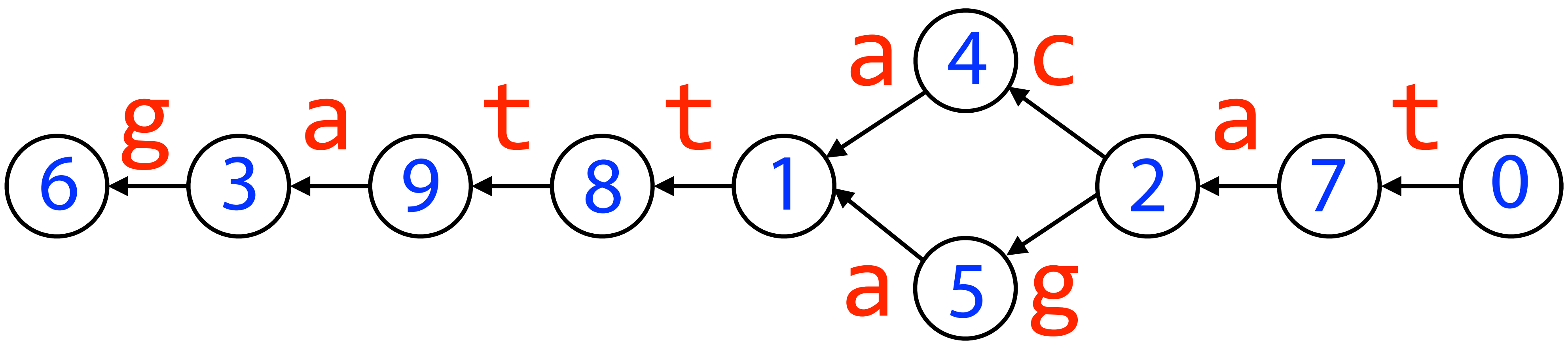
Wheeler graphs, part 5: Data structures

Ben Langmead

Johns Hopkins Whiting School of Engineering

Department of Computer Science

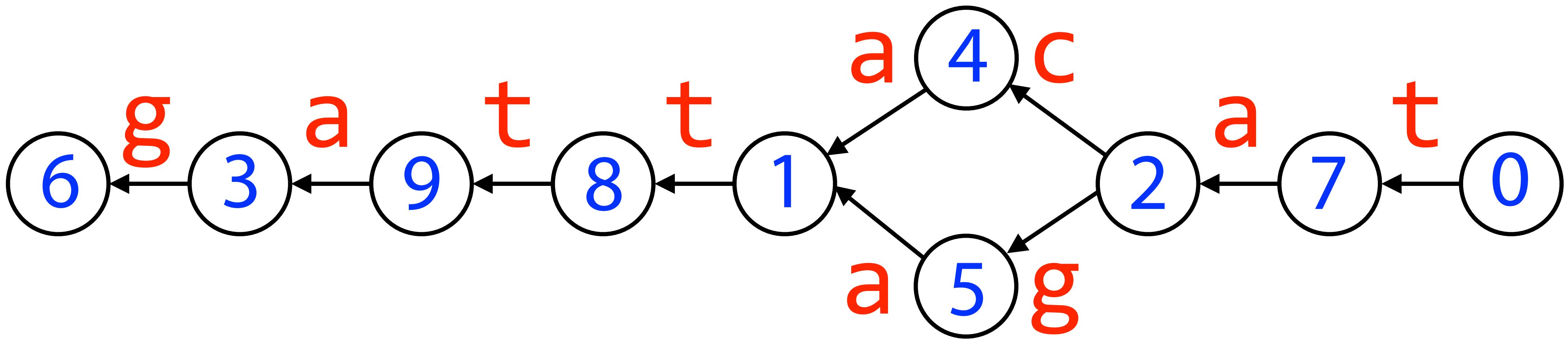
Wheeler graphs



Can we represent a Wheeler graph with ***bitvectors***?

🤔🤔🤔

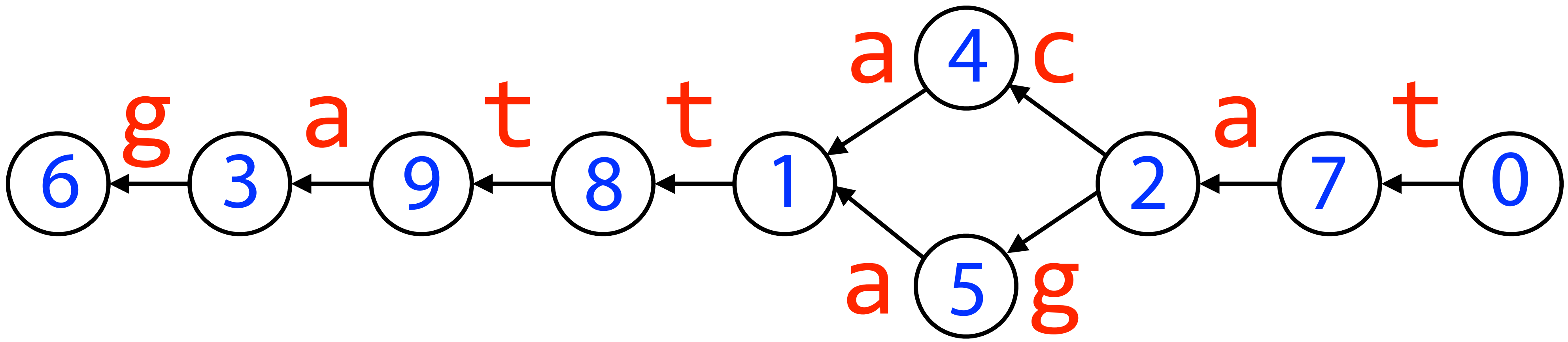
Wheeler graphs



Idea 1: Encode in- and outdegree of each node in unary

#	Unary

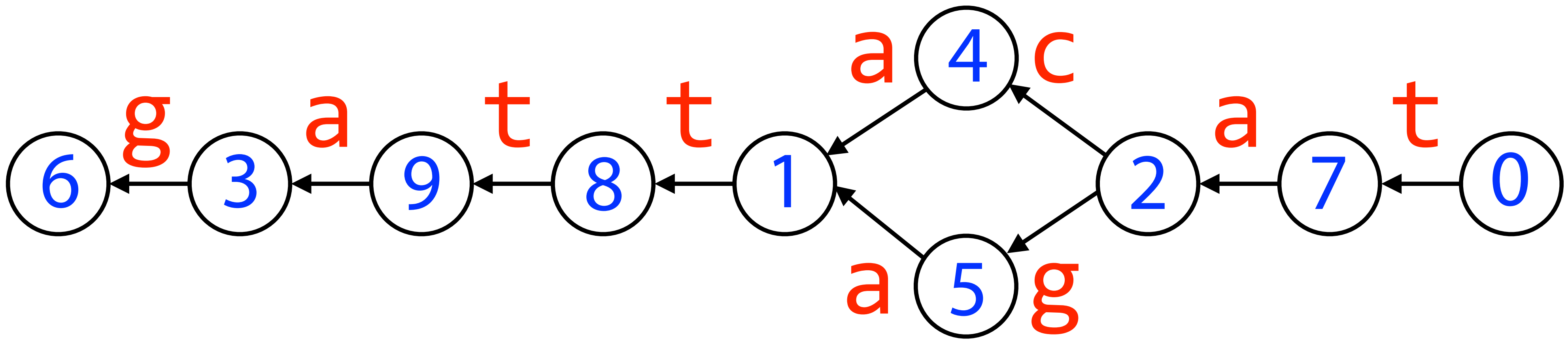
Wheeler graphs



Idea 1: Encode in- and outdegree of each node in unary

#	Unary
0	1
1	01
2	001
3	0001

Wheeler graphs



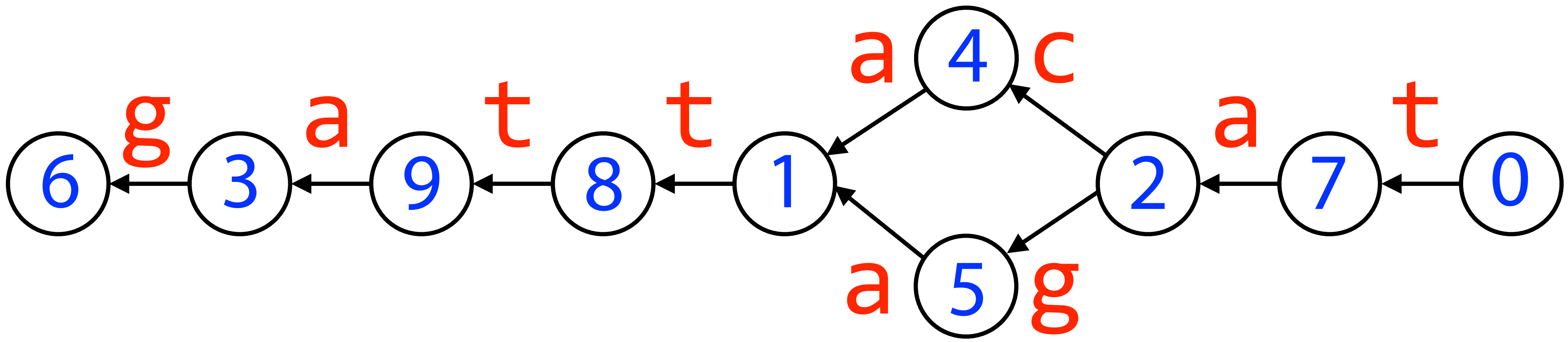
#	Unary
0	1
1	01
2	001
3	0001

Idea 1: Encode in- and outdegree of each node in unary

Idea 2: Concatenate in order by node

Outdegree bitvector $O =$

Wheeler graphs



Idea 1: Encode in- and outdegree of each node in unary

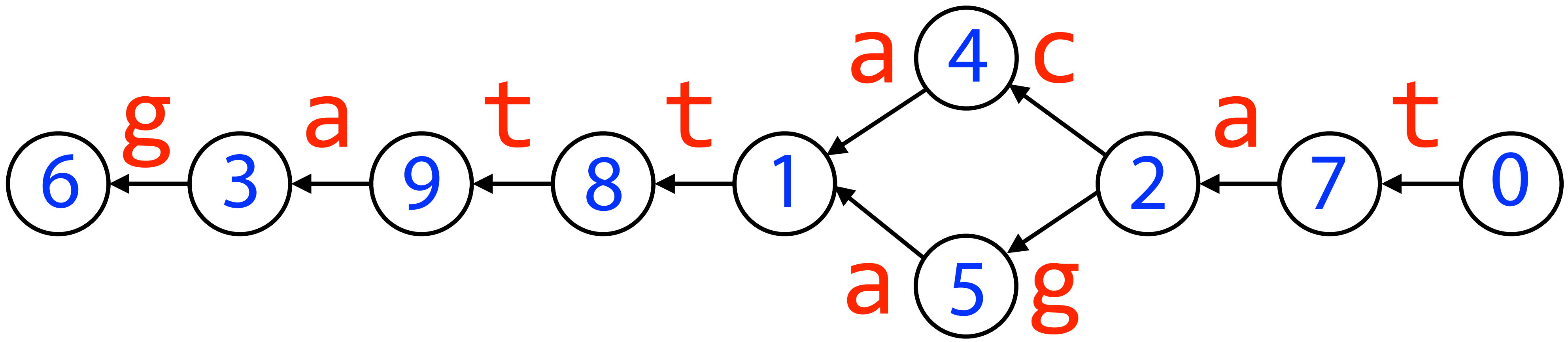
#	Unary
0	1
1	01
2	001
3	0001

Idea 2: Concatenate in order by node

Outdegree bitvector $O =$ `010100101010101101010 1`

Nodes:

Wheeler graphs



Idea 1: Encode in- and outdegree of each node in unary

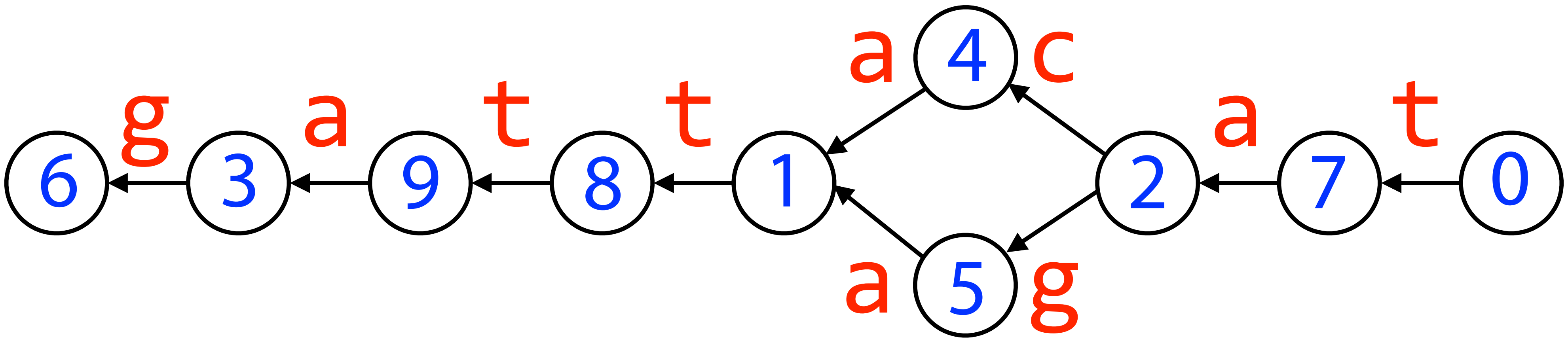
#	Unary
0	1
1	01
2	001
3	0001

Idea 2: Concatenate in order by node

Outdegree bitvector $O =$ 01010010101011010101

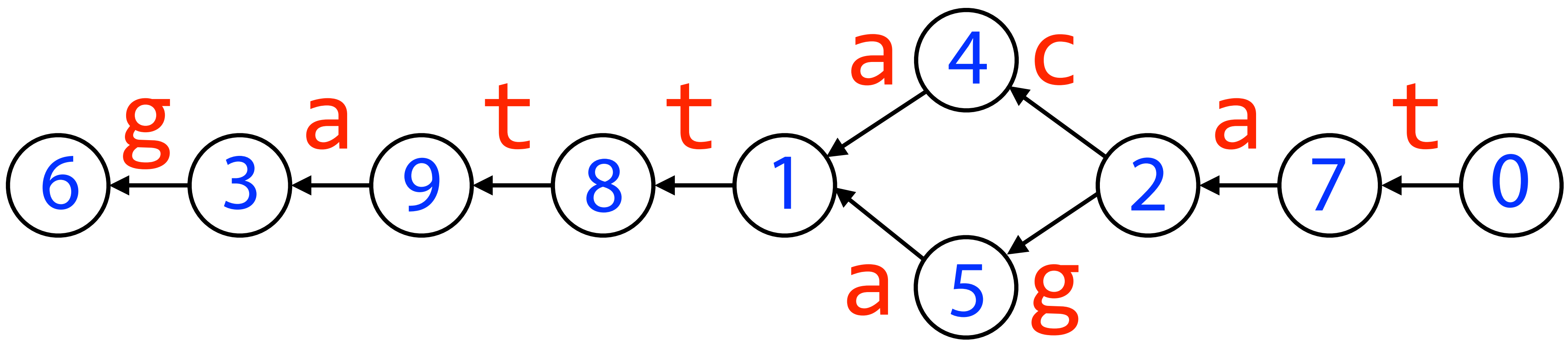
Nodes: 0 1 2 3 4 5 67 8 9

Wheeler graphs



$I =$

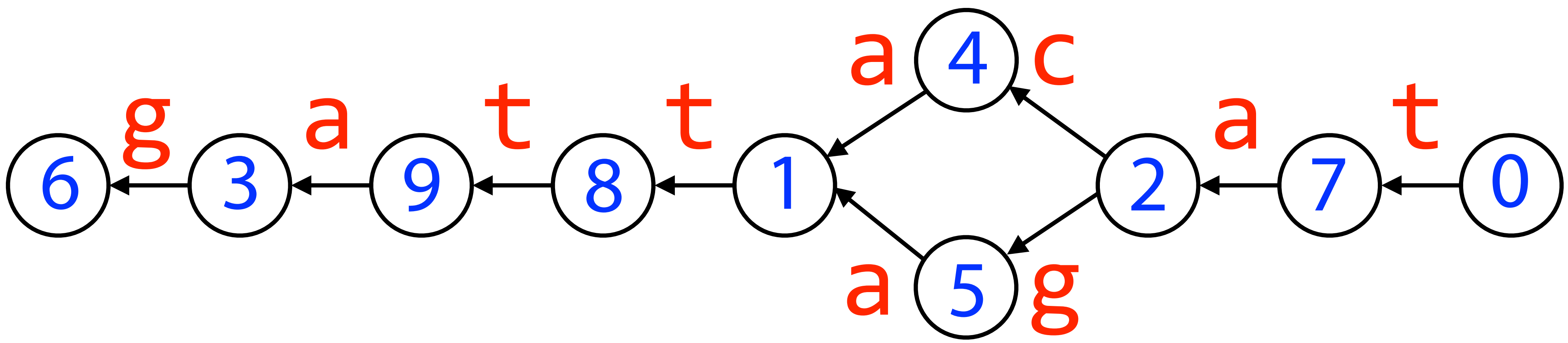
Wheeler graphs



$$I = 10010101010101010101$$

Nodes: 01 2 3 4 5 6 7 8 9

Wheeler graphs

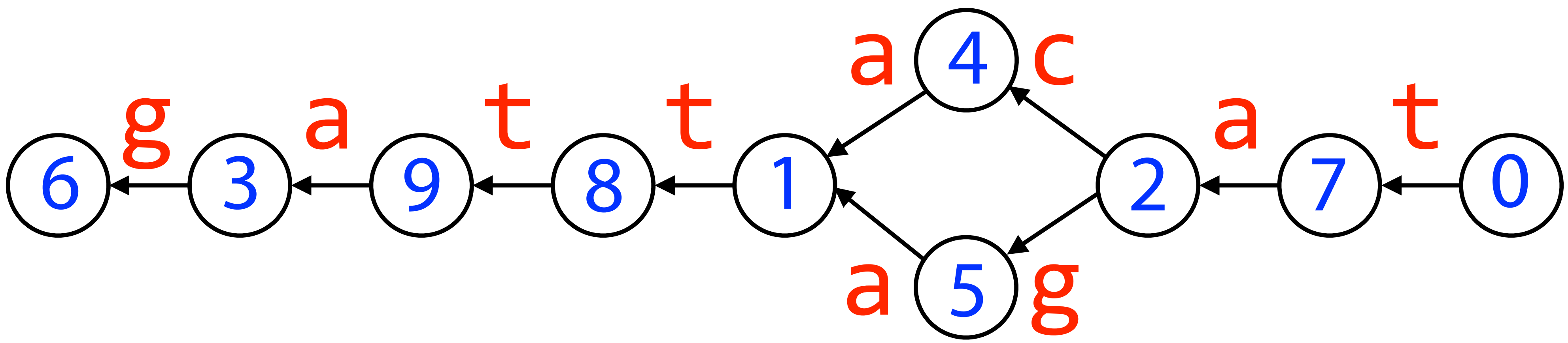


Idea 3: Encode edge labels corresponding to 0s in O

$O =$ 01010010101011010101

$L =$

Wheeler graphs

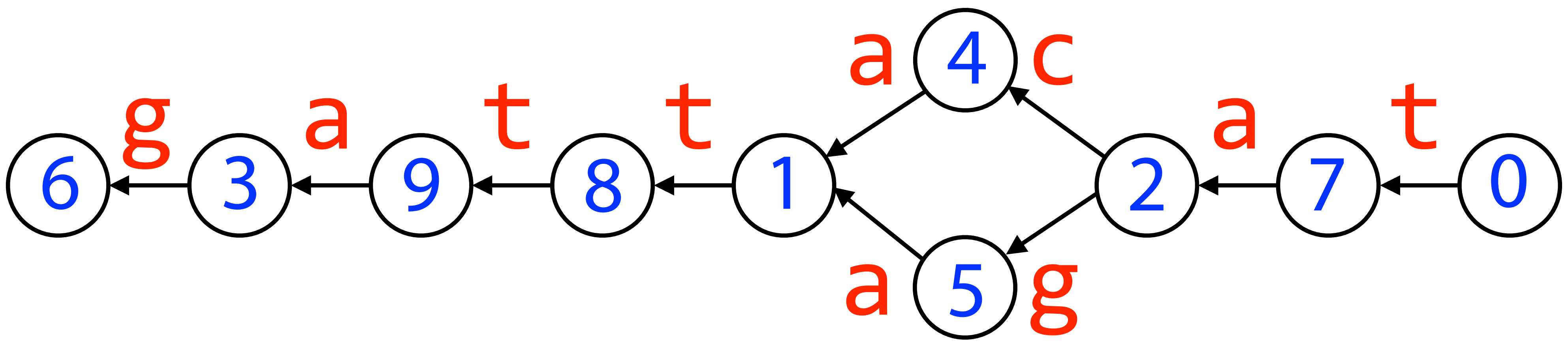


Idea 3: Encode edge labels corresponding to 0s in O

$O =$ 01010010101011010101

$L =$ t t cg g a a a t a

Wheeler graphs



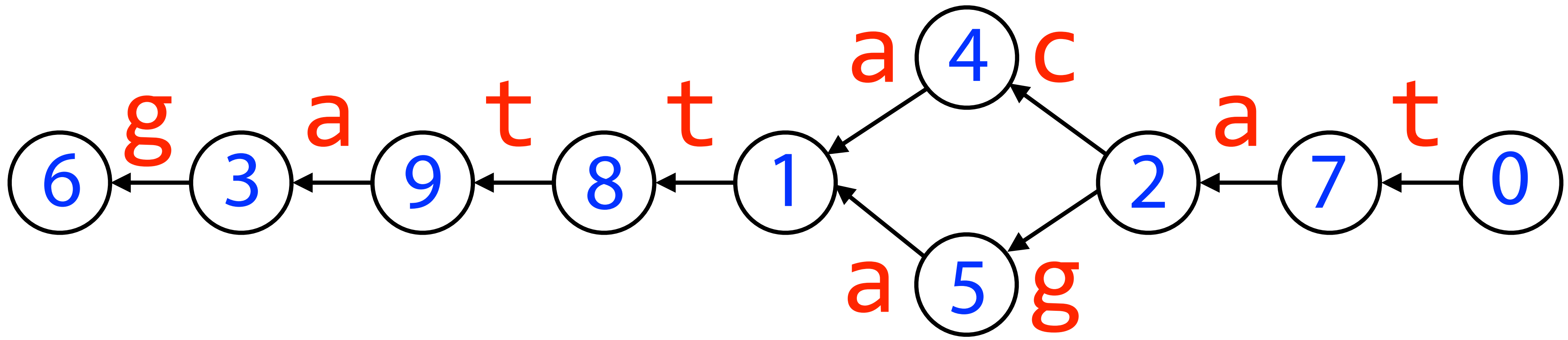
Idea 3: Encode edge labels corresponding to 0s in O

$O =$ 0101001010101101010 1

t t cg g a a a t a

$L =$ ttcggaaata

Wheeler graphs



$$I = \texttt{10010101010101010101}$$

$$O = \texttt{01010010101010110101}$$

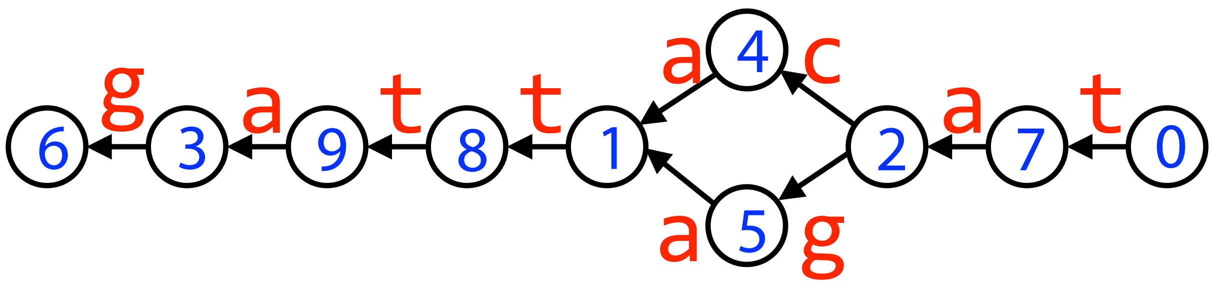
$$L = \texttt{ttcggaaata}$$

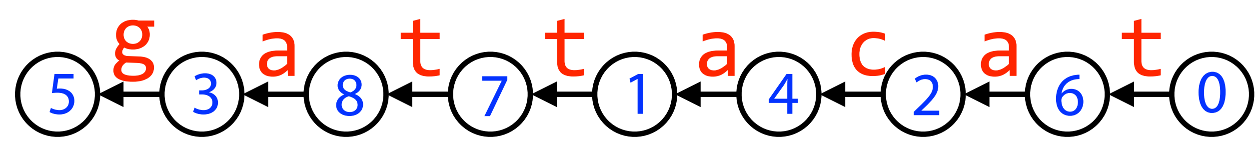
How long is I? (# edges) + (# nodes) bits

How long is O? (# edges) + (# nodes) bits

How long is L? (# edges) chars

Wheeler graphs



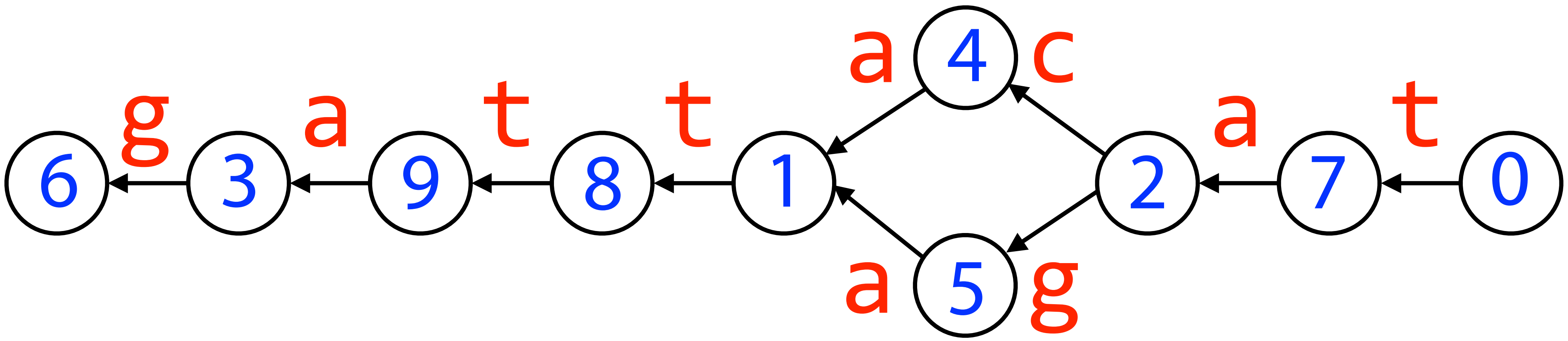


`I: 10010101010101010101`	`I: 10101010101010101`
`O: 01010010101011010101`	`O: 01010101011010101`
`L: ttcggaaata`	`L: ttcgaata`

$BWT(T) =$ `ttcga$ata`

L is like BWT; *I* & *O* are specifically for graph structure

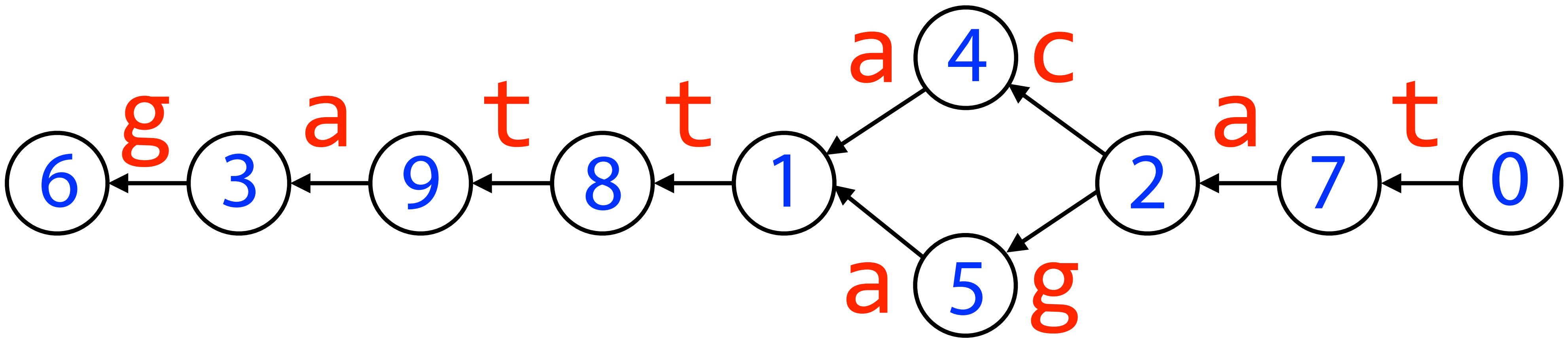
Wheeler graphs



How to find indegree of node $i = 3$?

$$I = \texttt{10010101010101010101}$$

Wheeler graphs



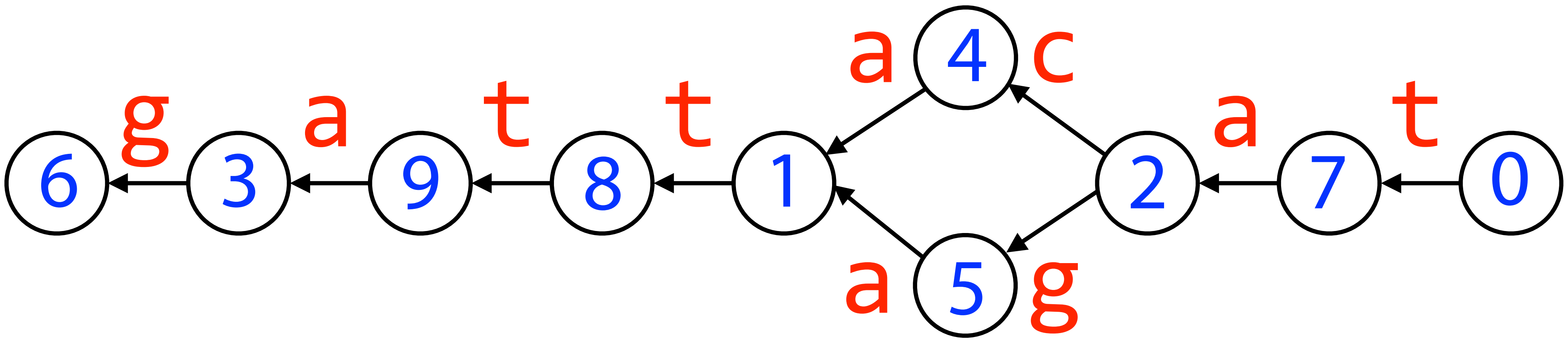
How to find indegree of node $i = 3$?

$$I = 1001010\underline{0}1010101010101$$

$I.\text{select}_1(3)$ - $I.\text{select}_1(2)$ - 1

Similar for outdegree

Wheeler graphs

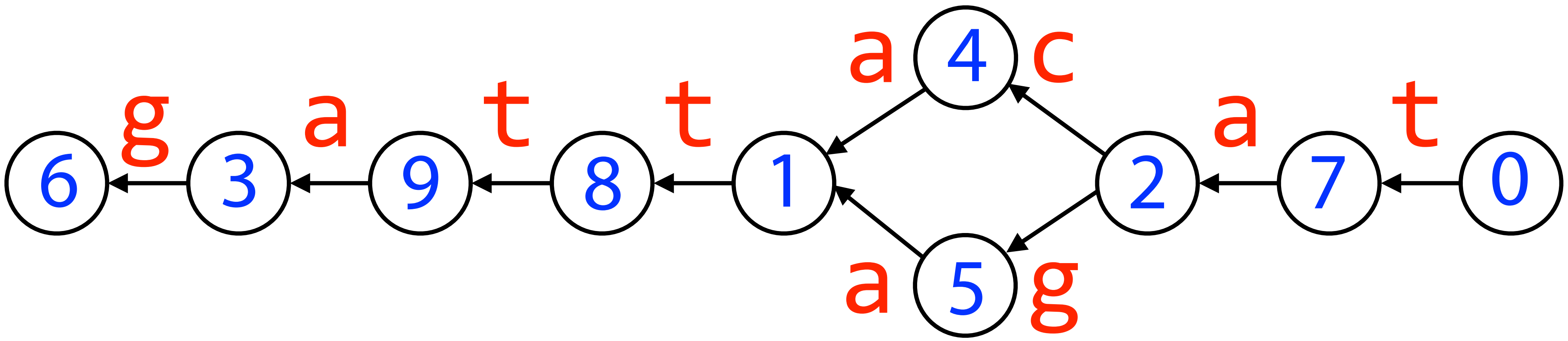


How to get labels of edges outgoing from node $i = 2$?

$L =$ ttcggaaata

$O =$ 0101001010101011010101

Wheeler graphs

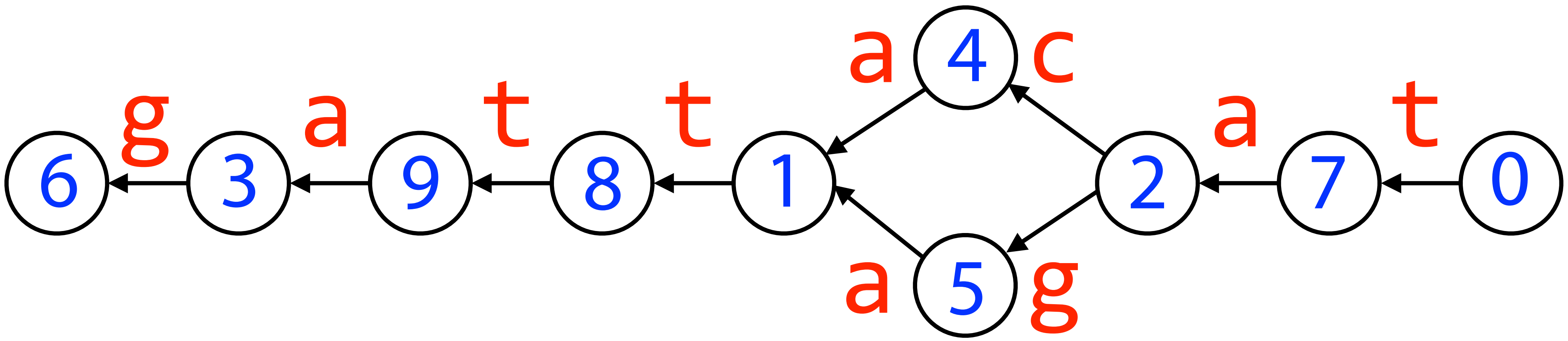


How to get labels of edges outgoing from node $i = 2$?

$L =$ ttcggaaata

$O =$ 01010010101011010101

Wheeler graphs



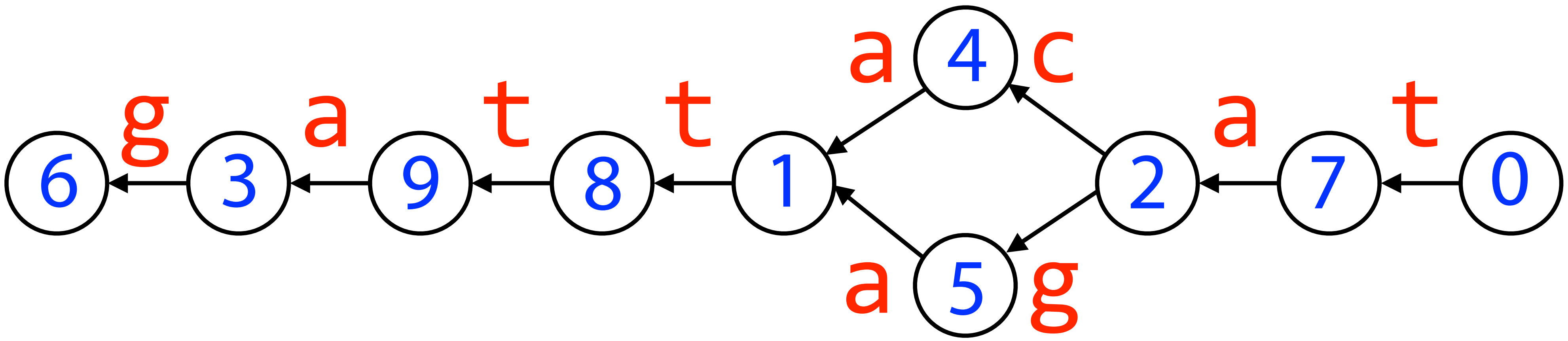
How to get labels of edges outgoing from node $i = 2$?

$L =$ ttcggaaata

$O =$ 0101001010101101010101

$O\,.\,\text{select}_1(1) = 3$ $O\,.\,\text{select}_1(2) = 6$

Wheeler graphs



How to get labels of edges outgoing from node $i = 2$?

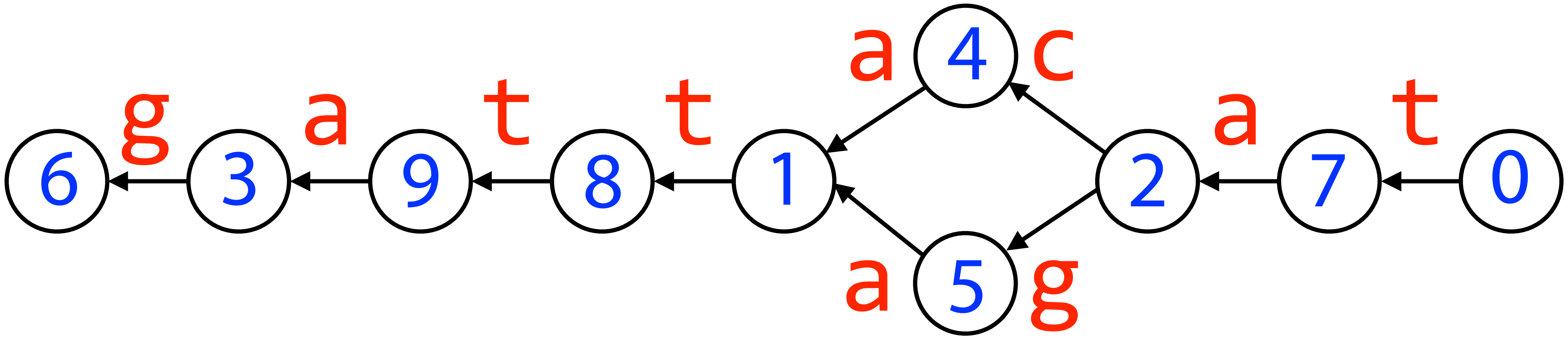
$L =$ ttcggaaata

$O =$ 0101001010101101010101

$O\,.\,\text{select}_1(1) = 3$ $\quad O\,.\,\text{select}_1(2) = 6$

Extract $L[\text{rank}_0(3) = 2 : \text{rank}_0(6) = 4]$ = cg

Wheeler graphs



How do we use these bitvectors for ***matching***?

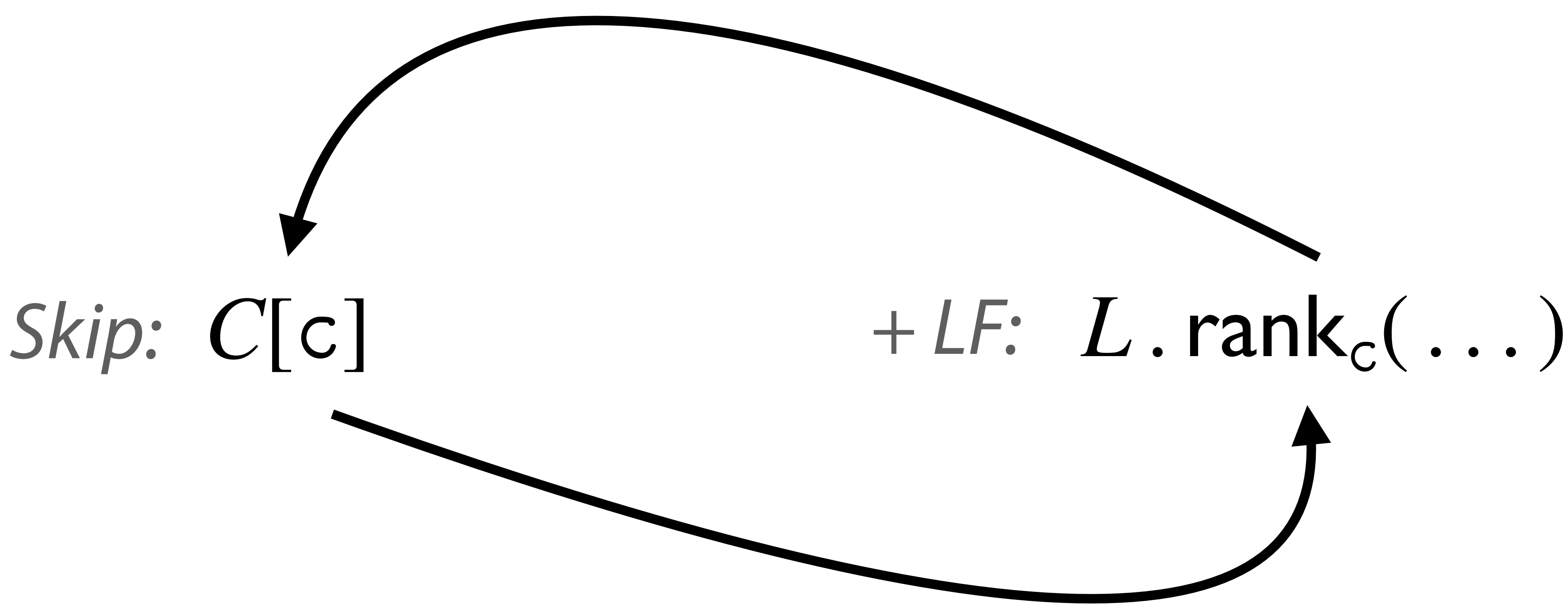
🤔🤔🤔

Wheeler graphs

P = aba

F						*L*
$	a	b	a	a	b	**a_0**
a_0	$	a	b	a	a	**b_0**
a_1	a	b	a	$	a	**b_1**
a_2	b	a	$	a	b	**a_1**
a_3	b	a	a	b	a	**$**
b_0	a	$	a	b	a	**a_2**
b_1	a	a	b	a	$	**a_3**

FM Index match query loop:

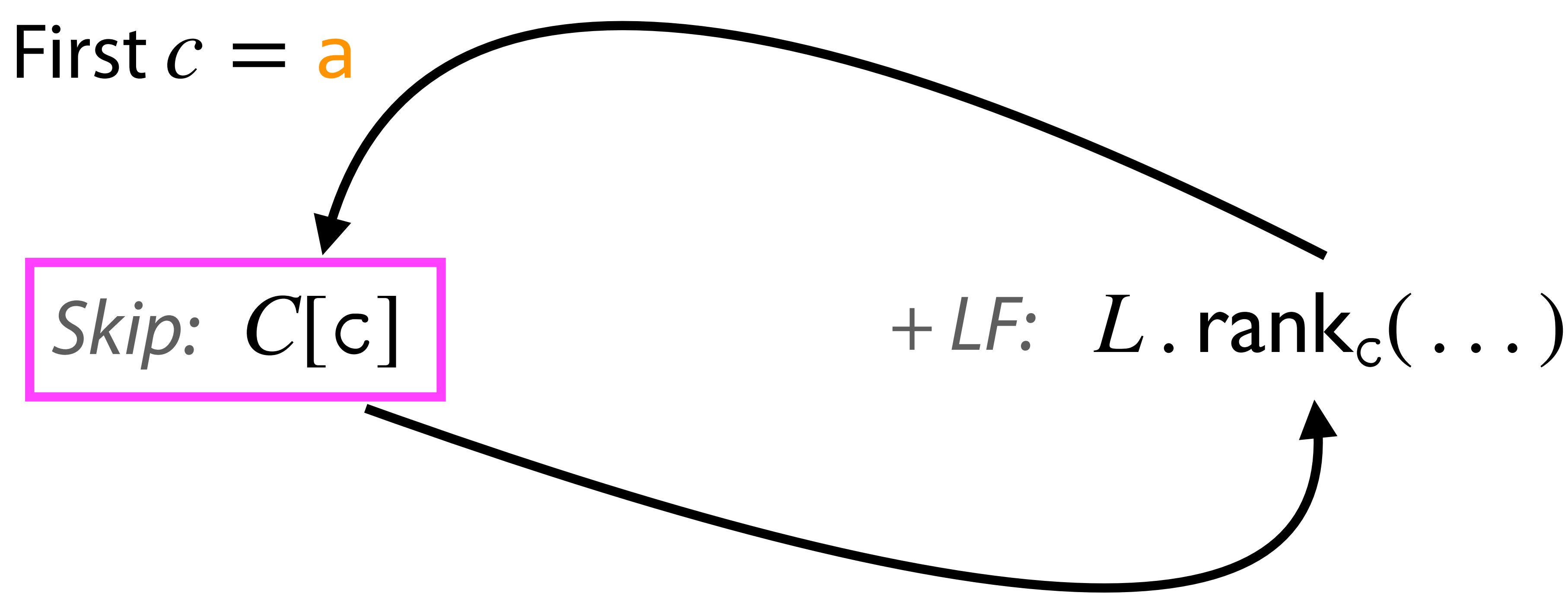


Wheeler graphs

P = aba

FM Index match query loop:

F						*L*
$	a	b	a	a	b	a_0
a_0	$	a	b	a	a	b_0
a_1	a	b	a	$	a	b_1
a_2	b	a	$	a	b	a_1
a_3	b	a	a	b	a	$
b_0	a	$	a	b	a	a_2
b_1	a	a	b	a	$	a_3

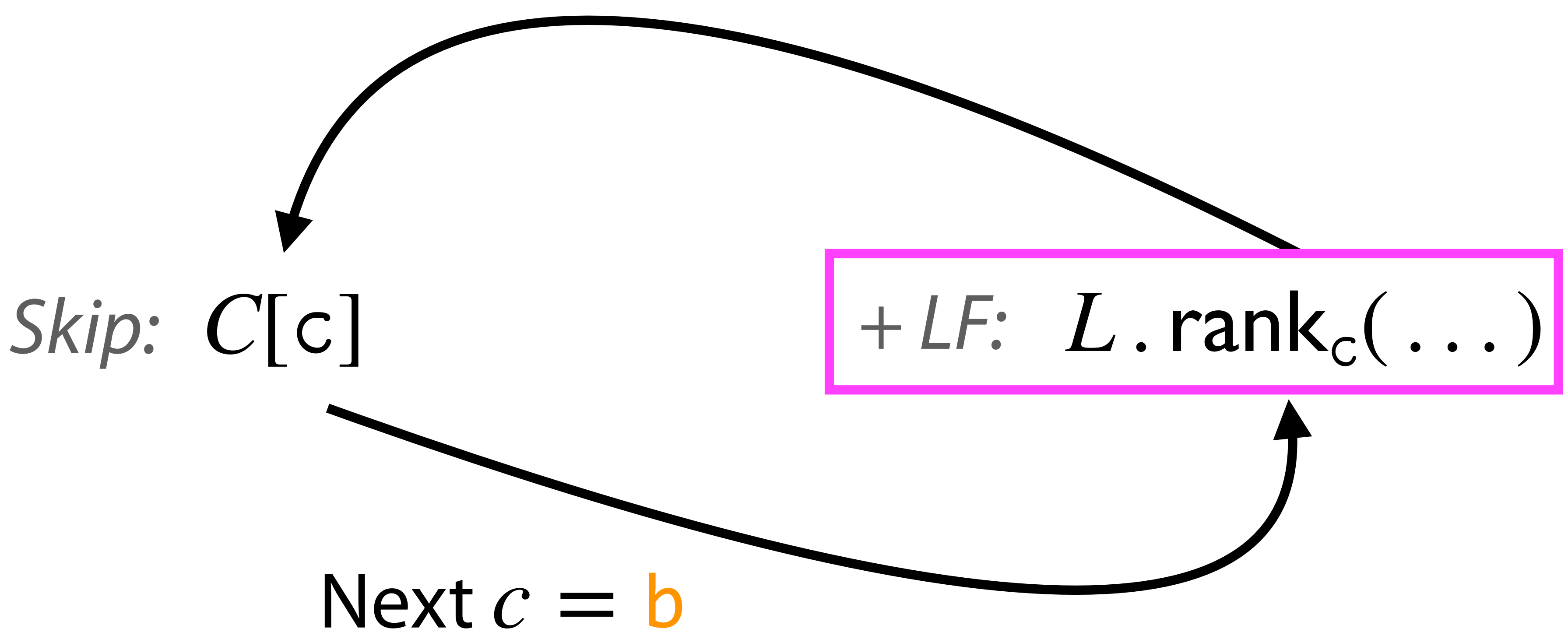


Wheeler graphs

P = aba

FM Index match query loop:

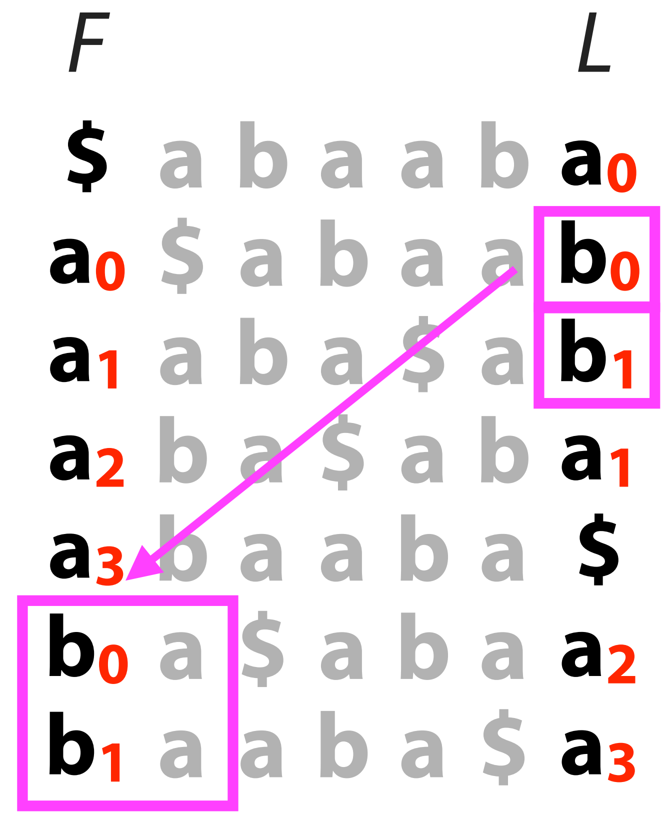
F						L
$	a	b	a	a	b	a_0
a_0	$	a	b	a	a	b_0
a_1	a	b	a	$	a	b_1
a_2	b	a	$	a	b	a_1
a_3	b	a	a	b	a	$
b_0	a	$	a	b	a	a_2
b_1	a	a	b	a	$	a_3

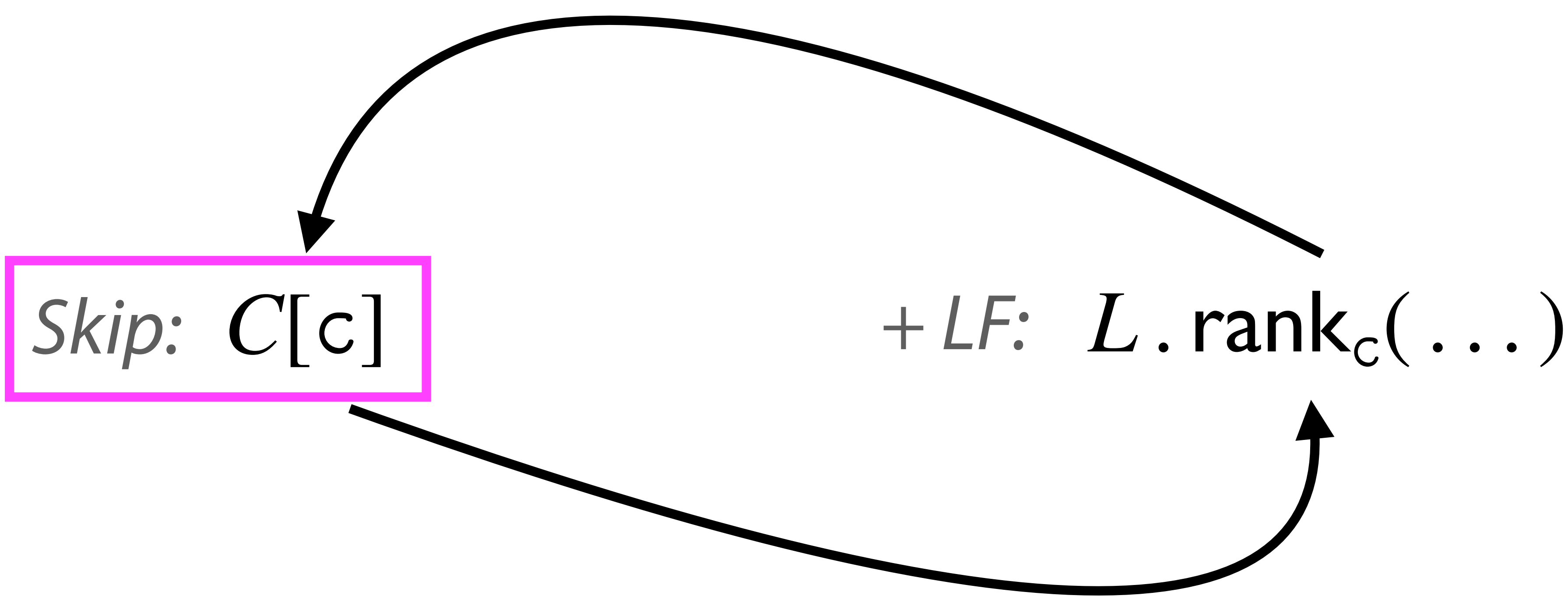


Wheeler graphs

P = aba

FM Index match query loop:



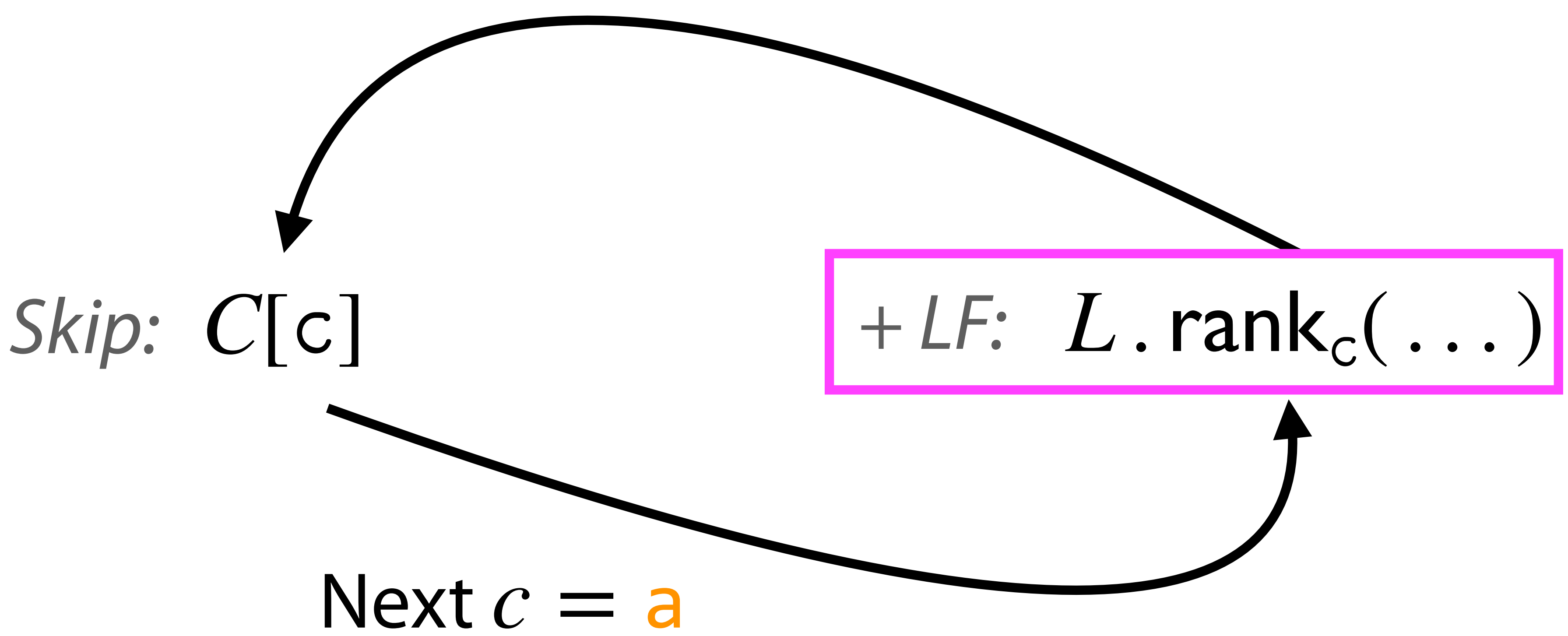


Wheeler graphs

P = aba

F						L
$	a	b	a	a	b	a_0
a_0	$	a	b	a	a	b_0
a_1	a	b	a	$	a	b_1
a_2	b	a	$	a	b	a_1
a_3	b	a	a	b	a	$
b_0	a	$	a	b	a	a_2
b_1	a	a	b	a	$	a_3

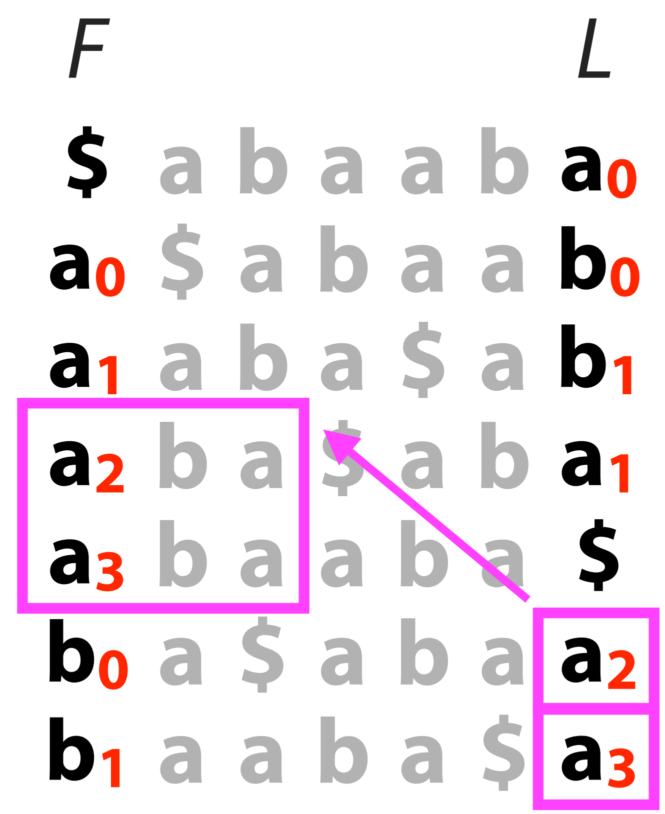
FM Index match query loop:

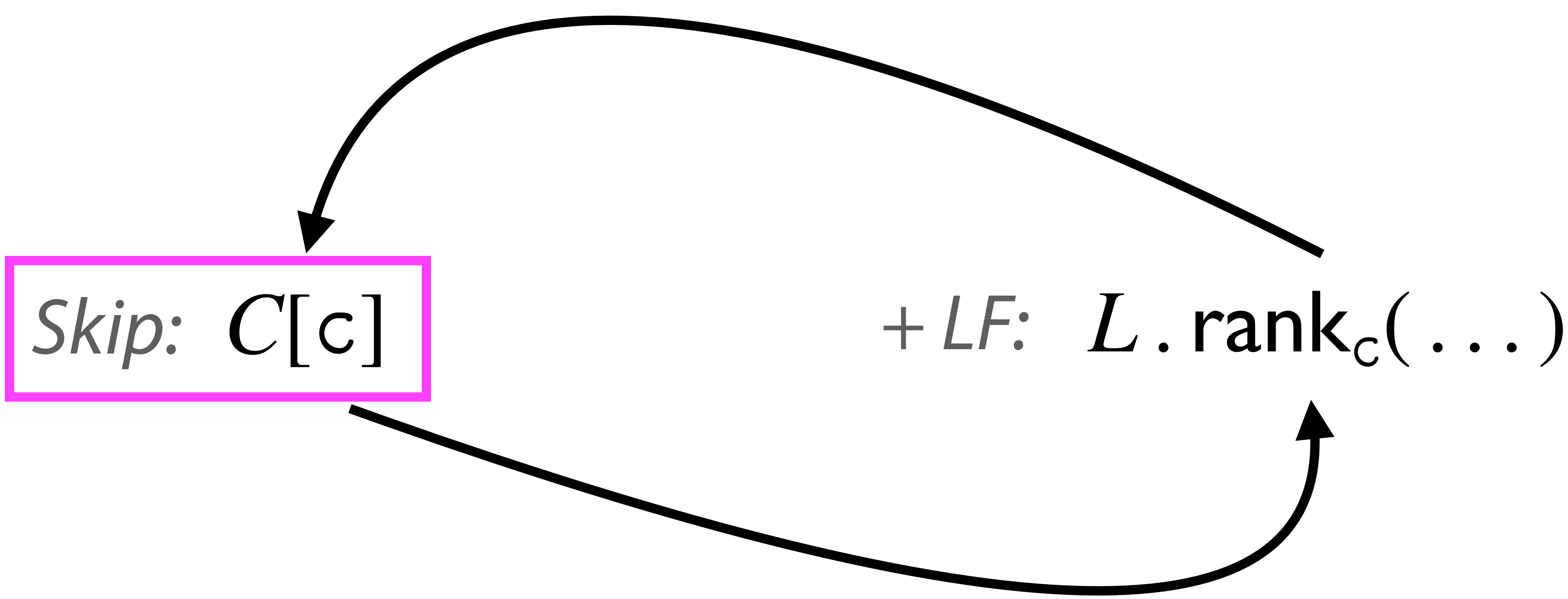


Wheeler graphs

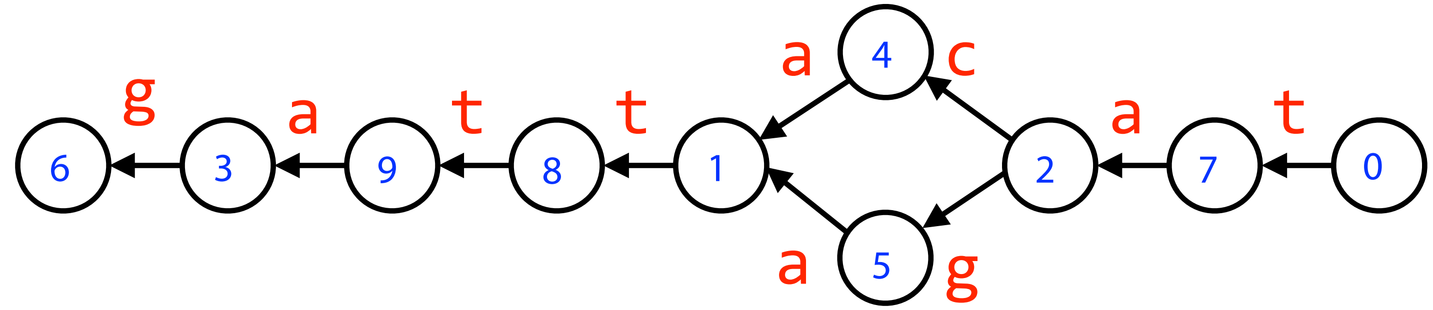
P = aba

FM Index match query loop:





Wheeler graphs



Wheeler graph match query loop:

I : 10010101010101010101

O : 01010010101011010101

L : ttcggaaata

Find range of characters in L:

$L\,.\,\mathrm{rank}_c(\,\ldots\,)$

Skip: $C[\mathrm{c}]$

Follow incoming edges in I:

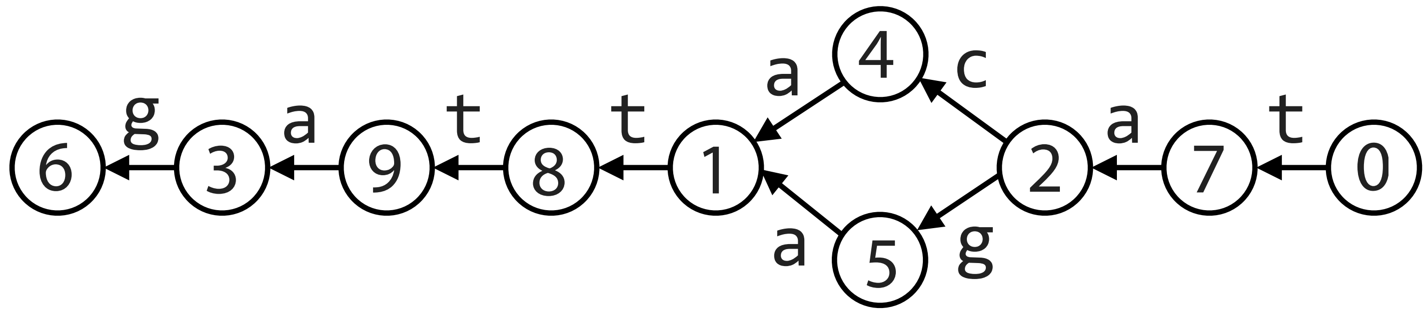
$I\,.\,\mathrm{rank}_1(I\,.\,\mathrm{select}_0(\,\ldots\,))$

Find outgoing edges in O:

$O\,.\,\mathrm{rank}_0(O\,.\,\mathrm{select}_1(\,\ldots\,))$

Wheeler graphs

P = aga




```
F: aaaacggttt
I: 10010101010101010101
O: 01010010101011010101
L: ttcggaaata
```

$L\,.\,\mathrm{rank}_c(\,\ldots\,)$

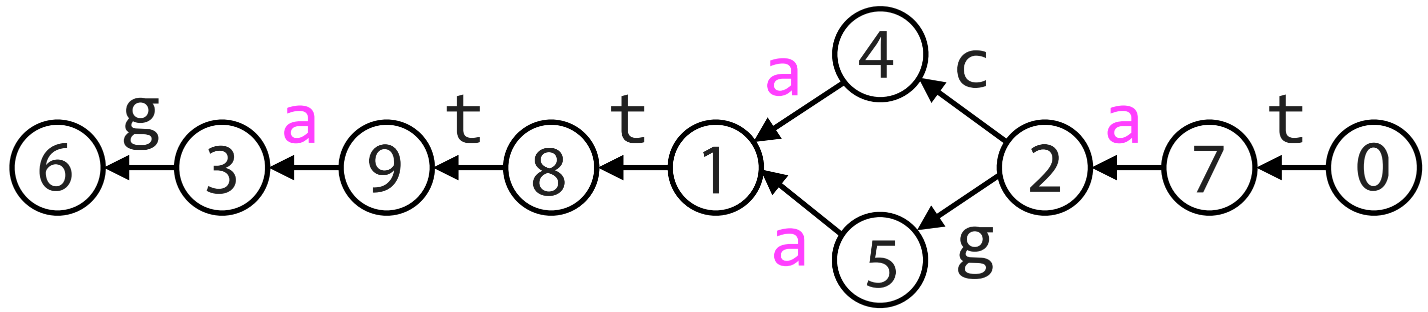
$C[\mathtt{c}]$

$O\,.\,\mathrm{rank}_0(O\,.\,\mathrm{select}_1(\,\ldots\,))$

$I\,.\,\mathrm{rank}_1(I\,.\,\mathrm{select}_0(\,\ldots\,))$

Wheeler graphs

P = aga




```
F: aaaacggttt

I: 1001010101010101010101

O: 0101001010101011010101

L: ttcggaaata
```

First $c =$ a

$L\,.\,\mathrm{rank}_c(\ldots)$

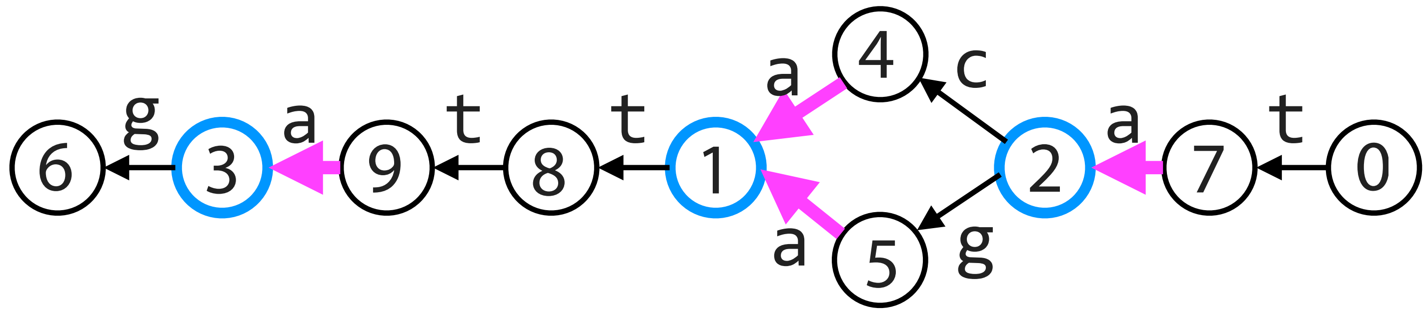
$C[\mathrm{c}]$

$I\,.\,\mathrm{rank}_1(I\,.\,\mathrm{select}_0(\ldots))$

$O\,.\,\mathrm{rank}_0(O\,.\,\mathrm{select}_1(\ldots))$

Wheeler graphs

P = aga

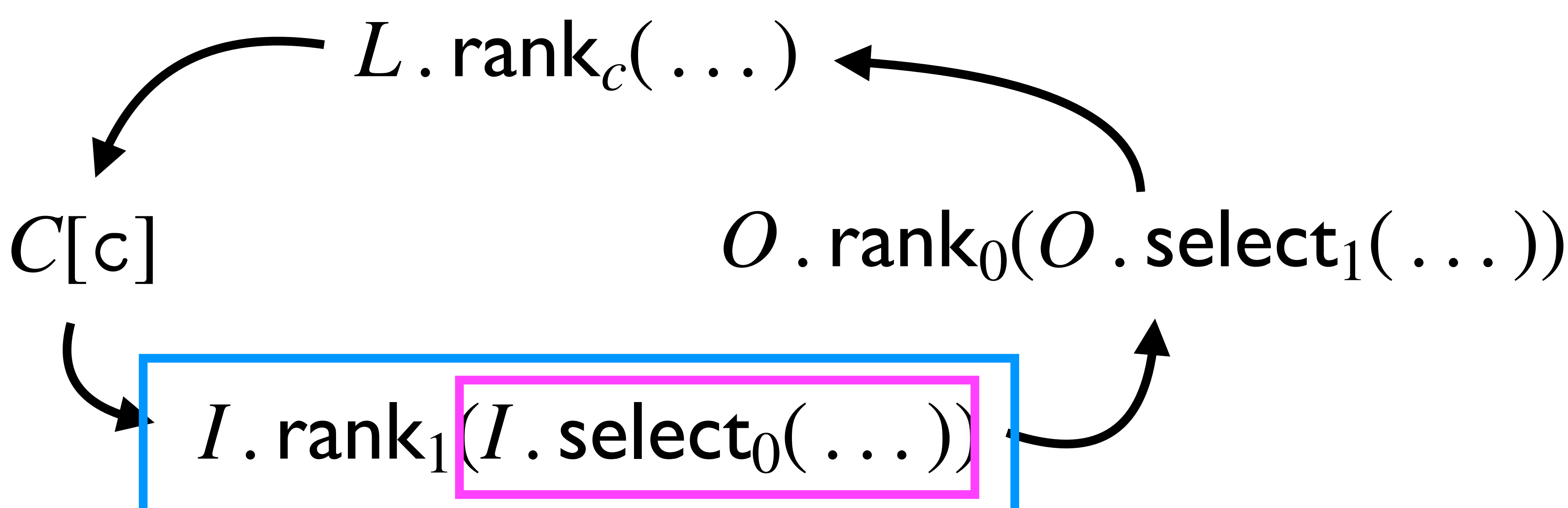


F: aaaacggttt

I: 1001010101010101010101

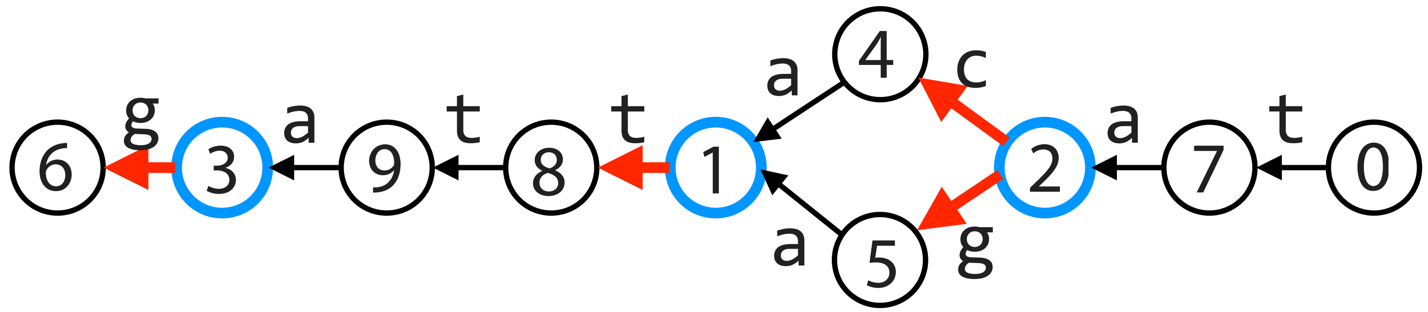
O: 0101001010101101010101

L: ttcggaaata

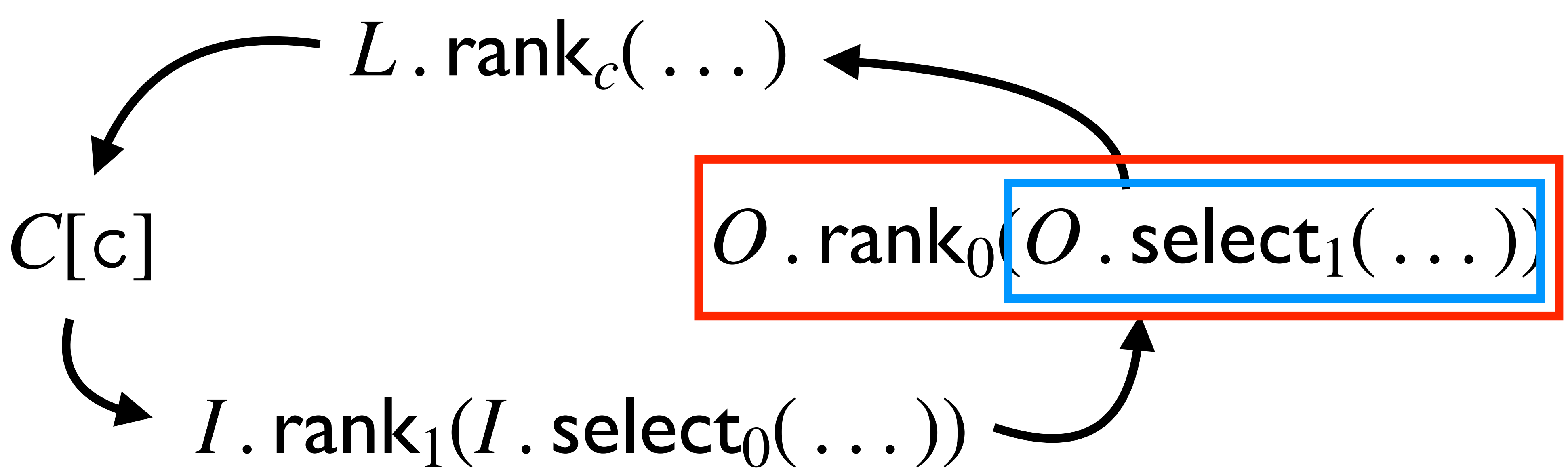


Wheeler graphs

P = aga

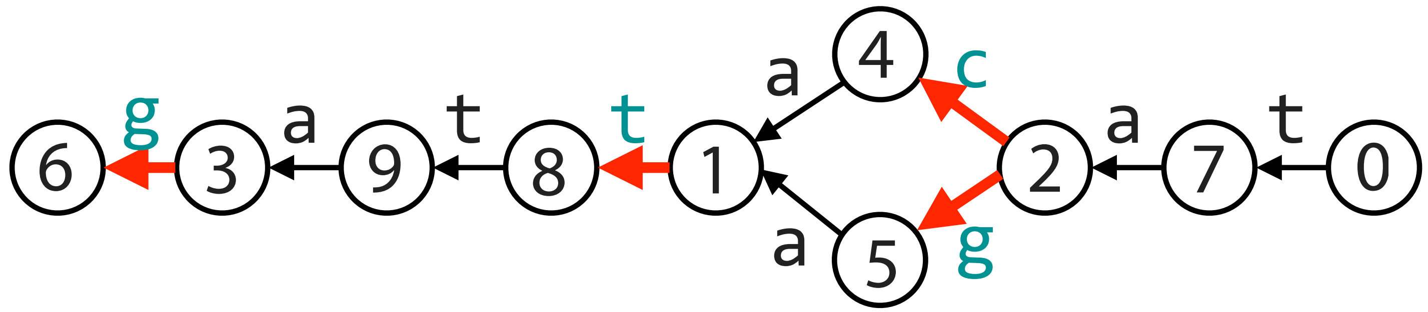



```
F: aaaacggttt

I: 1001010101010101010101

O: 0101001010101011010101

L: ttcggaaata
```

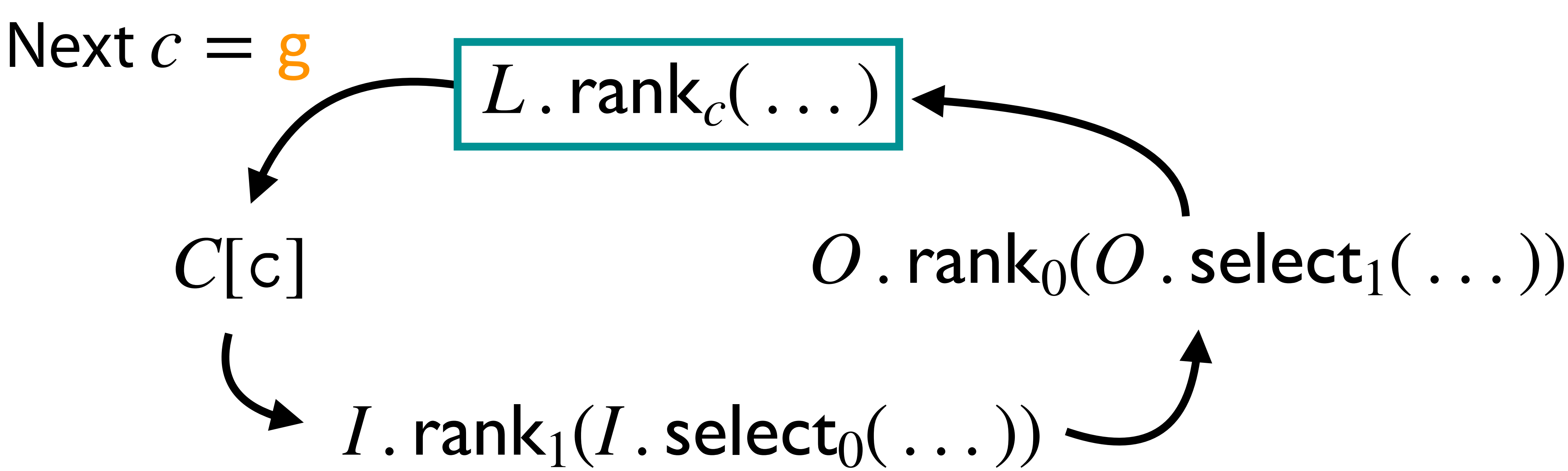


Wheeler graphs

P = aga

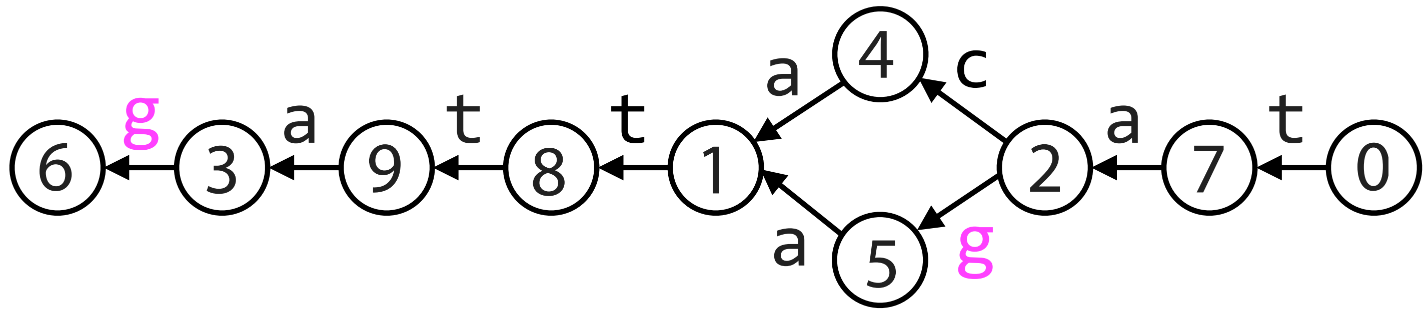



```
F: aaaacggttt

I: 1001010101010101010101

O: 0101001010101011010101

L: ttcggaaata
```



Wheeler graphs

P = aga

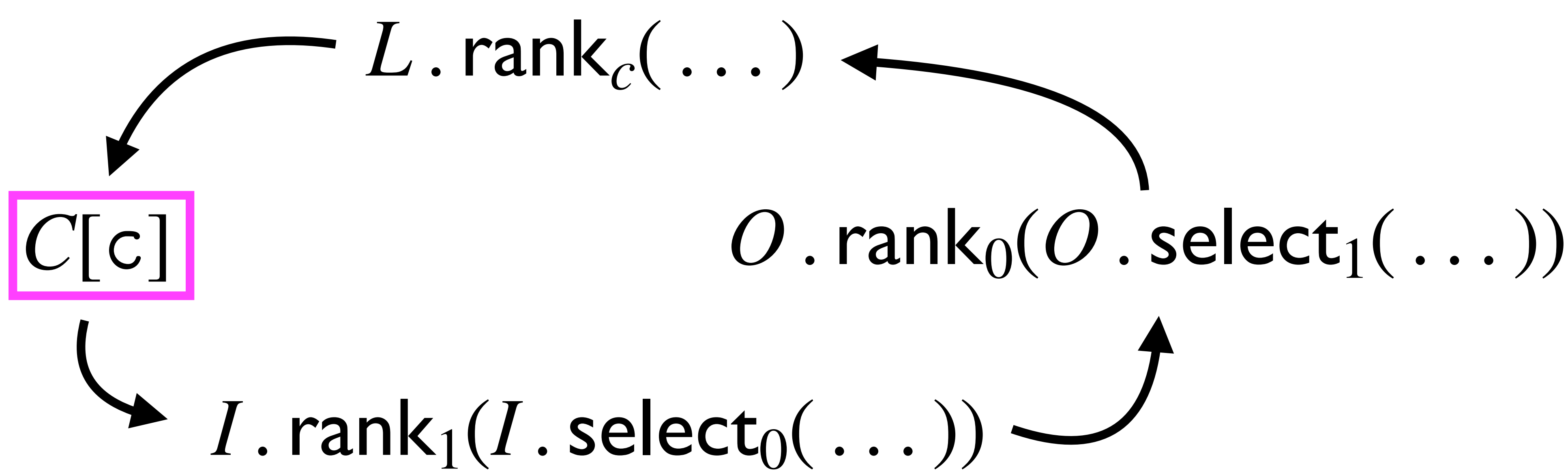


F: aaaacggttt

I: 10010101010101010101

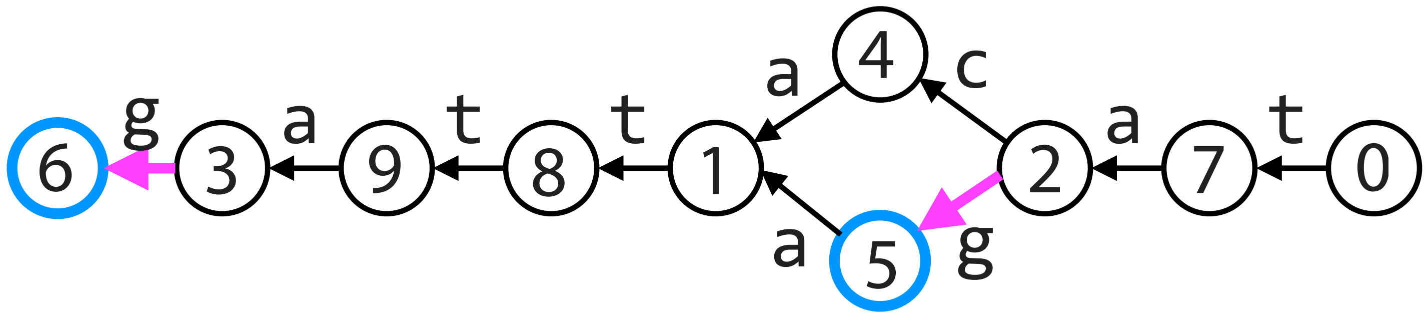
O: 01010010101011010101

L: ttcggaaata

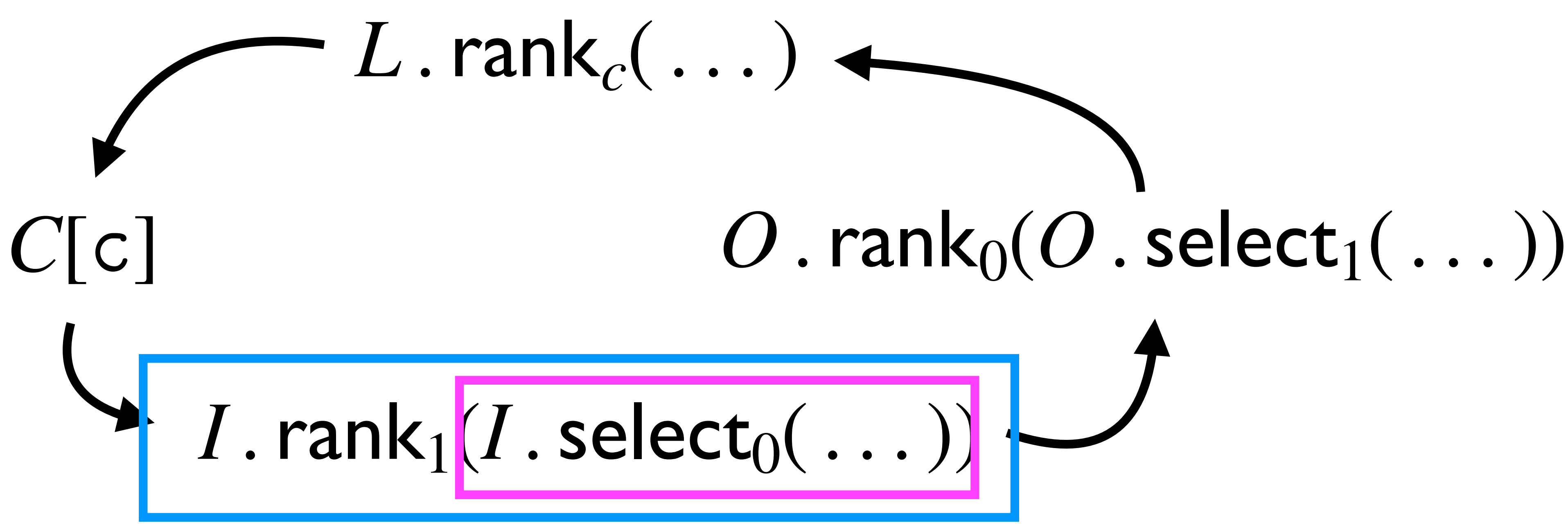


Wheeler graphs

P = aga

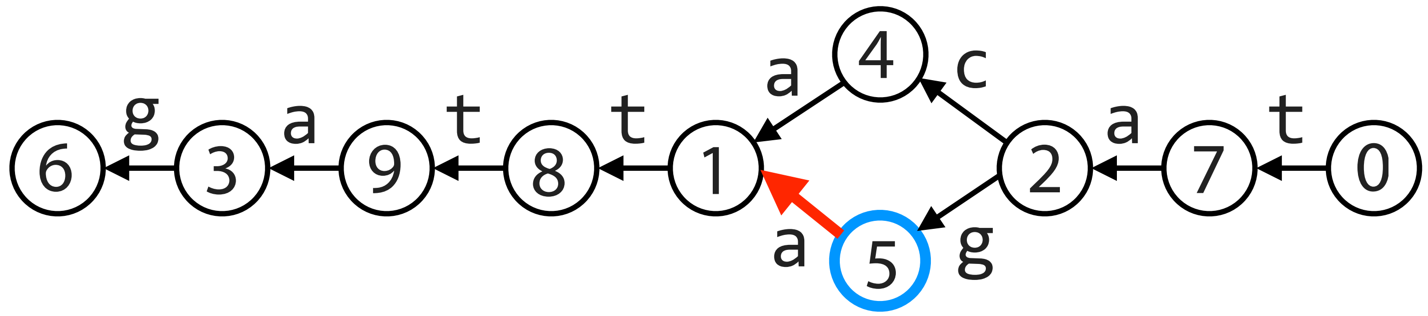



```
F: aaaacggttt

I: 10010101010101010101
O: 01010010101011010101

L: ttcggaaata
```



Wheeler graphs

P = aga




```
F:   aaaacggttt

I: 10010101010101010101

O: 01010010101011010101

L:   ttcggaaata
```

$L\,.\,\text{rank}_c(\,\ldots\,)$

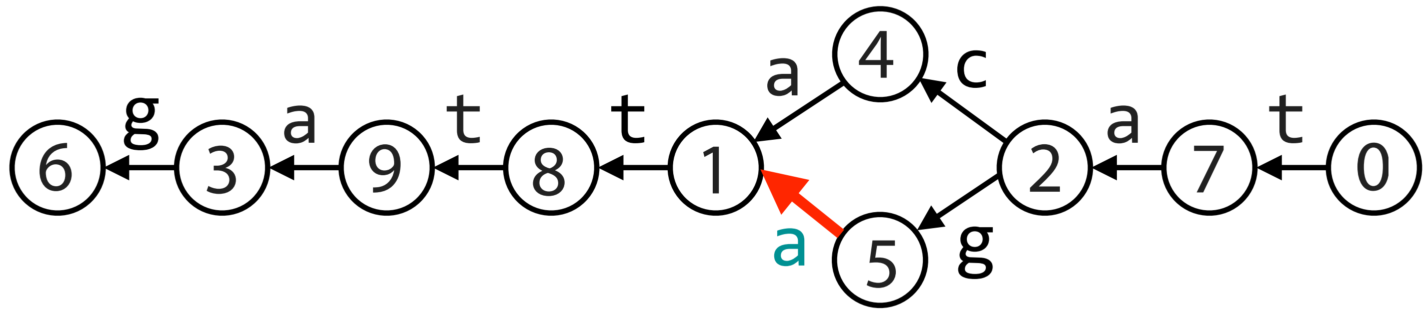
$C[\text{c}]$

$O\,.\,\text{rank}_0(O\,.\,\text{select}_1(\,\ldots\,))$

$I\,.\,\text{rank}_1(I\,.\,\text{select}_0(\,\ldots\,))$

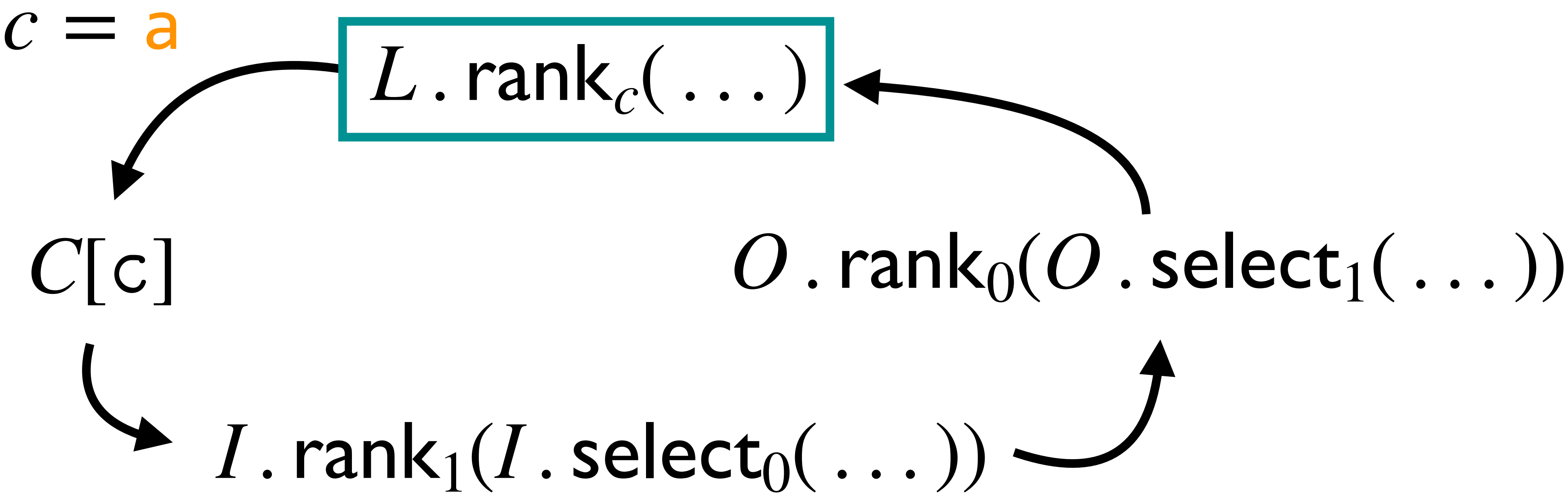
Wheeler graphs

P = aga



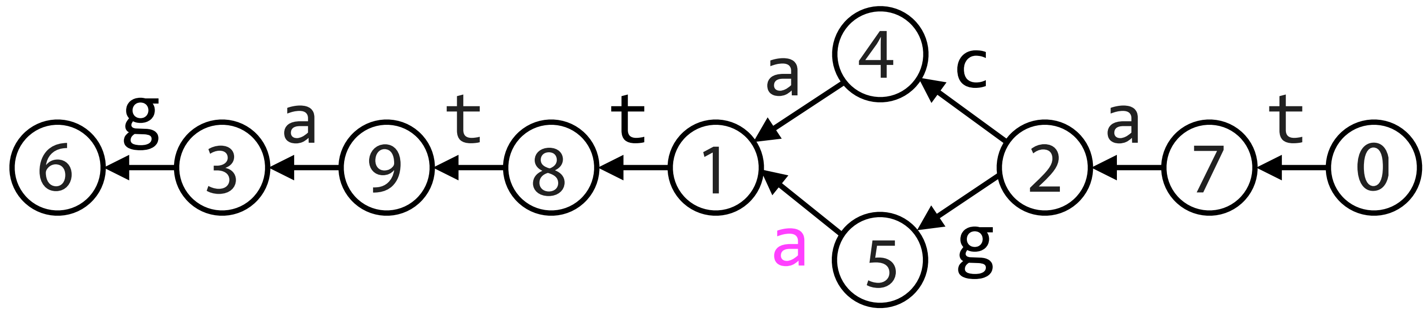

```
F: aaaacggttt
I: 10010101010101010101
O: 01010010101011010101
L: ttcggaaata
```

Next c = a



Wheeler graphs

P = aga




```
   F: aaaacggttt
I: 10010101010101010101
O: 01010010101011010101
   L: ttcggaaata
```

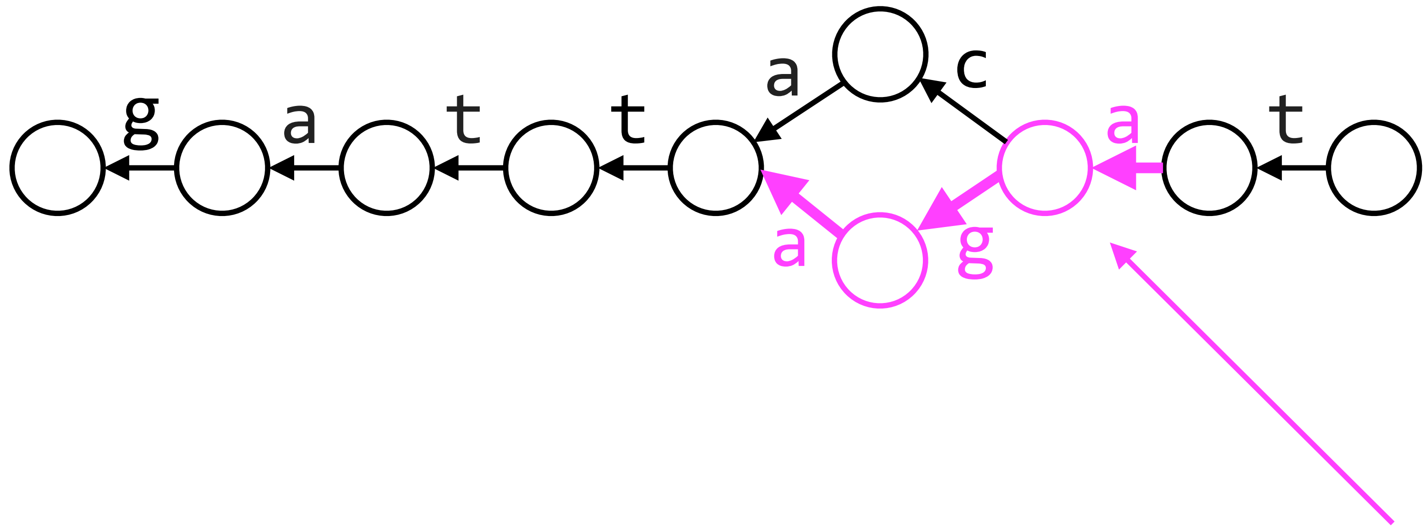
$L\,.\,\mathrm{rank}_c(\,\ldots\,)$

$C[\mathrm{c}]$ $\qquad$ $O\,.\,\mathrm{rank}_0(O\,.\,\mathrm{select}_1(\,\ldots\,))$

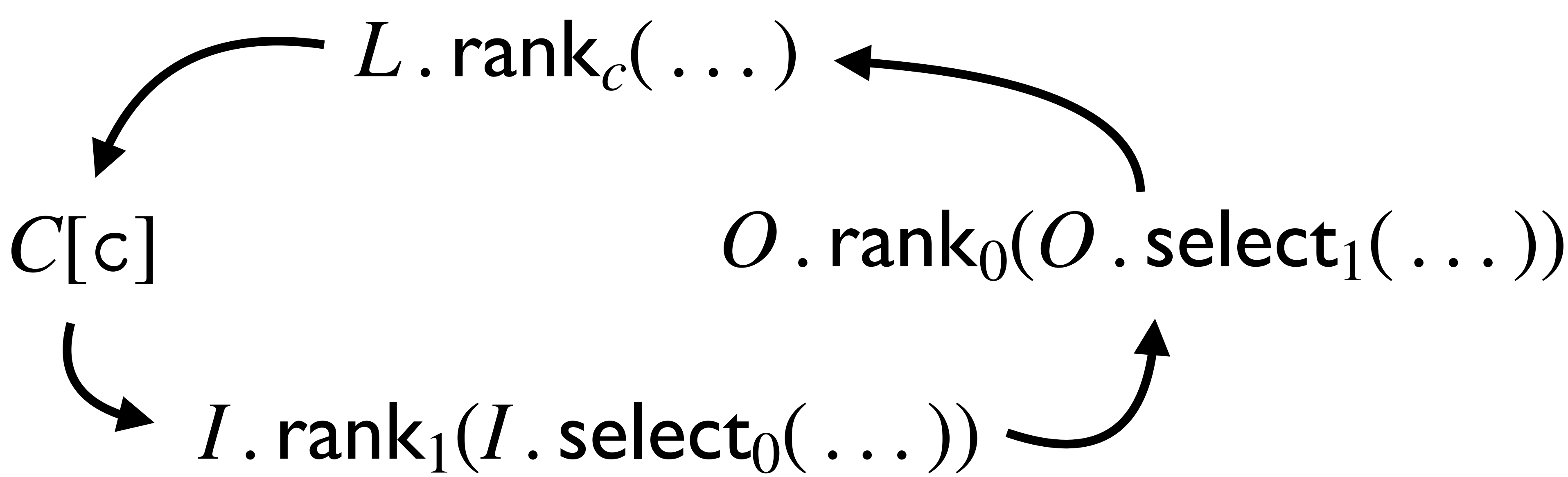
$I\,.\,\mathrm{rank}_1(I\,.\,\mathrm{select}_0(\,\ldots\,))$

Wheeler graphs

P = aga



Final answer: 1 match, corresponding to this path



Wheeler graph

Given character c & next character c', one step of matching process:

FM index:

$$r_{top} \leftarrow C[c]$$
$$r_{bot} \leftarrow C[c{+}1]$$
$$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{top}, c')$$
$$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{bot}, c')$$

Wheeler graph:

$$r_{top} \leftarrow C[c]$$
$$i_{top} \leftarrow I\,.\,\text{select}_0(r_{top})$$
$$j_{top} \leftarrow I\,.\,\text{rank}_1(i_{top})$$
$$k_{top} \leftarrow O\,.\,\text{select}_1(j_{top} - 1)$$
$$\ell_{top} \leftarrow O\,.\,\text{rank}_0(k_{top})$$
$$m_{top} \leftarrow S\,.\,\text{rank}_c(\ell_{top}, c')$$

$$r_{bot} \leftarrow C[c{+}1]$$
$$i_{bot} \leftarrow I\,.\,\text{select}_0(r_{bot} - 1)$$
$$j_{bot} \leftarrow I\,.\,\text{rank}_1(i_{bot}) + 1$$
$$k_{bot} \leftarrow O\,.\,\text{select}_1(j_{bot} - 1)$$
$$\ell_{bot} \leftarrow O\,.\,\text{rank}_0(k_{bot})$$
$$m_{bot} \leftarrow S\,.\,\text{rank}_c(\ell_{bot}, c')$$

Wheeler graph

What takes **space**?

FM index:

$r_{top} \leftarrow C[c]$

$r_{bot} \leftarrow C[c{+}1]$

$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{top}, c')$

$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{bot}, c')$

Wheeler graph:

$r_{top} \leftarrow C[c]$

$i_{top} \leftarrow I\,.\,\text{select}_0(r_{top})$

$j_{top} \leftarrow I\,.\,\text{rank}_1(i_{top})$

$k_{top} \leftarrow O\,.\,\text{select}_1(j_{top} - 1)$

$\ell_{top} \leftarrow O\,.\,\text{rank}_0(k_{top})$

$m_{top} \leftarrow L\,.\,\text{rank}_c(\ell_{top}, c')$

$r_{bot} \leftarrow C[c{+}1]$

$i_{bot} \leftarrow I\,.\,\text{select}_0(r_{bot} - 1)$

$j_{bot} \leftarrow I\,.\,\text{rank}_1(i_{bot}) + 1$

$k_{bot} \leftarrow O\,.\,\text{select}_1(j_{bot} - 1)$

$\ell_{bot} \leftarrow O\,.\,\text{rank}_0(k_{bot})$

$m_{bot} \leftarrow L\,.\,\text{rank}_c(\ell_{bot}, c')$

C array: $\sigma \log n$

$WT(BWT)$ rank: $n \log \sigma + \check{o}(n \log \sigma)$

(units are bits)

Wheeler graph

What takes **space**?

FM index:

$r_{top} \leftarrow C[c]$

$r_{bot} \leftarrow C[c+1]$

$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{top}, c')$

$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{bot}, c')$

Wheeler graph:

$r_{top} \leftarrow C[c]$ $\quad$ $r_{bot} \leftarrow C[c+1]$

$i_{top} \leftarrow I\,.\,\text{select}_0(r_{top})$ $\quad$ $i_{bot} \leftarrow I\,.\,\text{select}_0(r_{bot} - 1)$

$j_{top} \leftarrow I\,.\,\text{rank}_1(i_{top})$ $\quad$ $j_{bot} \leftarrow I\,.\,\text{rank}_1(i_{bot}) + 1$

$k_{top} \leftarrow O\,.\,\text{select}_1(j_{top} - 1)$ $\quad$ $k_{bot} \leftarrow O\,.\,\text{select}_1(j_{bot} - 1)$

$\ell_{top} \leftarrow O\,.\,\text{rank}_0(k_{top})$ $\quad$ $\ell_{bot} \leftarrow O\,.\,\text{rank}_0(k_{bot})$

$m_{top} \leftarrow L\,.\,\text{rank}_c(\ell_{top}, c')$ $\quad$ $m_{bot} \leftarrow L\,.\,\text{rank}_c(\ell_{bot}, c')$

C array: $\sigma \log |E|$

I rank+select: $|E| + |N| + \check{o}(|E| + |N|)$

O rank+select: $|E| + |N| + \check{o}(|E| + |N|)$

$WT(L)$ rank: $|E| \log \sigma + \check{o}(|E| \log \sigma)$

(units are bits)

Wheeler graph

What takes **space**?

FM index:

C array: $\sigma \log n$

$WT(BWT)$ rank: $n \log \sigma + \check{o}(n \log \sigma)$

Wheeler graph:

C array: $\sigma \log |E|$

I rank+select: $|E| + |N| + \check{o}(|E| + |N|)$

O rank+select: $|E| + |N| + \check{o}(|E| + |N|)$

$WT(L)$ rank: $|E| \log \sigma + \check{o}(|E| \log \sigma)$

Wheeler graph

What takes **time**?

FM index:

$r_{top} \leftarrow C[c]$

$r_{bot} \leftarrow C[c+1]$

$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{top}, c')$

$m_{top} \leftarrow BWT(T)\,.\,\text{rank}_c(r_{bot}, c')$

Wheeler graph:

$r_{top} \leftarrow C[c]$	$r_{bot} \leftarrow C[c+1]$
$i_{top} \leftarrow I\,.\,\text{select}_0(r_{top})$	$i_{bot} \leftarrow I\,.\,\text{select}_0(r_{bot} - 1)$
$j_{top} \leftarrow I\,.\,\text{rank}_1(i_{top})$	$j_{bot} \leftarrow I\,.\,\text{rank}_1(i_{bot}) + 1$
$k_{top} \leftarrow O\,.\,\text{select}_1(j_{top} - 1)$	$k_{bot} \leftarrow O\,.\,\text{select}_1(j_{bot} - 1)$
$\ell_{top} \leftarrow O\,.\,\text{rank}_0(k_{top})$	$\ell_{bot} \leftarrow O\,.\,\text{rank}_0(k_{bot})$
$m_{top} \leftarrow L\,.\,\text{rank}_c(\ell_{top}, c')$	$m_{bot} \leftarrow L\,.\,\text{rank}_c(\ell_{bot}, c')$

Ranks on wavelet trees: $O(\log \sigma)$

Ranks and selects on bitvectors: $O(1)$