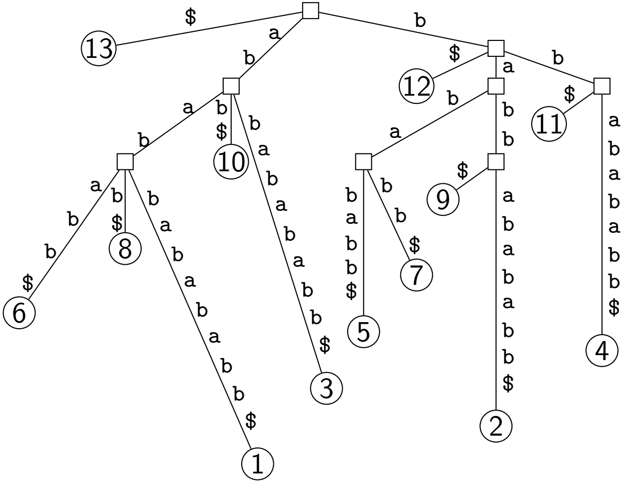
Suffix Trees

Suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\ \underset{2}{\mathtt{b}}\ \underset{3}{\mathtt{a}}\ \underset{4}{\mathtt{b}}\ \underset{5}{\mathtt{b}}\ \underset{6}{\mathtt{a}}\ \underset{7}{\mathtt{b}}\ \underset{8}{\mathtt{a}}\ \underset{9}{\mathtt{b}}\ \underset{10}{\mathtt{a}}\ \underset{11}{\mathtt{b}}\ \underset{12}{\mathtt{b}}$



Suffix Trees

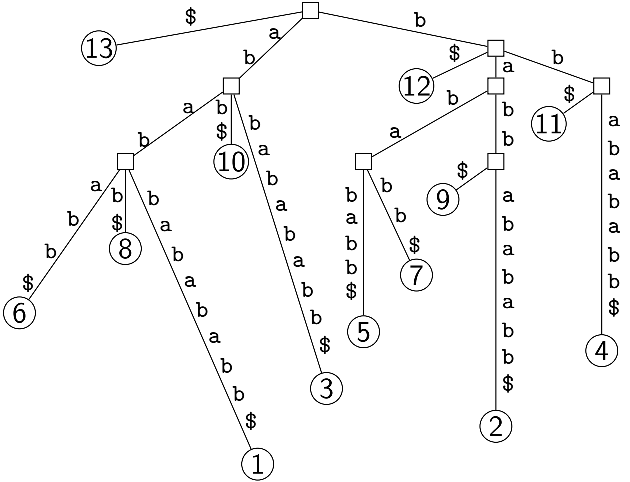
Suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\,\underset{2}{\mathtt{b}}\,\underset{3}{\mathtt{a}}\,\underset{4}{\mathtt{b}}\,\underset{5}{\mathtt{b}}\,\underset{6}{\mathtt{a}}\,\underset{7}{\mathtt{b}}\,\underset{8}{\mathtt{a}}\,\underset{9}{\mathtt{b}}\,\underset{10}{\mathtt{a}}\,\underset{11}{\mathtt{b}}\,\underset{12}{\mathtt{b}}$



- subwords of T and suffixes of $T\$$ in lexicographic order

Suffix Trees

Suffix tree of $\mathrm{T} = \underset{1}{\texttt{a}}\,\underset{2}{\texttt{b}}\,\underset{3}{\texttt{a}}\,\underset{4}{\texttt{b}}\,\underset{5}{\texttt{b}}\,\underset{6}{\texttt{a}}\,\underset{7}{\texttt{b}}\,\underset{8}{\texttt{a}}\,\underset{9}{\texttt{b}}\,\underset{10}{\texttt{a}}\,\underset{11}{\texttt{b}}\,\underset{12}{\texttt{b}}$



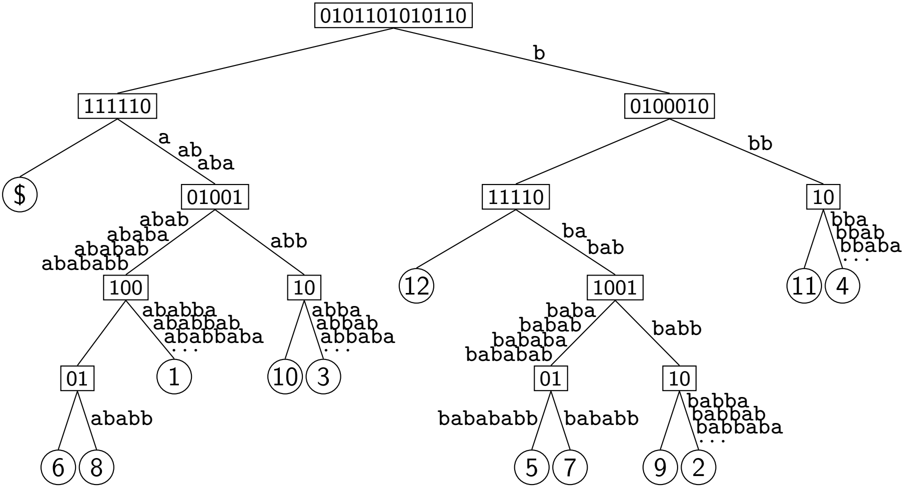
\+ subwords of T and suffixes of $T\$$ in lexicographic order

\- high depth

\- non-uniform arity

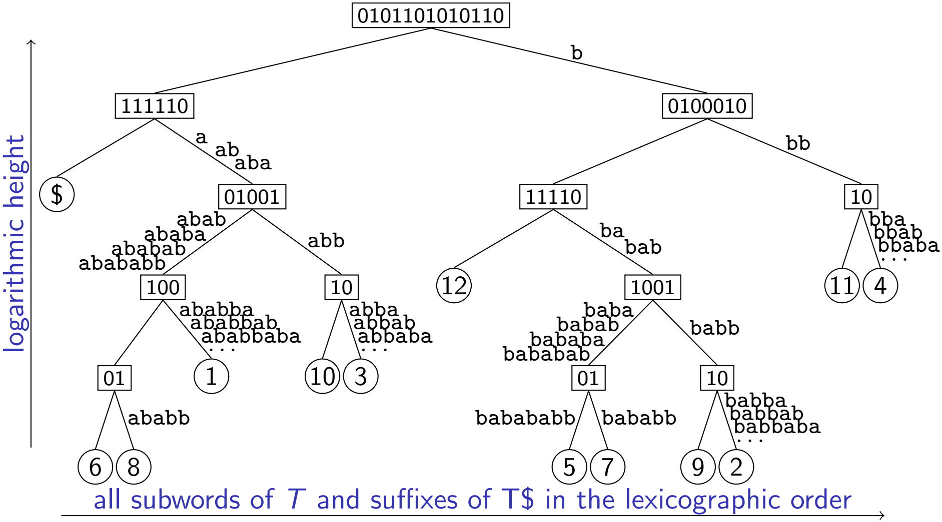
Wavelet Suffix Trees

Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\ \underset{2}{\mathtt{b}}\ \underset{3}{\mathtt{a}}\ \underset{4}{\mathtt{b}}\ \underset{5}{\mathtt{b}}\ \underset{6}{\mathtt{a}}\ \underset{7}{\mathtt{b}}\ \underset{8}{\mathtt{a}}\ \underset{9}{\mathtt{b}}\ \underset{10}{\mathtt{a}}\ \underset{11}{\mathtt{b}}\ \underset{12}{\mathtt{b}}$



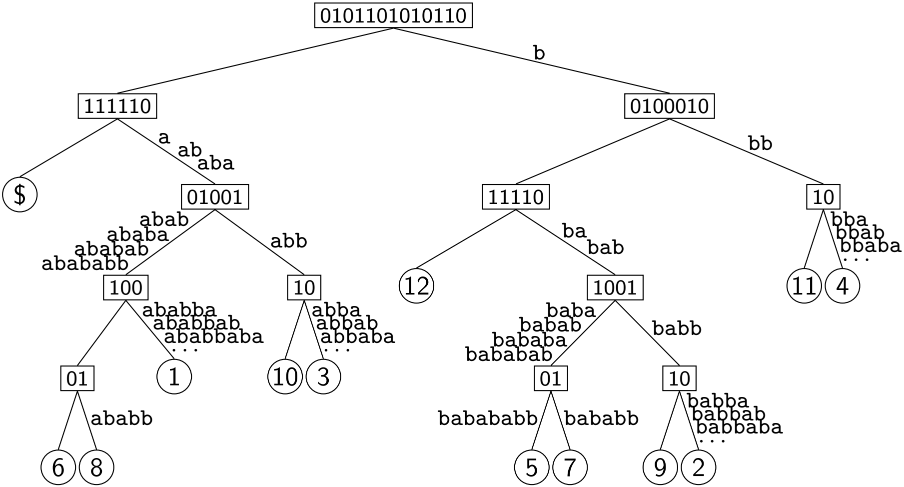
Wavelet Suffix Trees

Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\ \underset{2}{\mathtt{b}}\ \underset{3}{\mathtt{a}}\ \underset{4}{\mathtt{b}}\ \underset{5}{\mathtt{b}}\ \underset{6}{\mathtt{a}}\ \underset{7}{\mathtt{b}}\ \underset{8}{\mathtt{a}}\ \underset{9}{\mathtt{b}}\ \underset{10}{\mathtt{a}}\ \underset{11}{\mathtt{b}}\ \underset{12}{\mathtt{b}}$



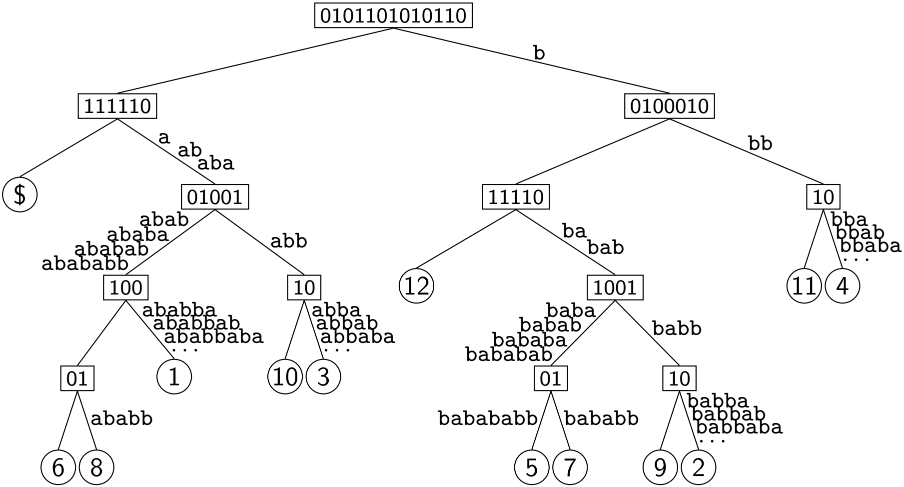
Subword Suffix Queries with Wavelet Suffix Trees

Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\ \underset{2}{\mathtt{b}}\ \underset{3}{\mathtt{a}}\ \underset{4}{\mathtt{b}}\ \underset{5}{\mathtt{b}}\ \underset{6}{\mathtt{a}}\ \underset{7}{\mathtt{b}}\ \underset{8}{\mathtt{a}}\ \underset{9}{\mathtt{b}}\ \underset{10}{\mathtt{a}}\ \underset{11}{\mathtt{b}}\ \underset{12}{\mathtt{b}}$



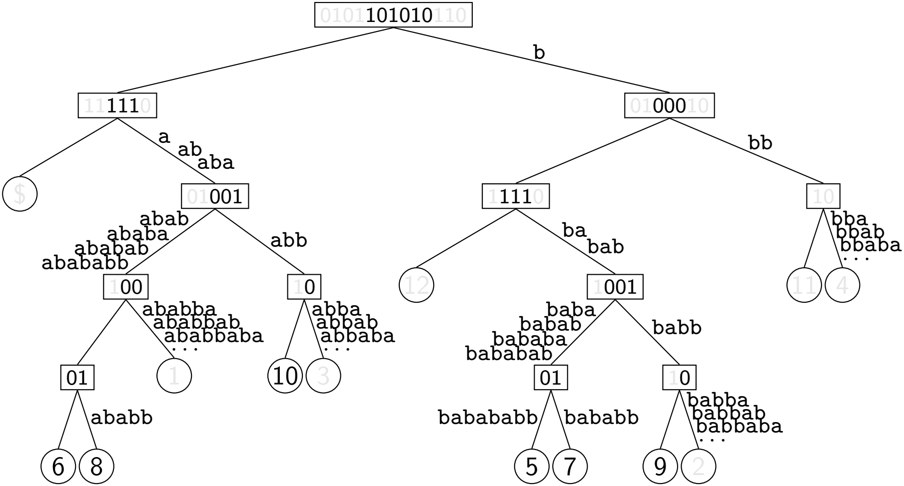
Subword Suffix Queries with Wavelet Suffix Trees

Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\ \underset{2}{\mathtt{b}}\ \underset{3}{\mathtt{a}}\ \underset{4}{\mathtt{b}}\ \underset{5}{\mathtt{b}}\ \underset{6}{\mathtt{a}}\ \underset{7}{\mathtt{b}}\ \underset{8}{\mathtt{a}}\ \underset{9}{\mathtt{b}}\ \underset{10}{\mathtt{a}}\ \underset{11}{\mathtt{b}}\ \underset{12}{\mathtt{b}}$



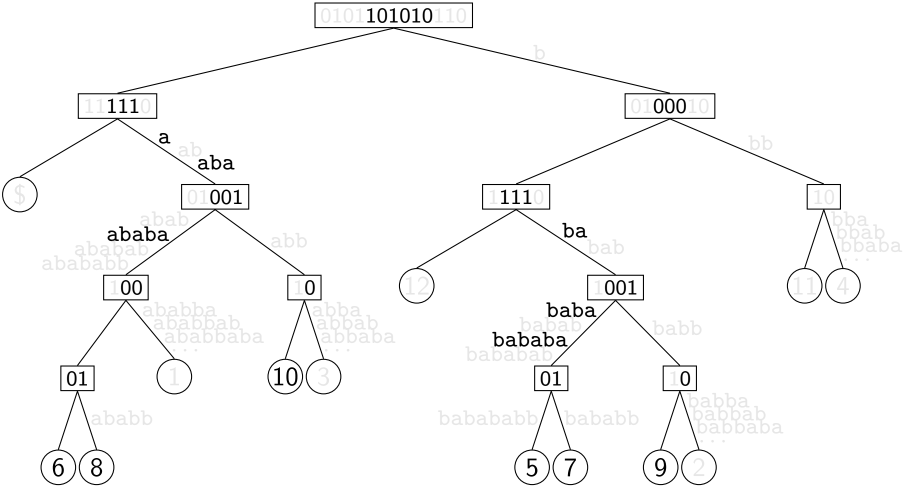
Subword Suffix Queries with Wavelet Suffix Trees

Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\,\underset{2}{\mathtt{b}}\,\underset{3}{\mathtt{a}}\,\underset{4}{\mathtt{b}}\,\underset{5}{\mathtt{b}}\,\underset{6}{\mathtt{a}}\,\underset{7}{\mathtt{b}}\,\underset{8}{\mathtt{a}}\,\underset{9}{\mathtt{b}}\,\underset{10}{\mathtt{a}}\,\underset{11}{\mathtt{b}}\,\underset{12}{\mathtt{b}}$



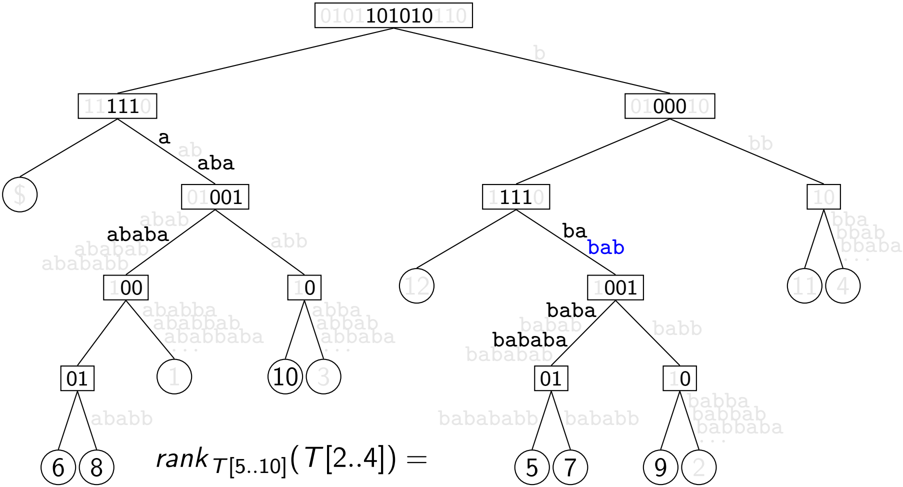
Subword Suffix Queries with Wavelet Suffix Trees

Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\,\underset{2}{\mathtt{b}}\,\underset{3}{\mathtt{a}}\,\underset{4}{\mathtt{b}}\,\underset{5}{\mathtt{b}}\,\underset{6}{\mathtt{a}}\,\underset{7}{\mathtt{b}}\,\underset{8}{\mathtt{a}}\,\underset{9}{\mathtt{b}}\,\underset{10}{\mathtt{a}}\,\underset{11}{\mathtt{b}}\,\underset{12}{\mathtt{b}}$



Subword Suffix Queries with Wavelet Suffix Trees

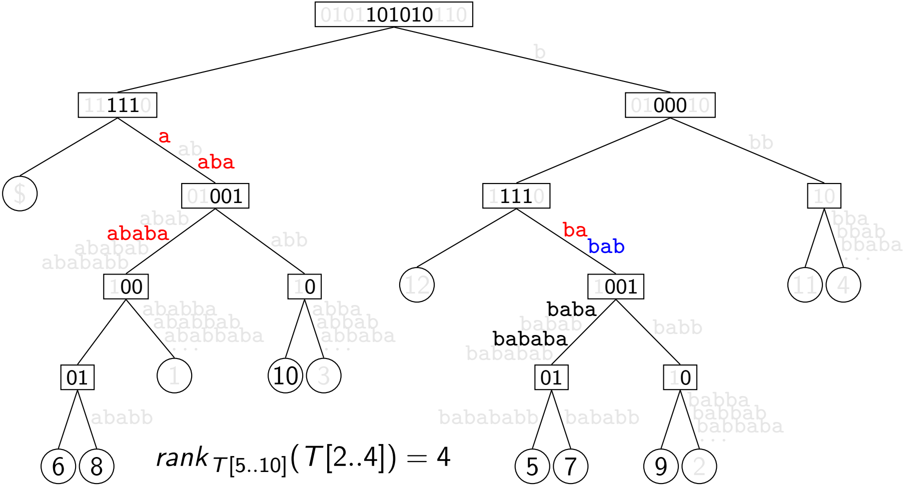
Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\ \underset{2}{\mathtt{b}}\ \underset{3}{\mathtt{a}}\ \underset{4}{\mathtt{b}}\ \underset{5}{\mathtt{b}}\ \underset{6}{\mathtt{a}}\ \underset{7}{\mathtt{b}}\ \underset{8}{\mathtt{a}}\ \underset{9}{\mathtt{b}}\ \underset{10}{\mathtt{a}}\ \underset{11}{\mathtt{b}}\ \underset{12}{\mathtt{b}}$



$$rank_{T[5..10]}(T[2..4]) =$$

Subword Suffix Queries with Wavelet Suffix Trees

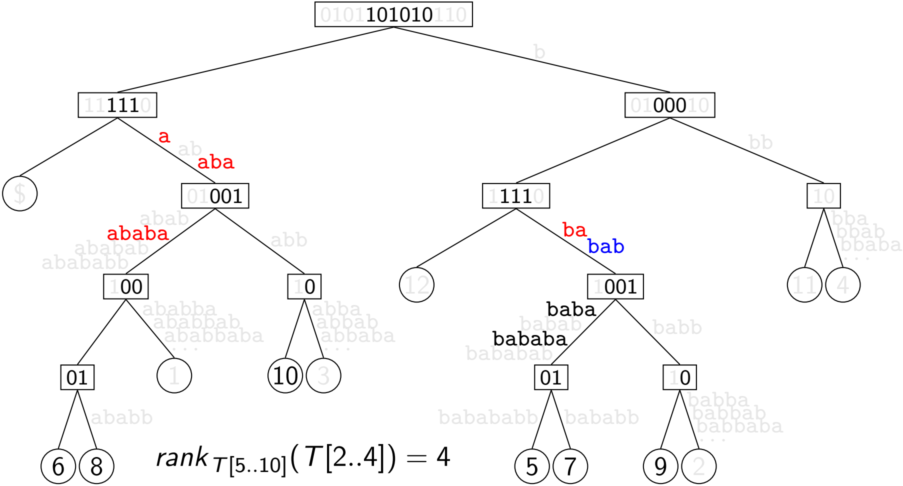
Wavelet suffix tree of $\mathrm{T} = \underset{1}{\mathtt{a}}\,\underset{2}{\mathtt{b}}\,\underset{3}{\mathtt{a}}\,\underset{4}{\mathtt{b}}\,\underset{5}{\mathtt{b}}\,\underset{6}{\mathtt{a}}\,\underset{7}{\mathtt{b}}\,\underset{8}{\mathtt{a}}\,\underset{9}{\mathtt{b}}\,\underset{10}{\mathtt{a}}\,\underset{11}{\mathtt{b}}\,\underset{12}{\mathtt{b}}$



$rank_{T[5..10]}(T[2..4]) = 4$

Subword Suffix Queries with Wavelet Suffix Trees

Wavelet suffix tree of $\mathrm{T} = \underset{1}{a}\,\underset{2}{b}\,\underset{3}{a}\,\underset{4}{b}\,\underset{5}{b}\,\underset{6}{a}\,\underset{7}{b}\,\underset{8}{a}\,\underset{9}{b}\,\underset{10}{a}\,\underset{11}{b}\,\underset{12}{b}$



$$rank_{T[5..10]}(T[2..4]) = 4$$

Operations count suffixes in subtrees, generate suffixes on an edge.

Tools INTERNAL PATTERN MATCHING and bitmasks.

Conclusions & Open Problems

Our results:

- $\mathcal{O}(n\sqrt{\log n})$ construction of wavelet trees,
- simultaneously obtained state-of-the-art construction and query time range selection,
- developed wavelet suffix trees to answer substring suffix rank & selection,
- applied wavelet suffix trees for substring compression with Burrows-Wheeler transform and run-length encoding.

Conclusions & Open Problems

Our results:

- $\mathcal{O}(n\sqrt{\log n})$ construction of wavelet trees,
- simultaneously obtained state-of-the-art construction and query time range selection,
- developed wavelet suffix trees to answer substring suffix rank & selection,
- applied wavelet suffix trees for substring compression with Burrows-Wheeler transform and run-length encoding.

Open problems:

- Can the $\mathcal{O}(n\sqrt{\log n})$ construction time be improved?
  - Would affect counting inversions.

Conclusions & Open Problems

Our results:

- $\mathcal{O}(n\sqrt{\log n})$ construction of wavelet trees,
- simultaneously obtained state-of-the-art construction and query time range selection,
- developed wavelet suffix trees to answer substring suffix rank & selection,
- applied wavelet suffix trees for substring compression with Burrows-Wheeler transform and run-length encoding.

Open problems:

- Can the $\mathcal{O}(n\sqrt{\log n})$ construction time be improved?
  - Would affect counting inversions.
- Are substring suffix queries inherently harder than analogous range queries?
  - Currently $\mathcal{O}(\log n)$ vs $\mathcal{O}(\frac{\log n}{\log\log n})$.

Thank you

Thank you for your attention!