BWT for repetitive texts, part 4b: Toehold lemma & Locate queries

Ben Langmead

For original Keynote files email me ben.langmead@gmail.com

Locate query

Put these in a table called "N"

N

i	0	7	8	10	11	12	14	15	16	17	18	19	20	41	42	44	45	53	54	55	56	57	58	59	60	61	62	63
SA[$q-1$]	22	29	30	32	33	34	36	37	38	39	40	41	42	15	13	16	0	10	18	20	9	63	19	17	11	8	-	21
SA[$q+1$]	47	62	57	61	-	43	42	45	60	55	59	56	0	21	22	24	25	33	34	35	36	37	38	39	40	41	58	46

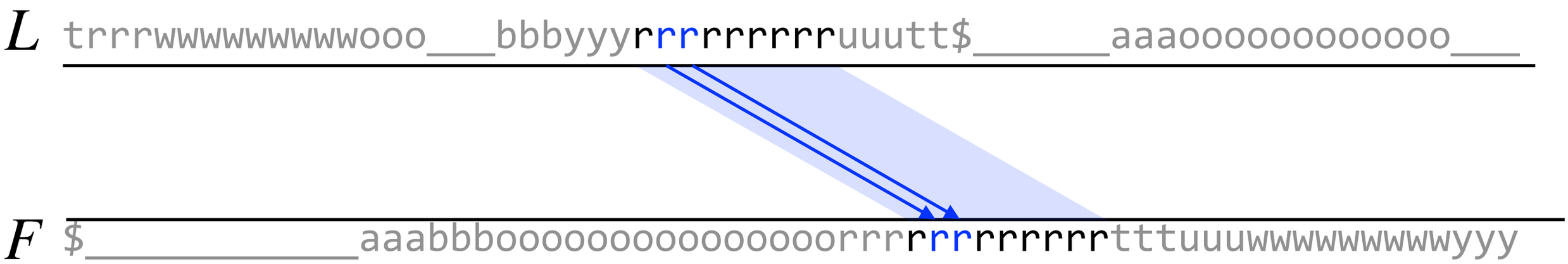
T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

How do we use N to answer arbitrary queries of the form:

Given q, what are SA[$q-1$] & SA[$q+1$]?

Runny LF

Two positions in the same BWT run move through LF in a **parallel** fashion

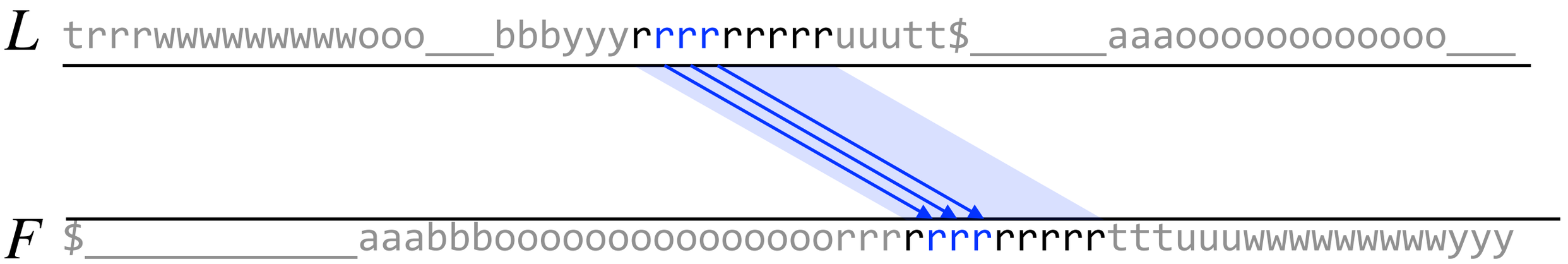


If $\text{BWT}[q] = \text{BWT}[q+1]$

then $\text{LF}[q+1] = \text{LF}[q] + 1$

Runny LF

Three positions in the same BWT run move through LF in a **parallel** fashion

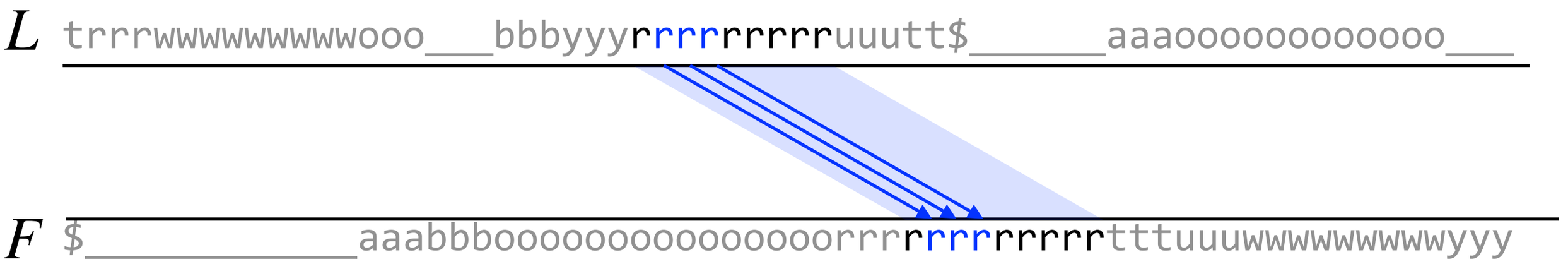


If $\mathrm{BWT}[q-1] = \mathrm{BWT}[q] = \mathrm{BWT}[q+1]$

then $\mathrm{LF}[q+1] = \mathrm{LF}[q] + 1$ & $\mathrm{LF}[q-1] = \mathrm{LF}[q] - 1$

Runny LF

Three positions in the same BWT run move through LF in a **parallel** fashion



If BWT$[q-1] = \text{BWT}[q] = \text{BWT}[q+1]$

If BWT$[q]$ is **not** a run head or tail

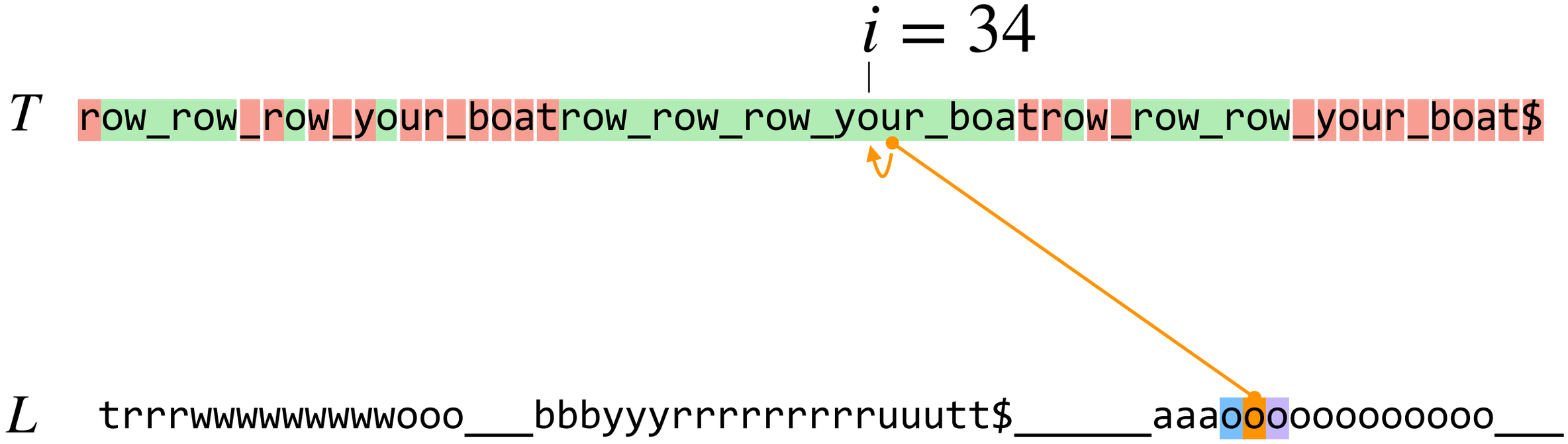
equivalent

then $\text{LF}[q+1] = \text{LF}[q] + 1$ & $\text{LF}[q-1] = \text{LF}[q] - 1$

Locate query

Say we have $q = 50$ and know SA$[q] = 35$ via toehold

We **want** to know SA$[q-1 = 49]$ and SA$[q+1 = 51]$

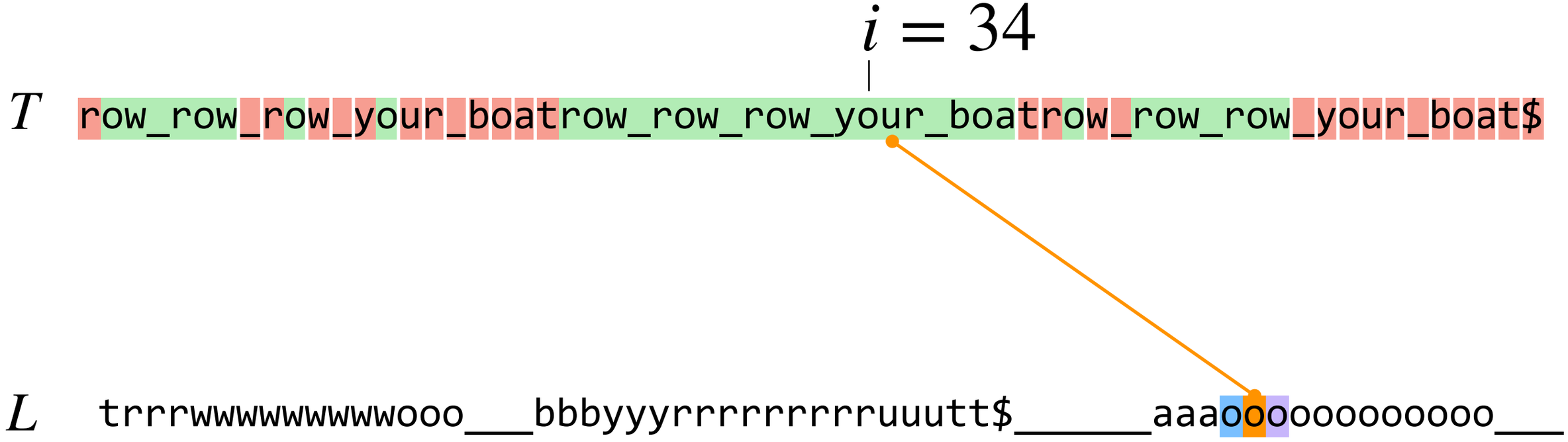


Locate query

Say we have $q = 50$ and know SA$[q] = 35$ via toehold

We **want** to know SA$[q-1 = 49]$ and SA$[q+1 = 51]$

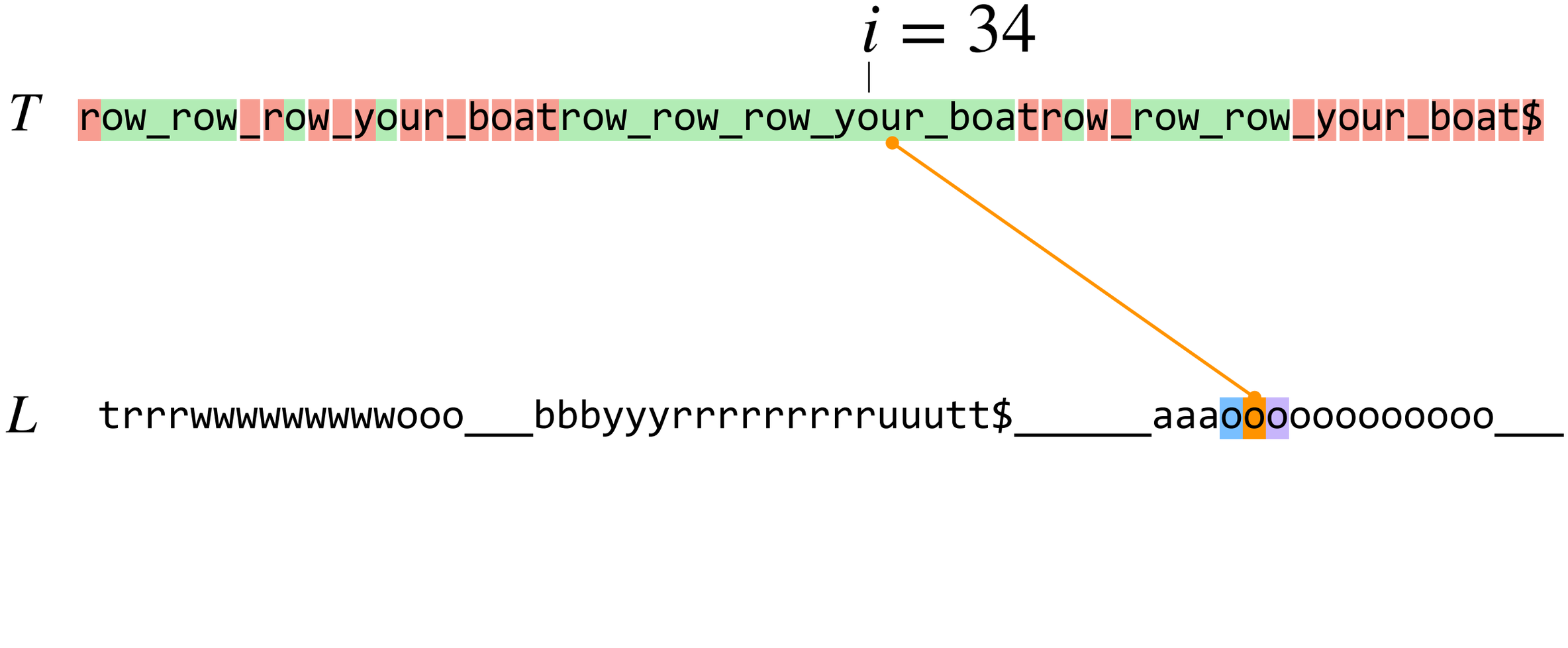
T$[i=34]$ is **not** a phrase head; cannot look up directly



Locate query

Because BWT[50] is **not a run head or tail**

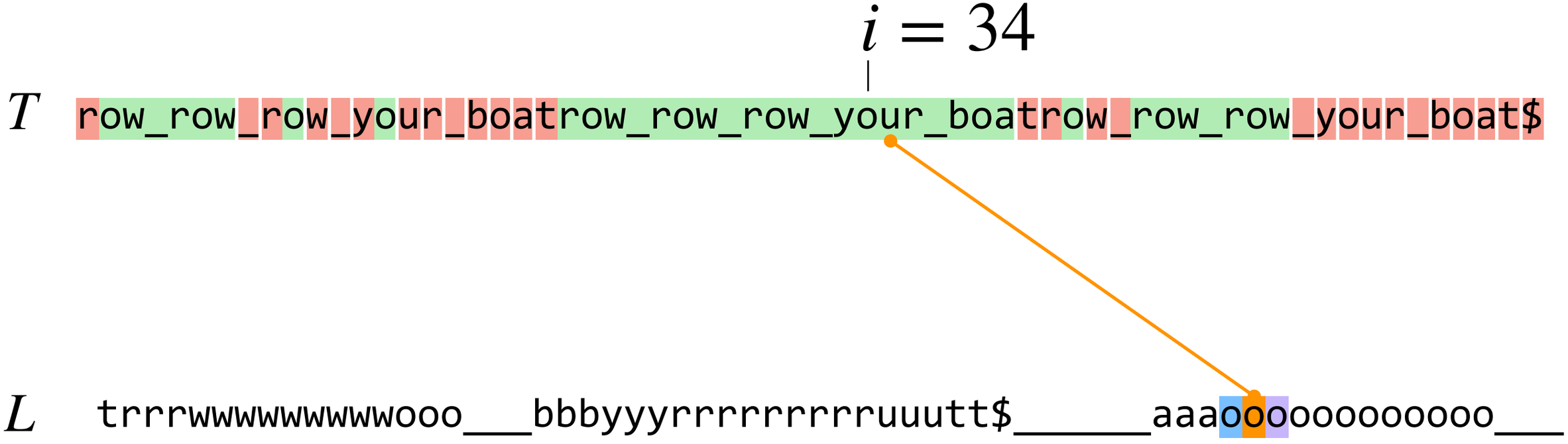
LF[49] LF[50] LF[51] move in parallel



Locate query

Because BWT[50] is
not a run head or tail
LFs move in parallel

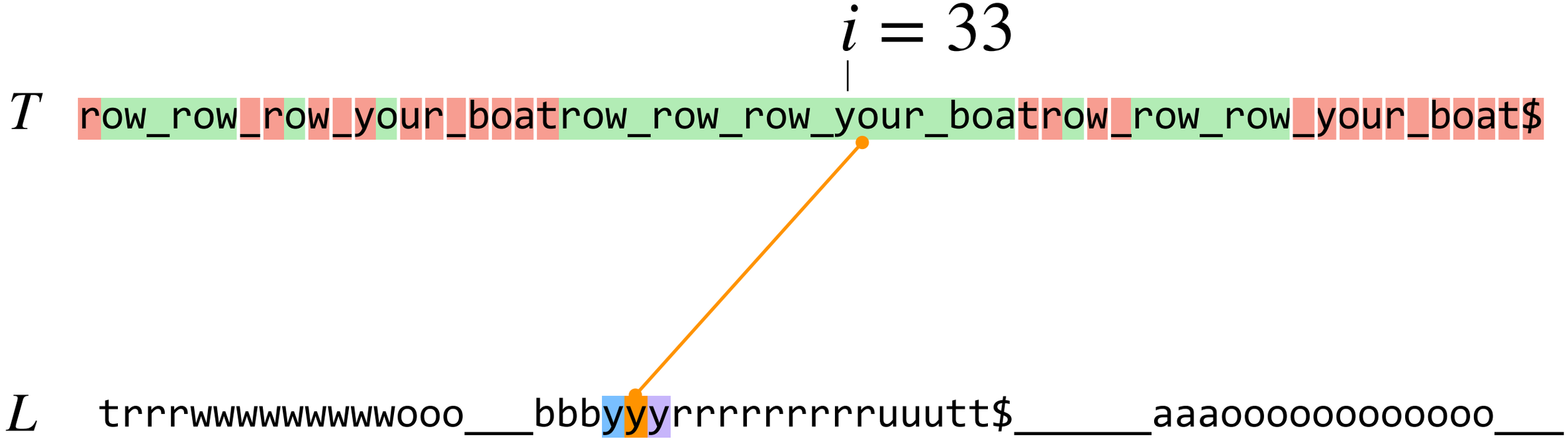
		$SA[..]$		
i	q	$[q-1]$	$[q]$	$[q+1]$
34	50	?	35	?



Locate query

Because BWT[50] is
not a run head or tail
LFs move in parallel

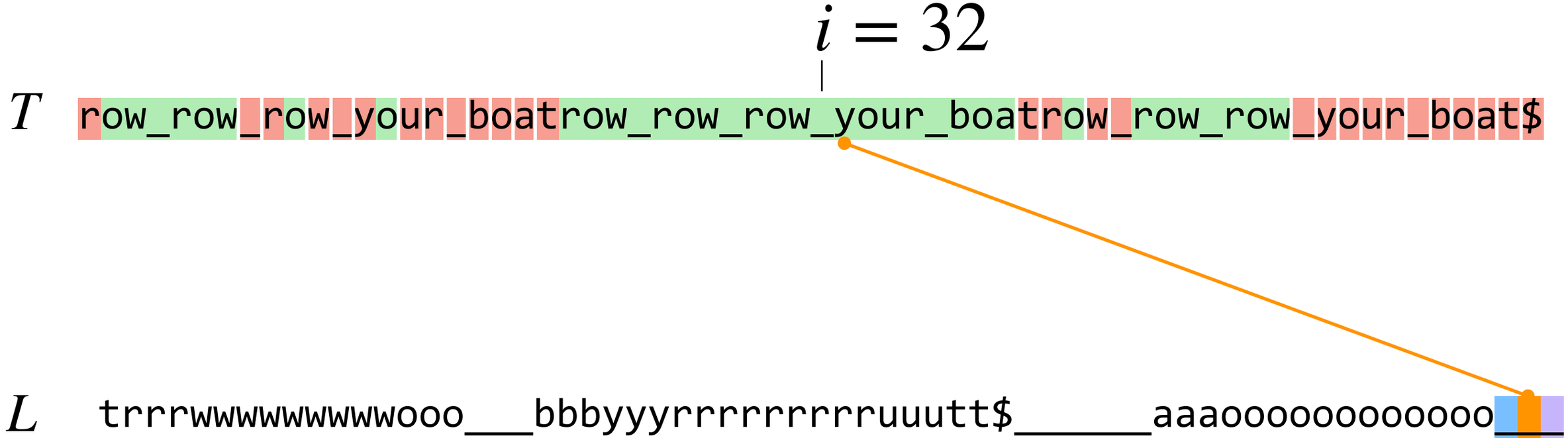
	i	q	$SA[q-1]$	$SA[q]$	$SA[q+1]$
LF	34	50	?	35	?
	33	23	?	34	?



Locate query

Because BWT[50] is
not a run head or tail
LFs move in parallel

	i	q	$SA[\,..\,]$		
			$[q-1]$	$[q]$	$[q+1]$
	34	50	?	35	?
LF	33	23	?	34	?
	32	62	?	33	?



Locate query

Several steps later…

N

i	17	18	19	20	41	42	44	4
SA[$q-1$]	⋯	40	41	42	15	13	16	⋯
SA[$q+1$]	55	59	56	0	21	22	24	2

			$SA[\,..\,]$		
	i	q	$[q-1]$	$[q]$	$[q+1]$
	34	50	?	35	?
	33	23	?	34	?
	32	62	?	33	?
	31	11	?	32	?
	30	59	?	31	?
	29	32	?	30	?
	28	44	?	29	?
	27	8	?	28	?
LFs	26	56	?	27	?
	25	29	?	26	?
	24	41	?	25	?
	23	5	?	24	?
	22	53	?	23	?
	21	26	?	22	?
	20	38	?	21	?

$i = 20$

T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

L `trrrwwwwwwwwwooo___bbbyyyrrrrrrrrruuutt$_____aaaooooooooooooo___`

Locate query

Several steps later…

N

i	17	18	19	20	41	42	44	4
SA[$q-1$]	⋯	40	41	42	15	13	16	⋯
SA[$q+1$]	55	59	56	0	21	22	24	2

LFs

i	q	$SA[\,..\,][q-1]$	$SA[\,..\,][q]$	$SA[\,..\,][q+1]$
34	50	?	35	?
33	23	?	34	?
32	62	?	33	?
31	11	?	32	?
30	59	?	31	?
29	32	?	30	?
28	44	?	29	?
27	8	?	28	?
26	56	?	27	?
25	29	?	26	?
24	41	?	25	?
23	5	?	24	?
22	53	?	23	?
21	26	?	22	?
20	38	42	21	0

$i = 20$

T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

L `trrrwwwwwwwwwooo___bbbyyyrrrrrrrrruuutt$_____aaaooooooooooooo___`

Locate query

Several steps later…

N

i	17	18	19	20	41	42	44	4
SA[$q-1$]	⋯	40	41	42	15	13	16	⋯
SA[$q+1$]	55	59	56	0	21	22	24	2

LFs

i	q	$SA[\,..\,]$ $[q-1]$	$[q]$	$[q+1]$
34	50	56	35	14
33	23	55	34	13
32	62	54	33	12
31	11	53	32	11
30	59	52	31	10
29	32	51	30	9
28	44	50	29	8
27	8	49	28	7
26	56	48	27	6
25	29	47	26	5
24	41	46	25	4
23	5	45	24	3
22	53	44	23	2
21	26	43	22	1
20	38	42	21	0

$i = 20$

T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

L `trrrwwwwwwwwwooo___bbbyyyrrrrrrrrruuutt$_____aaaooooooooooooo___`

Locate query

Several steps later…

N

i	17	18	19	20	41	42	44	4
SA[$q-1$]	⋯	40	41	42	15	13	16	⋯
SA[$q+1$]	55	59	56	0	21	22	24	2

We **want** to know

SA[$q-1 = 49$] and

SA[$q+1 = 51$]

i	q	$SA[\,.\,.\,][q-1]$	$SA[\,.\,.\,][q]$	$SA[\,.\,.\,][q+1]$
34	50	56	35	14
33	23	55	34	13
32	62	54	33	12
31	11	53	32	11
30	59	52	31	10
29	32	51	30	9
28	44	50	29	8
27	8	49	28	7
26	56	48	27	6
25	29	47	26	5
24	41	46	25	4
23	5	45	24	3
22	53	44	23	2
21	26	43	22	1
20	38	42	21	0

LFs

Answer:

SA[49] = Answer from N + LF steps = 42 + 14 = 56

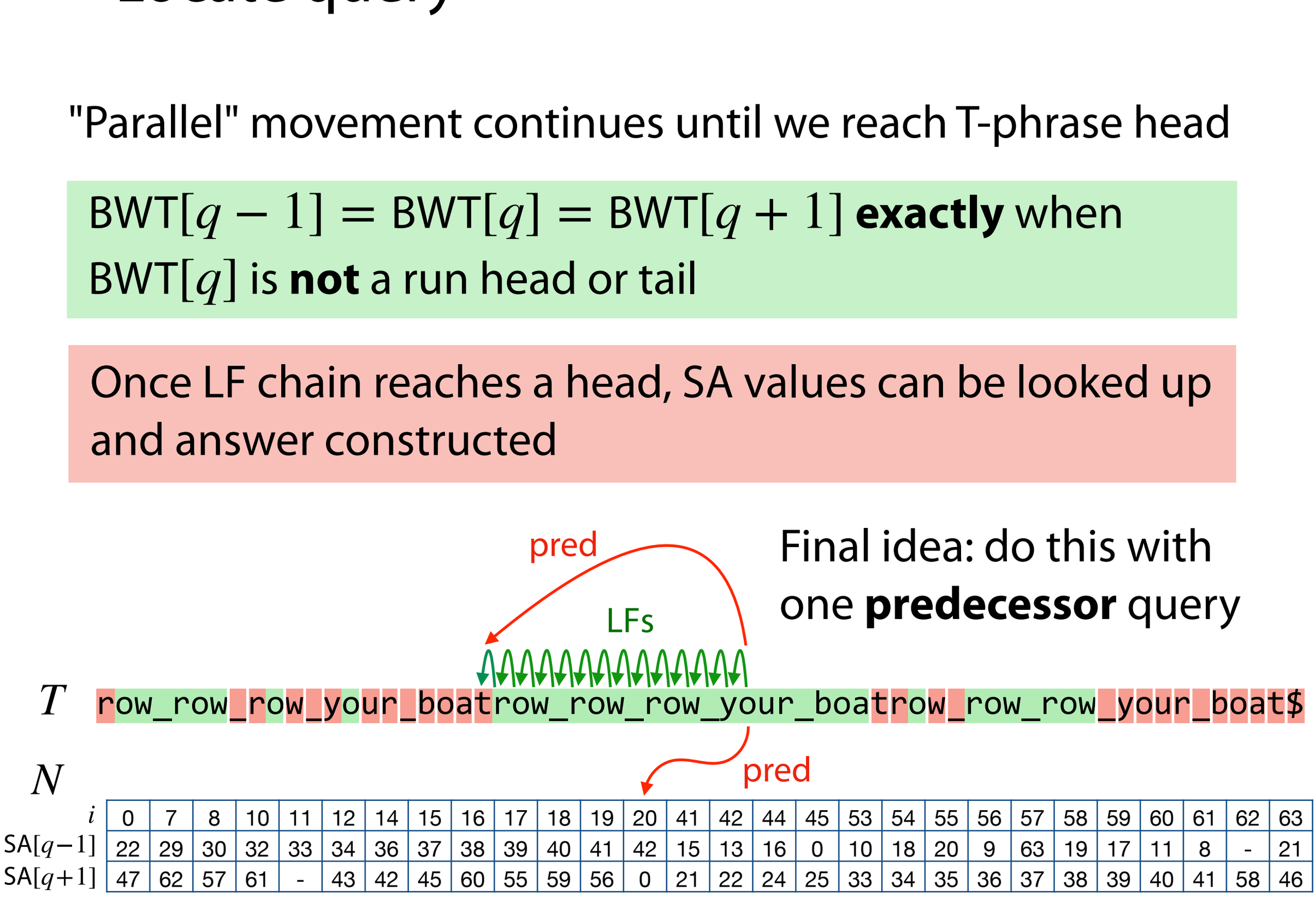
SA[51] = Answer from N + LF steps = 0 + 14 = 14

Locate query

"Parallel" movement continues until we reach T-phrase head

BWT$[q-1]$ = BWT$[q]$ = BWT$[q+1]$ **exactly** when BWT$[q]$ is **not** a run head or tail

Once LF chain reaches a head, SA values can be looked up and answer constructed



i	0	7	8	10	11	12	14	15	16	17	18	19	20	41	42	44	45	53	54	55	56	57	58	59	60	61	62	63
SA$[q-1]$	22	29	30	32	33	34	36	37	38	39	40	41	42	15	13	16	0	10	18	20	9	63	19	17	11	8	-	21
SA$[q+1]$	47	62	57	61	-	43	42	45	60	55	59	56	0	21	22	24	25	33	34	35	36	37	38	39	40	41	58	46

Locate query

N

i	0	7	8	10	11	12	14	15	16	17	18	19	20	41	42	44	45	53	54	55	56	57	58	59	60	61	62	63
SA[$q-1$]	22	29	30	32	33	34	36	37	38	39	40	41	42	15	13	16	0	10	18	20	9	63	19	17	11	8	-	21
SA[$q+1$]	47	62	57	61	-	43	42	45	60	55	59	56	0	21	22	24	25	33	34	35	36	37	38	39	40	41	58	46

T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

Say $q =$ ⬚ $i =$ ⬚ and SA$[q] =$ ⬚

$i' \leftarrow \text{pred}(i)$ ⬚

$x, y \leftarrow N[i']$ ⬚

$k \leftarrow i - i'$ ⬚

$\text{SA}[q-1] = x + k$ ⬚

$\text{SA}[q+1] = y + k$ ⬚

Locate query

N

i	0	7	8	10	11	12	14	15	16	17	18	19	20	41	42	44	45	53	54	55	56	57	58	59	60	61	62	63
SA[$q-1$]	22	29	30	32	33	34	36	37	38	39	40	41	42	15	13	16	0	10	18	20	9	63	19	17	11	8	-	21
SA[$q+1$]	47	62	57	61	-	43	42	45	60	55	59	56	0	21	22	24	25	33	34	35	36	37	38	39	40	41	58	46

T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

Say $q =$ 34 $i =$ 50 and SA$[q] =$ 35

$i' \leftarrow \text{pred}(i)$

$x, y \leftarrow N[i']$

$k \leftarrow i - i'$

$\text{SA}[q-1] = x + k$

$\text{SA}[q+1] = y + k$

Locate query

N

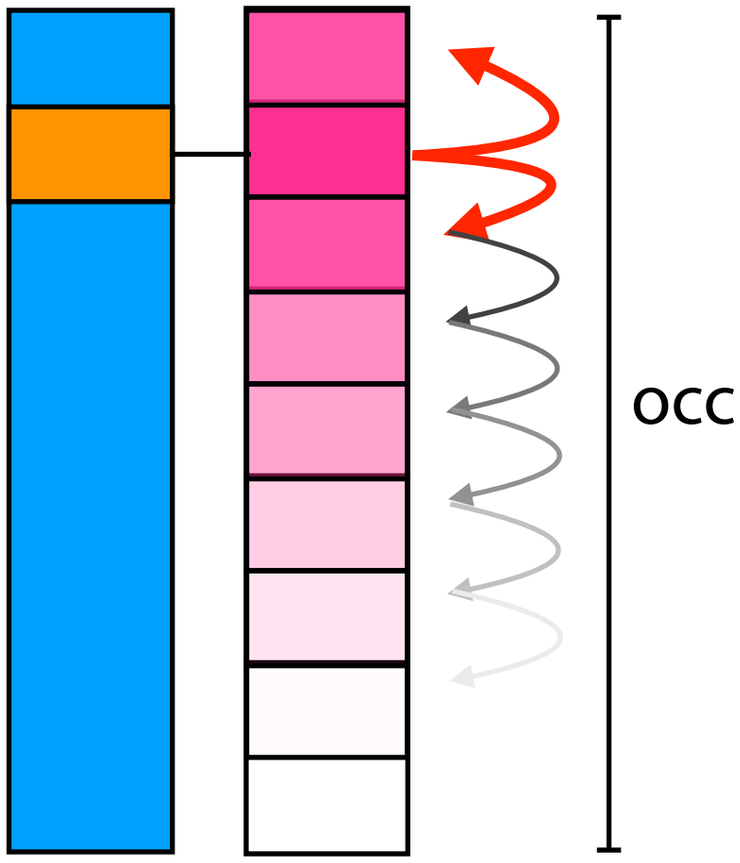
i	0	7	8	10	11	12	14	15	16	17	18	19	20	41	42	44	45	53	54	55	56	57	58	59	60	61	62	63
SA[$q-1$]	22	29	30	32	33	34	36	37	38	39	40	41	42	15	13	16	0	10	18	20	9	63	19	17	11	8	-	21
SA[$q+1$]	47	62	57	61	-	43	42	45	60	55	59	56	0	21	22	24	25	33	34	35	36	37	38	39	40	41	58	46

T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

Say $q =$ 34 $i =$ 50 and SA$[q] =$ 35

$i' \leftarrow \text{pred}(i)$	$i' \leftarrow 20$
$x, y \leftarrow N[i']$	$x \leftarrow 42 \quad y \leftarrow 0$
$k \leftarrow i - i'$	$k \leftarrow 34 - 20 = 14$
$\text{SA}[q-1] = x + k$	$42 + 14 = 56$
$\text{SA}[q+1] = y + k$	$0 + 14 = 14$

O(r) locate queries



One such query uses $O(\log \log n/r)$ time $O(r)$ space

Propagates SA knowledge out by one step

Must repeat $O(\text{occ})$ times where occ = # of occurrences

$O(\text{occ} \cdot \log \log n/r)$ time overall

Putting it together

N

i	0	7	8	10	11	12	14	15	16	17	18	19	20	41	42	44	45	53	54	55	56	57	58	59	60	61	62	63
SA[$q-1$]	22	29	30	32	33	34	36	37	38	39	40	41	42	15	13	16	0	10	18	20	9	63	19	17	11	8	-	21
SA[$q+1$]	47	62	57	61	-	43	42	45	60	55	59	56	0	21	22	24	25	33	34	35	36	37	38	39	40	41	58	46

T `row_row_row_your_boatrow_row_row_your_boatrow_row_row_your_boat$`

Samples at T-phrase heads + predecessor structure

R

$	39					
_	16	18	40	45	61	63
a	46	48				
b	19	21				
o	13	15	49	60		

r	1	3	25	33
t	0	37	38	
u	34	36		
w	4	12		
y	22	24		

L `trrrwwwwwwwwwooo___bbbyyyyrrrrrrrrruuutt$_____aaaooooooooooo___`

Samples at run heads & tails + alphabet-stratified predecessor structure

Putting it together

N

i	0	7	8	10	11	12	14	15	16	17	18	19	20	41	42	44	45	53	54	55	56	57	58	59	60	61	62	63
SA[$q-1$]	22	29	30	32	33	34	36	37	38	39	40	41	42	15	13	16	0	10	18	20	9	63	19	17	11	8	-	21
SA[$q+1$]	47	62	57	61	-	43	42	45	60	55	59	56	0	21	22	24	25	33	34	35	36	37	38	39	40	41	58	46

$O(r)$ space $O(\text{occ} \cdot \log\log(n/r))$ time for all occ locate queries

R

$	39					
_	16	18	40	45	61	63
a	46	48				
b	19	21				
o	13	15	49	60		

r	1	3	25	33
t	0	37	38	
u	34	36		
w	4	12		
y	22	24		

$O(r)$ space $O(\log\log(\sigma + n/r))$ time per backward search iteration keeping same $O(m \log\log(\sigma + n/r))$ bound

Overall backward search + locate in $O(r)$ space

$O(m \log\log(\sigma + n/r) + \text{occ} \cdot \log\log(n/r))$

Putting it together

Index	Space	**Locate time**
Kreft and Navarro [2013, Thm. 4.11]	$O(z)$	$O(m^2h + (m + occ)\log z)$
Christiansen and Ettienne [2018, Thm. 2(3)]	$O(z\log(n/z))$	$O(m + \log^\epsilon z + occ(\log^\epsilon z + \log\log n))$
Christiansen and Ettienne [2018, Thm. 2(1)]	$O(z\log(n/z) + z\log\log z)$	$O(m + occ(\log^\epsilon z + \log\log n))$
Bille et al. [2018, Cor. 1]	$O(z\log(n/z)\log\log z)$	$O(m(1 + \log^\epsilon z/\log(n/z)) + occ\log\log n)$
Bille et al. [2018, Cor. 1]	$O(z(\log(n/z) + \log\log z)\log\log z)$	$O(m + occ\log\log n)$
Claude and Navarro [2012, Thm. 1]	$O(g)$	$O(m^2\log\log_g n + (m + occ)\log g)$
Gagie et al. [2012, Thm. 4]	$O(g + z\log\log z)$	$O(m^2 + (m + occ)\log\log n)$
Mäkinen et al. [2010, Thm. 20]	$O(r + n/s)$	$O((m + s\cdot occ)(\frac{\log\sigma}{\log\log r} + (\log\log n)^2))$
Belazzougui et al. [2015a, Thm. 3]	$O(\overline{r} + z)$	$O(m(\log z + \log\log n) + occ(\log^\epsilon z + \log\log n))$
This paper (Thm. 3.6)	$O(r)$	$O((m + occ)\log\log_w(\sigma + n/r))$
This paper (Thm. 4.10)	$O(r\log\log_w(\sigma + n/r))$	$O(m + occ)$
This paper (Thm. 4.11)	$O(rw\log_\sigma\log_w n)$	$O(\lceil m\log(\sigma)/w\rceil + occ)$
Belazzougui and Cunial [2017a, Thm. 1]	$O(e)$	$O(m + occ)$

we discussed the first one

Gagie, T., Navarro, G., & Prezza, N. (2020). Fully functional suffix trees and optimal text searching in BWT-runs bounded space. Journal of the ACM (JACM), 67(1), 1-54.