Accessing the Suffix Array via ϕ^{-1}-Forest

Christina Boucher [1]

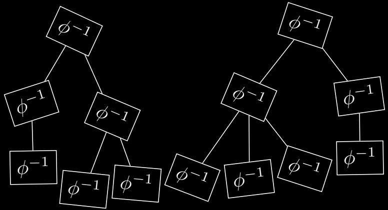
Dominik Köppl [2]

Herman Perera [1]

Massimiliano Rossi [1]

[1]Herbert Wertheim College of Engineering, University of Florida, USA

[2]M&D Data Science Center, Tokyo Medical and Dental University, Japan



extended talk on `https://youtu.be/SjAX1Ru2_gE`

what's it all about?

goal

given the r-index, can we practically speed up the time for retrieving a suffix array (SA) entry?

why interesting?

- r-index is a version of the FM-index with refined SA samples
- while using less space than the FM-index on repetitive texts, the sparse SA samples make random accesses to $\mathrm{SA}[i]$ slow

example

index sequences

- GATTACAT
- GATACAT
- GATTAGATA

for that:

- concatenate with \$, and
- use $\#$ as terminal symbol

input becomes $T = \texttt{GATTACAT\$GATACAT\$GATTAGATA\#}$

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

FM-index for text $T :=$
`GATTACAT$GATACAT$GATTAGATA#`

- uses BWT and a wavelet tree for pattern matching
- counting pattern occurrences works out of the box
- for locating the pattern occurrences, we need SA
- FM index samples SA by text position distance

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

FM-index for text $T :=$ `GATTACAT$GATACAT$GATTAGATA#`

- uses BWT and a wavelet tree for pattern matching
- counting pattern occurrences works out of the box
- for locating the pattern occurrences, we need SA
- FM index samples SA by text position distance

r-index

- only stores SA samples at run boundaries

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

SA access

how do we get $\mathrm{SA}[i]$ with the r-index?

Toehold Lemma, Gagie+'18

$\mathrm{BWT}[i] = \mathrm{BWT}[i+1] \Rightarrow$
$\mathrm{SA}[i+1] - \mathrm{SA}[i] = \mathrm{SA}[j+1] - \mathrm{SA}[j]$ for
$\mathrm{SA}[j] := \mathrm{SA}[i] - 1$

Example

- $\mathrm{SA}[2] = 9$, $\mathrm{SA}[20] = 8$
- $\mathrm{BWT}[2] = \mathrm{BWT}[3] = T$
- $\mathrm{SA}[3] - \mathrm{SA}[2] = \mathrm{SA}[21] - \mathrm{SA}[20]$.

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

Corollary

- let $k \geq 0$ be largest value such that $\mathrm{BWT}[i'] = \mathrm{BWT}[i'+1]$ for all i' with $\mathrm{SA}[i'] \in [\mathrm{SA}[i] - k + 1..\mathrm{SA}[i]]$
- let $\mathrm{SA}[j] := \mathrm{SA}[i] - k$

Then:

- $\mathrm{BWT}[j] \neq \mathrm{BWT}[j+1]$ but still
- $\mathrm{SA}[i+1] - \mathrm{SA}[i] = \mathrm{SA}[j+1] - \mathrm{SA}[j]$.

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

Corollary

- let $k \geq 0$ be largest value such that $\mathrm{BWT}[i'] = \mathrm{BWT}[i'+1]$ for all i' with $\mathrm{SA}[i'] \in [\mathrm{SA}[i] - k + 1..\mathrm{SA}[i]]$
- let $\mathrm{SA}[j] := \mathrm{SA}[i] - k$

Then:

- $\mathrm{BWT}[j] \neq \mathrm{BWT}[j+1]$ but still
- $\mathrm{SA}[i+1] - \mathrm{SA}[i] = \mathrm{SA}[j+1] - \mathrm{SA}[j]$.

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

Corollary

- let $k \geq 0$ be largest value such that $\mathrm{BWT}[i'] = \mathrm{BWT}[i'+1]$ for all i' with $\mathrm{SA}[i'] \in [\mathrm{SA}[i] - k + 1..\mathrm{SA}[i]]$
- let $\mathrm{SA}[j] := \mathrm{SA}[i] - k$

Then:

- $\mathrm{BWT}[j] \neq \mathrm{BWT}[j+1]$ but still
- $\mathrm{SA}[i+1] - \mathrm{SA}[i] = \mathrm{SA}[j+1] - \mathrm{SA}[j]$.

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

Corollary

- let $k \geq 0$ be largest value such that $\mathrm{BWT}[i'] = \mathrm{BWT}[i' + 1]$ for all i' with $\mathrm{SA}[i'] \in [\mathrm{SA}[i] - k + 1..\mathrm{SA}[i]]$
- let $\mathrm{SA}[j] := \mathrm{SA}[i] - k$

Then:

- $\mathrm{BWT}[j] \neq \mathrm{BWT}[j + 1]$ but still
- $\mathrm{SA}[i + 1] - \mathrm{SA}[i] = \mathrm{SA}[j + 1] - \mathrm{SA}[j]$.

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

Corollary

- let $k \geq 0$ be largest value such that $\mathrm{BWT}[i'] = \mathrm{BWT}[i'+1]$ for all i' with $\mathrm{SA}[i'] \in [\mathrm{SA}[i] - k + 1..\mathrm{SA}[i]]$
- let $\mathrm{SA}[j] := \mathrm{SA}[i] - k$

Then:

- $\mathrm{BWT}[j] \neq \mathrm{BWT}[j+1]$ but still
- $\mathrm{SA}[i+1] - \mathrm{SA}[i] = \mathrm{SA}[j+1] - \mathrm{SA}[j]$.

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4	26	A#GATTACAT$GATACAT$GATTAGA	T
5	5	ACAT$GATACAT$GATTAGATA#GAT	T
6	13	ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11	11	ATACAT$GATTAGATA#GATTACAT$	G
12	2	ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15	14	CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21	16	T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

Corollary

- let $k \geq 0$ be largest value such that $\mathrm{BWT}[i'] = \mathrm{BWT}[i'+1]$ for all i' with $\mathrm{SA}[i'] \in [\mathrm{SA}[i]-k+1..\mathrm{SA}[i]]$
- let $\mathrm{SA}[j] := \mathrm{SA}[i] - k$

Then:

- $\mathrm{BWT}[j] \neq \mathrm{BWT}[j+1]$ but still
- $\mathrm{SA}[i+1] - \mathrm{SA}[i] = \mathrm{SA}[j+1] - \mathrm{SA}[j]$.

why it works:
for each backward step, we move to the preceding character pair in the text

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3		$GATTAGATA#GATTACAT$GATACA	T
4		A#GATTACAT$GATACAT$GATTAGA	T
5		ACAT$GATACAT$GATTAGATA#GAT	T
6		ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11		ATACAT$GATTAGATA#GATTACAT$	G
12		ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15		CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21		T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

r-index in $\mathcal{O}(r)$ space

store

- $\mathcal{S}[x]$: sample at start of x-th run
- $\mathcal{E}[x]$: sample at end of x-th run

where $x \in [1..r]$, and r is the number of character runs in BWT.

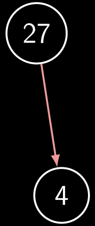
interested in following queries on $\mathcal{E}$:

- $\mathcal{E}.\mathrm{pred}(p) : \max\{q \in \mathcal{E} : q \leq p\}$
- $\mathcal{E}.\mathrm{succ}(p) : \min\{q \in \mathcal{E} : q > p\}$

for that: build predecessor and successor data structure on $\mathcal{E}$

$\mathcal{E}.pred(\mathcal{S}[x+1])$

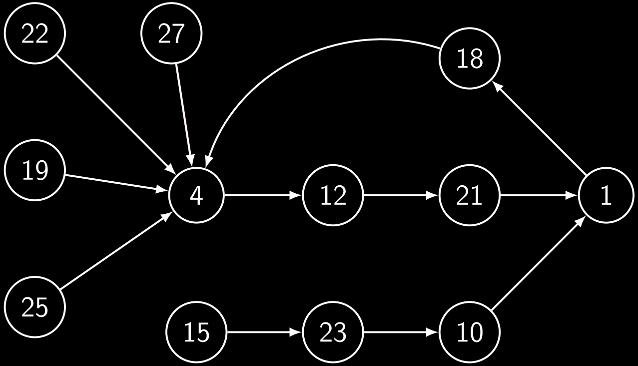
x		$\mathcal{S}[x]$	$\mathcal{E}[x]$
1	27	27	27
2	4	9	22
3	4	7	15
4	23	24	19
5	4	6	23
6	10	10	10
7	1	1	1
8	18	18	18
9	4	8	25
10	4	4	4
11	12	12	12
12	21	21	21
13	1	3	20



ϕ^{-1} graph

- each entry of $\mathcal{E}$ is a node
- create an arc from $\mathcal{E}[x]$ to $\mathcal{E}[y]$) if $\mathcal{E}[y] = \mathcal{E}.\mathrm{pred}(\mathcal{S}[x+1])$ (sorted $\mathcal{E} =$ $[1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27]$)

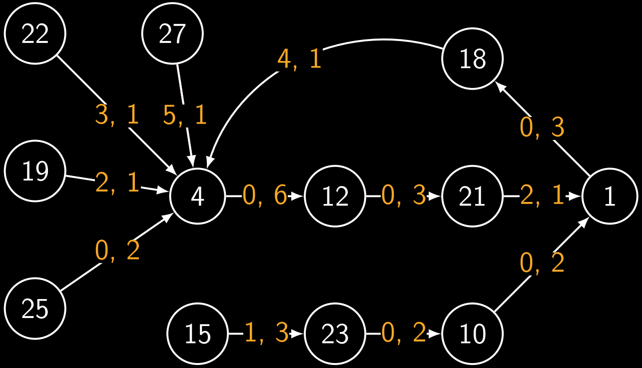
x	$\mathcal{E}.pred(\mathcal{S}[x+1])$	$\mathcal{S}[x]$	$\mathcal{E}[x]$
1	27	27	27
2	4	9	22
3	4	7	15
4	23	24	19
5	4	6	23
6	10	10	10
7	1	1	1
8	18	18	18
9	4	8	25
10	4	4	4
11	12	12	12
12	21	21	21
13	1	3	20



ϕ^{-1} graph

- each entry of $\mathcal{E}$ is a node
- create an arc from $\mathcal{E}[x]$ to $\mathcal{E}[y]$) if $\mathcal{E}[y] = \mathcal{E}.\mathrm{pred}(\mathcal{S}[x+1])$ (sorted $\mathcal{E} =$ $[1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27]$)

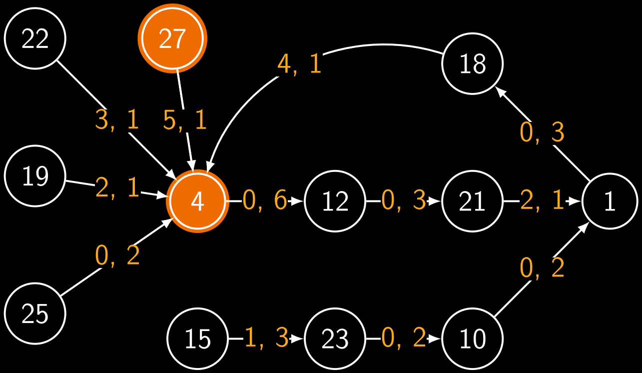
x	$\mathcal{E}.pred(\mathcal{S}[x+1])$	$\mathcal{S}[x]$	$\mathcal{E}[x]$	c_x	ℓ_x
1	27	27	27	5	1
2	4	9	22	3	1
3	4	7	15	1	3
4	23	24	19	2	1
5	4	6	23	0	2
6	10	10	10	0	2
7	1	1	1	0	3
8	18	18	18	4	1
9	4	8	25	0	2
10	4	4	4	0	6
11	12	12	12	0	3
12	21	21	21	0	2
13	1	3	20	-	2



- label the arc of x-th run with cost c_x and limit ℓ_x
- $c_x := \mathcal{S}[x+1] - \mathcal{E}.\mathrm{pred}(\mathcal{S}[x+1])$
- $\ell_x := \mathcal{E}.\mathrm{succ}(\mathcal{E}[x]) - \mathcal{E}[x]$ (sorted $\mathcal{E} =$ $[1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27]$)

to compute $\mathrm{SA}[i+1]$ from $\mathrm{SA}[i]$

1. start node is $p := \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$
2. initial cost c_0 is $\mathrm{SA}[i] - p$
3. stop if accumulated cost c is above limit of arc
4. add cost of arc to c and move to next node v
5. report SA value $v + c$
6. goto 3.

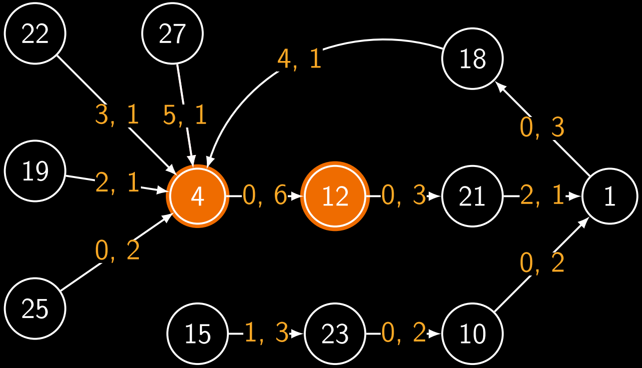


example

- $\mathrm{SA}[1] = 27, p = 27, c \leftarrow c_0 = 0$
- arc $(27, 4)$ has limit $1 > c \Rightarrow$ move to 4
- add cost 5 of arc $(27, 4)$ to $c \leftarrow c_0 + 5 = 5$
- $\mathrm{SA}[2] = 9$ is node label 4 plus cost 5

to compute $\mathrm{SA}[i+1]$ from $\mathrm{SA}[i]$

1. start node is $p := \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$
2. initial cost c_0 is $\mathrm{SA}[i] - p$
3. stop if accumulated cost c is above limit of arc
4. add cost of arc to c and move to next node v
5. report SA value $v + c$
6. goto 3.

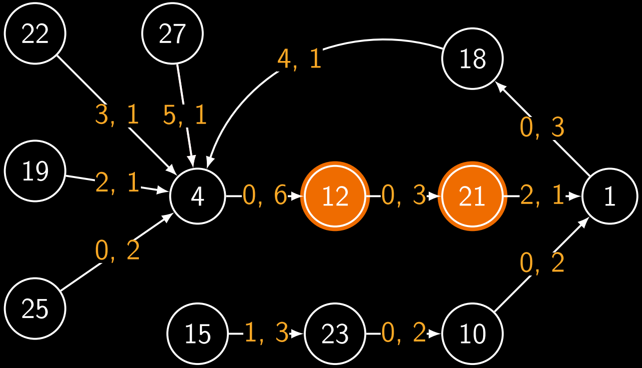


example

- are at node 4 with accumulated cost $c = 5$
- arc $(4, 12)$ has limit $6 > c \Rightarrow$ move to 12
- $\mathrm{SA}[3] = 17$ is node label 12 plus cost 5

to compute $\mathrm{SA}[i+1]$ from $\mathrm{SA}[i]$

1. start node is $p := \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$
2. initial cost c_0 is $\mathrm{SA}[i] - p$
3. stop if accumulated cost c is above limit of arc
4. add cost of arc to c and move to next node v
5. report SA value $v + c$
6. goto 3.

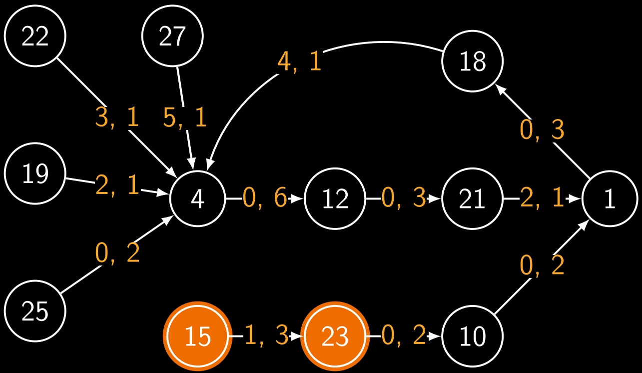


example

- are at node 12 with accumulated cost $c = 5$
- arc $(12, 21)$ has limit $3 < c \Rightarrow$ stop
- continue with $\mathrm{SA}[3] = 17$
- $p = \mathcal{E}.\mathrm{pred}(17) = 15$, $c_0 = 2$.

to compute $\mathrm{SA}[i+1]$ from $\mathrm{SA}[i]$

1. start node is $p := \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$
2. initial cost c_0 is $\mathrm{SA}[i] - p$
3. stop if accumulated cost c is above limit of arc
4. add cost of arc to c and move to next node v
5. report SA value $v + c$
6. goto 3.

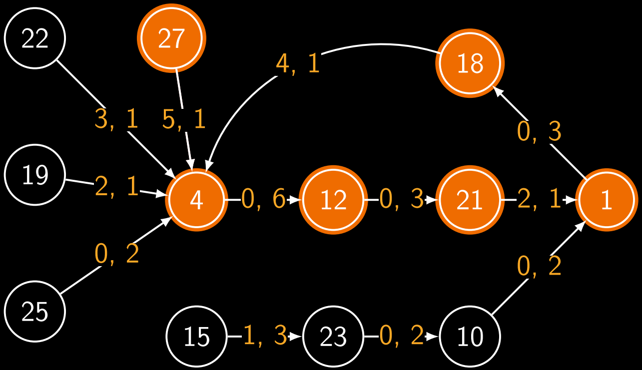


example

- $p = \mathcal{E}.\mathrm{pred}(17) = 15, c \leftarrow c_0 = 2$
- arc $(15, 23)$ has limit $3 > c \Rightarrow$ continue
- add cost 1 of arc $(15, 23)$ to $c \leftarrow c_0 + 1 = 3$
- $\mathrm{SA}[4] = 26$ is node label 23 plus cost 3

to compute $\mathrm{SA}[i+1]$ from $\mathrm{SA}[i]$

1. start node is $p := \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$
2. initial cost c_0 is $\mathrm{SA}[i] - p$
3. stop if accumulated cost c is above limit of arc
4. add cost of arc to c and move to next node v
5. report SA value $v + c$
6. goto 3.

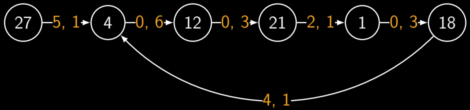


further speedup

- for each traversed arc, we can omit a predecessor query on $\mathcal{E}$
- if such traversable paths are long, we add shortcuts

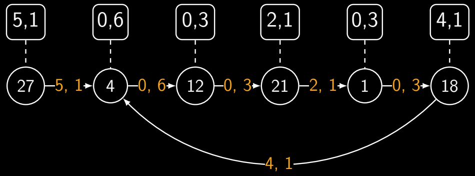
$\Rightarrow$ build ϕ^{-1} trees on long paths

ϕ^{-1} tree



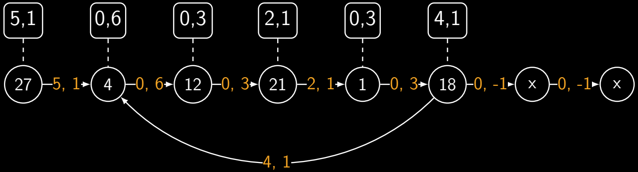
ϕ^{-1} tree

- identify label of out-going arc with node itself



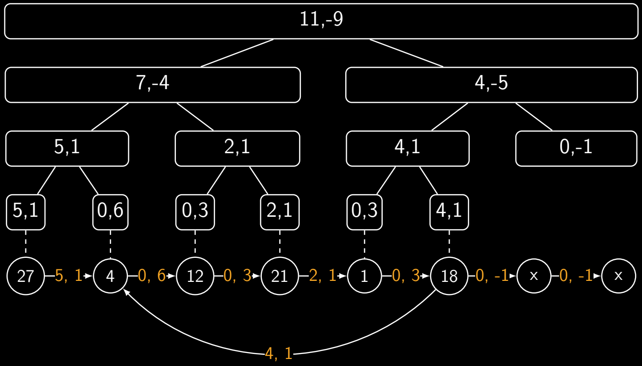
ϕ^{-1} tree

- identify label of out-going arc with node itself
- fill up path with dummy arcs having label $0, -1$ (=untraversable)



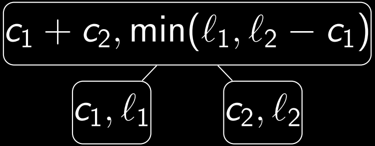
ϕ^{-1} tree

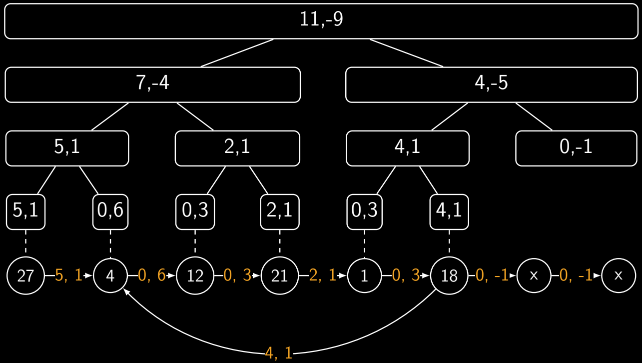
- identify label of out-going arc with node itself
- fill up path with dummy arcs having label $0, -1$ (=untraversable)
- build perfect binary tree on path by partitioning it



ϕ^{-1} tree

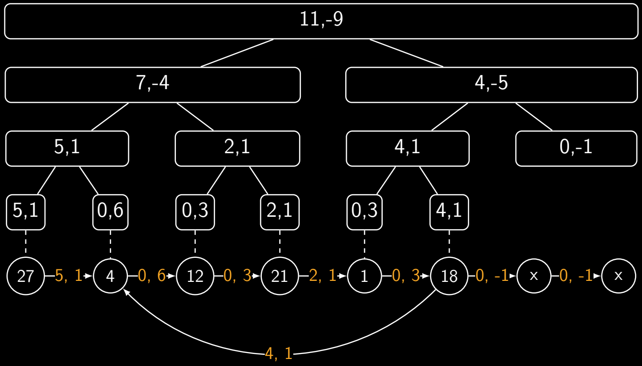
- identify label of out-going arc with node itself
- fill up path with dummy arcs having label $0, -1$ (=untraversable)
- build perfect binary tree on path by partitioning it
- label of internal node is based on the label of its children as below:





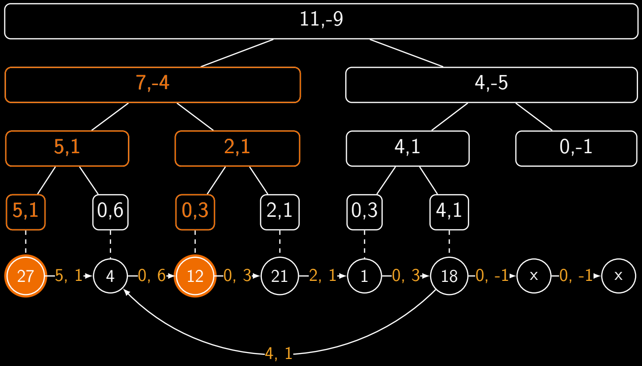
querying ϕ^{-1} tree

1. climb up until exceeding the limit at a node
2. climb down to leaf at which we exceed the limit the first time



querying ϕ^{-1} tree

1. climb up until exceeding the limit at a node
2. climb down to leaf at which we exceed the limit the first time



example

1. start at 27 with cost 0
2. climb up to $7, -4$ with limit $-4 < 0$
3. take cost of left child and descend to $0, 3$
4. return 12 with cost 5, having skipped 4

experiments

dataset

- Chromosome 19 sequences from the 1000 Genomes Project

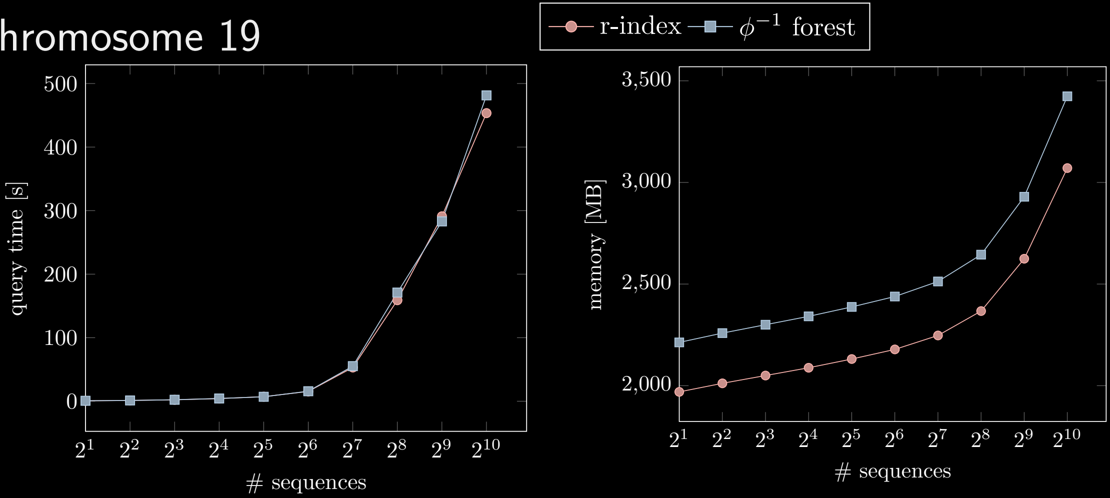
machine

- AMD EPYC 75F3 32-core processor
- 512 GB of RAM
- 64-bit Linux

alternative solutions

- standard *r*-index, Gagie+'18
- *sr*-index, Cobas+'21
- RLCSA, Mäkinen+'10

Chromosome 19



ϕ^{-1} forest has no clear advantage

conclusion

introduced ϕ^{-1} forest

- $\mathcal{O}(r)$ space data structure on top of r-index
- provides random access to SA
- query time depends on
  - length of a run
  - values of costs and limits

open problems

- only trivial bound known
  - $\mathcal{O}(\log r)$ time per ϕ^{-1} tree traversal
  - $\mathcal{O}(\log \log_w(n/r))$ time for a predecessor query
- need theoretical analysis of number of predecessor calls

Thank you for listening. Any questions are welcome!

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3		$GATTAGATA#GATTACAT$GATACA	T
4		A#GATTACAT$GATACAT$GATTAGA	T
5		ACAT$GATACAT$GATTAGATA#GAT	T
6		ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11		ATACAT$GATTAGATA#GATTACAT$	G
12		ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15		CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21		T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

how to compute $\mathrm{SA}[i+1]$ from $\mathrm{SA}[i]$ with the r-index:

- if $\mathrm{SA}[i] = \mathcal{E}[x]$ for an $x \in [1..r]$
- $\Rightarrow$ $\mathrm{SA}[i+1] = \mathcal{S}[x+1]$
- otherwise, take $p = \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$, and let j be such that $\mathrm{SA}[j] = p$
- apply toehold lemma:
- $\mathrm{SA}[i+1] - \mathrm{SA}[i] = \mathrm{SA}[j+1] - \mathrm{SA}[j]$
- we obtain $\mathrm{SA}[j+1]$ from the above case ($\mathrm{SA}[j] \in \mathcal{E}$)

ϕ^{-1}

p	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
$T[p]$	G	A	T	T	A	C	A	T	\$	G	A	T	A	C	A	T	\$	G	A	T	T	A	G	A	T	A	#
$\phi^{-1}[p]$	18	19	20	12	13	14	15	16	17	1	2	21	22	23	24	25	26	8	6	27	3	7	10	11	4	5	9

computing $\mathrm{SA}[i+1]$ from $\mathrm{SA}[i]$ is actually an application of

$$\phi^{-1}[\mathrm{SA}[i]] := \begin{cases} 1 & \text{if } i = n, \\ \mathrm{SA}[i+1] & \text{otherwise} \end{cases}$$

then:

- $\phi^{-1}(\mathcal{E}[x]) = \mathcal{S}[x+1]$
- take $p = \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$, and let j be such that $\mathrm{SA}[j] = p$
- rewrite $\mathrm{SA}[i+1] = \mathrm{SA}[j+1] + \mathrm{SA}[i] - \mathrm{SA}[j]$ as

$$\phi^{-1}(\mathrm{SA}[i]) = \mathrm{SA}[i+1] = \phi^{-1}(p) + \mathrm{SA}[i] - p$$

	i	SA	rotation matrix	BWT
	1	27	#GATTACAT$GATACAT$GATTAGAT	A
$i \rightarrow$	2	9	$GATACAT$GATTAGATA#GATTACA	T
	3		$GATTAGATA#GATTACAT$GATACA	T
	4		A#GATTACAT$GATACAT$GATTAGA	T
	5		ACAT$GATACAT$GATTAGATA#GAT	T
	6		ACAT$GATTAGATA#GATTACAT$GA	T
	7	22	AGATA#GATTACAT$GATACAT$GAT	T
	8	7	AT$GATACAT$GATTAGATA#GATTA	C
	9	15	AT$GATTAGATA#GATTACAT$GATA	C
	10	24	ATA#GATTACAT$GATACAT$GATTA	G
	11		ATACAT$GATTAGATA#GATTACAT$	G
	12		ATTACAT$GATACAT$GATTAGATA#	G
	13	19	ATTAGATA#GATTACAT$GATACAT$	G
	14	6	CAT$GATACAT$GATTAGATA#GATT	A
	15		CAT$GATTAGATA#GATTACAT$GAT	A
	16	23	GATA#GATTACAT$GATACAT$GATT	A
	17	10	GATACAT$GATTAGATA#GATTACAT	$
	18	1	GATTACAT$GATACAT$GATTAGATA	#
	19	18	GATTAGATA#GATTACAT$GATACAT	$
	20	8	T$GATACAT$GATTAGATA#GATTAC	A
	21		T$GATTAGATA#GATTACAT$GATAC	A
	22	25	TA#GATTACAT$GATACAT$GATTAG	A
	23	4	TACAT$GATACAT$GATTAGATA#GA	T
$\phi^{-1}(4) \rightarrow$	24	12	TACAT$GATTAGATA#GATTACAT$G	A
	25	21	TAGATA#GATTACAT$GATACAT$GA	T
	26	3	TTACAT$GATACAT$GATTAGATA#G	A
	27	20	TTAGATA#GATTACAT$GATACAT$G	A

from $\mathrm{SA}[i]$ to $\mathrm{SA}[i+1]$

- $\phi^{-1}(\mathcal{E}[x]) = \mathcal{S}[x+1]$
- take $p = \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$, and let j be such that $\mathrm{SA}[j] = p$
- $\phi^{-1}(\mathrm{SA}[i]) = \phi^{-1}(p) + \mathrm{SA}[i] - p$

example

- $i = 2$, $\mathrm{SA}[2] = 9$ is known
- $4 = \mathcal{E}.\mathrm{pred}(9)$
- $\phi^{-1}(4) = 12$

$\Rightarrow$ $\mathrm{SA}[3] = 12 + 9 - 4 = 17$

(sorted $\mathcal{E} = [1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27]$)

	i	SA	rotation matrix	BWT
	1	27	#GATTACAT\$GATACAT\$GATTAGAT	A
	2	9	\$GATACAT\$GATTAGATA#GATTACA	T
$i \rightarrow$	3	17	\$GATTAGATA#GATTACAT\$GATACA	T
	4		A#GATTACAT\$GATACAT\$GATTAGA	T
	5		ACAT\$GATACAT\$GATTAGATA#GAT	T
	6		ACAT\$GATTAGATA#GATTACAT\$GA	T
	7	22	AGATA#GATTACAT\$GATACAT\$GAT	T
	8	7	AT\$GATACAT\$GATTAGATA#GATTA	C
	9	15	AT\$GATTAGATA#GATTACAT\$GATA	C
$\phi^{-1}(15) \rightarrow$	10	24	ATA#GATTACAT\$GATACAT\$GATTA	G
	11		ATACAT\$GATTAGATA#GATTACAT\$	G
	12		ATTACAT\$GATACAT\$GATTAGATA#	G
	13	19	ATTAGATA#GATTACAT\$GATACAT\$	G
	14	6	CAT\$GATACAT\$GATTAGATA#GATT	A
	15		CAT\$GATTAGATA#GATTACAT\$GAT	A
	16	23	GATA#GATTACAT\$GATACAT\$GATT	A
	17	10	GATACAT\$GATTAGATA#GATTACAT	\$
	18	1	GATTACAT\$GATACAT\$GATTAGATA	#
	19	18	GATTAGATA#GATTACAT\$GATACAT	\$
	20	8	T\$GATACAT\$GATTAGATA#GATTAC	A
	21		T\$GATTAGATA#GATTACAT\$GATAC	A
	22	25	TA#GATTACAT\$GATACAT\$GATTAG	A
	23	4	TACAT\$GATACAT\$GATTAGATA#GA	T
	24	12	TACAT\$GATTAGATA#GATTACAT\$G	A
	25	21	TAGATA#GATTACAT\$GATACAT\$GA	T
	26	3	TTACAT\$GATACAT\$GATTAGATA#G	A
	27	20	TTAGATA#GATTACAT\$GATACAT\$G	A

from $\mathrm{SA}[i]$ to $\mathrm{SA}[i+1]$

- $\phi^{-1}(\mathcal{E}[x]) = \mathcal{S}[x+1]$
- take $p = \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$, and let j be such that $\mathrm{SA}[j] = p$
- $\phi^{-1}(\mathrm{SA}[i]) = \phi^{-1}(p) + \mathrm{SA}[i] - p$

example

- $i = 3$, $\mathrm{SA}[3] = 17$ is known
- $15 = \mathcal{E}.\mathrm{pred}(17)$
- $\phi^{-1}(15) = 24$

$\Rightarrow$ $\mathrm{SA}[4] = 24 + 17 - 15 = 26$

(sorted $\mathcal{E} = [1, 4, 10, 12, 15, 18, 19, 20, 21, 22, 23, 25, 27]$)

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4		A#GATTACAT$GATACAT$GATTAGA	T
5		ACAT$GATACAT$GATTAGATA#GAT	T
6		ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11		ATACAT$GATTAGATA#GATTACAT$	G
12		ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15		CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21		T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

Observation

- \# predecessor queries is bounded by length of run

$\Rightarrow$ practically slow for long runs

Can we compute m iterations of ϕ^{-1} faster?

i	SA	rotation matrix	BWT
1	27	#GATTACAT$GATACAT$GATTAGAT	A
2	9	$GATACAT$GATTAGATA#GATTACA	T
3	17	$GATTAGATA#GATTACAT$GATACA	T
4		A#GATTACAT$GATACAT$GATTAGA	T
5		ACAT$GATACAT$GATTAGATA#GAT	T
6		ACAT$GATTAGATA#GATTACAT$GA	T
7	22	AGATA#GATTACAT$GATACAT$GAT	T
8	7	AT$GATACAT$GATTAGATA#GATTA	C
9	15	AT$GATTAGATA#GATTACAT$GATA	C
10	24	ATA#GATTACAT$GATACAT$GATTA	G
11		ATACAT$GATTAGATA#GATTACAT$	G
12		ATTACAT$GATACAT$GATTAGATA#	G
13	19	ATTAGATA#GATTACAT$GATACAT$	G
14	6	CAT$GATACAT$GATTAGATA#GATT	A
15		CAT$GATTAGATA#GATTACAT$GAT	A
16	23	GATA#GATTACAT$GATACAT$GATT	A
17	10	GATACAT$GATTAGATA#GATTACAT	$
18	1	GATTACAT$GATACAT$GATTAGATA	#
19	18	GATTAGATA#GATTACAT$GATACAT	$
20	8	T$GATACAT$GATTAGATA#GATTAC	A
21		T$GATTAGATA#GATTACAT$GATAC	A
22	25	TA#GATTACAT$GATACAT$GATTAG	A
23	4	TACAT$GATACAT$GATTAGATA#GA	T
24	12	TACAT$GATTAGATA#GATTACAT$G	A
25	21	TAGATA#GATTACAT$GATACAT$GA	T
26	3	TTACAT$GATACAT$GATTAGATA#G	A
27	20	TTAGATA#GATTACAT$GATACAT$G	A

suppose $\mathrm{SA}[i] = \mathcal{E}[x]$, and let

- $\mathcal{E}[x] := \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i])$
- $\mathcal{E}[y] := \mathcal{E}.\mathrm{pred}(\mathrm{SA}[i+1])$

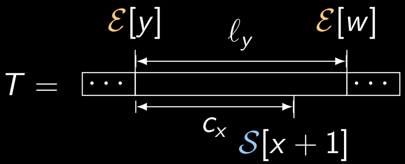
$$\text{then: } \phi^{-1}(\mathrm{SA}[i]) = \mathrm{SA}[i+1] = \mathcal{S}[x+1] \\ = \mathcal{E}[y] + \mathcal{S}[x+1] - \mathcal{E}[y] \\ = \mathcal{E}[y] + c_x$$

where $c_x := \mathcal{S}[x+1] - \mathcal{E}.\mathrm{pred}(\mathcal{S}[x+1])$ is the cost of x-th run.

example

- $i = 1$, $\mathrm{SA}[i] = \mathcal{E}[1] = 27$,
- $\mathcal{S}[2] = 9$, $\mathcal{E}[y] = \mathcal{E}.\mathrm{pred}(\mathcal{S}[2]) = 4$
- $c_1 = \mathcal{S}[2] - \mathcal{E}[y] = 5$; $\phi^{-1}(27) = 4 + 5 = 9$

suppose $\mathrm{SA}[i+1] \notin \mathcal{E}$.
Then:



$$\begin{aligned}\phi^{-1}(\mathrm{SA}[i+1]) &= \phi^{-1}(\phi^{-1}(\mathrm{SA}[i])) \\ &= \phi^{-1}(\mathcal{S}[x+1]) \\ &= \phi^{-1}(\mathcal{E}[y]+c_x)\end{aligned}$$

- apply toehold lemma for $\mathcal{E}.\mathrm{pred}(\mathcal{S}[x+1]) = \mathcal{E}[y]$:

$\Rightarrow$ $\phi^{-1}(\mathcal{E}[y]+c_x) = \phi^{-1}(\mathcal{E}[y]) + c_x$

- finally, $\phi^{-1}(\mathcal{E}[y]) = \mathcal{S}[y+1] = \mathcal{E}[z] + c_y$,
  where $\mathcal{E}[z] := \mathcal{E}.\mathrm{pred}(\mathcal{S}[y+1])$
- total: $\phi^{-1}(\mathrm{SA}[i+1]) = \mathcal{E}[z] + c_x + c_y$

given $\mathcal{E}[w] := \mathcal{E}.\mathrm{succ}(\mathcal{E}[y])$, we assumed that

- $c_x = \mathcal{S}[x+1] - \mathcal{E}[y] < \mathcal{E}[w] - \mathcal{E}[y] = \ell_y$,
  where $\ell_y := \mathcal{E}.\mathrm{succ}(\mathcal{E}[y]) - \mathcal{E}[y]$ is the limit of the y-th run

recursive application

- given $\mathrm{SA}[i]$ with $\mathcal{E}.\mathrm{pred}(\mathrm{SA}[i]) = \mathcal{E}[x_1]$,
- let $c_0 := \mathrm{SA}[i] - \mathcal{E}[x_1]$ and
- $x_1, x_2, \ldots, x_m$ be the indices of the runs we visit, such that $\mathcal{E}.\mathrm{pred}(\mathcal{E}[x_j] + \sum_{k=1}^{j-1} c_{x_k} + c_0) = \mathcal{E}[x_{j+1}]$ for all $j \in [2..m-1]$

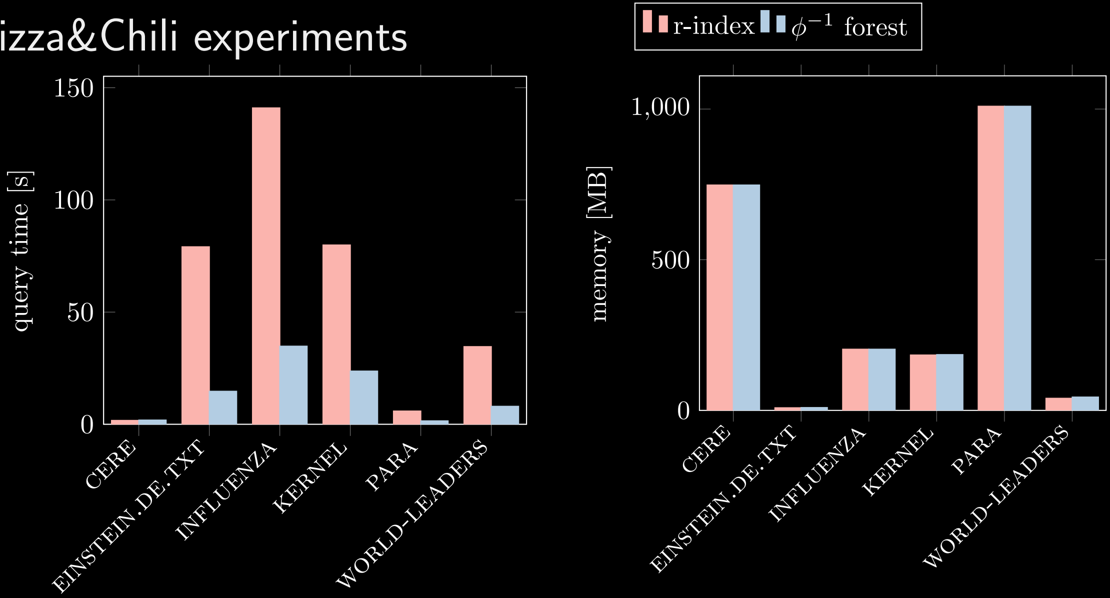
then:

$$\phi^{-m}(\mathrm{SA}[i]) = \mathcal{S}[x_m + 1] = \mathcal{E}[x_m] + \sum_{k=1}^{m-1} c_{x_k} + c_0$$

conclusion:

- just need to sum up $\sum_k c_{x_k}$
- but check that sum does not exceed the limit
- can translate this to a path problem on a directed labeled graph

Pizza&Chili experiments



ϕ^{-1} forest faster with negligible memory overhead