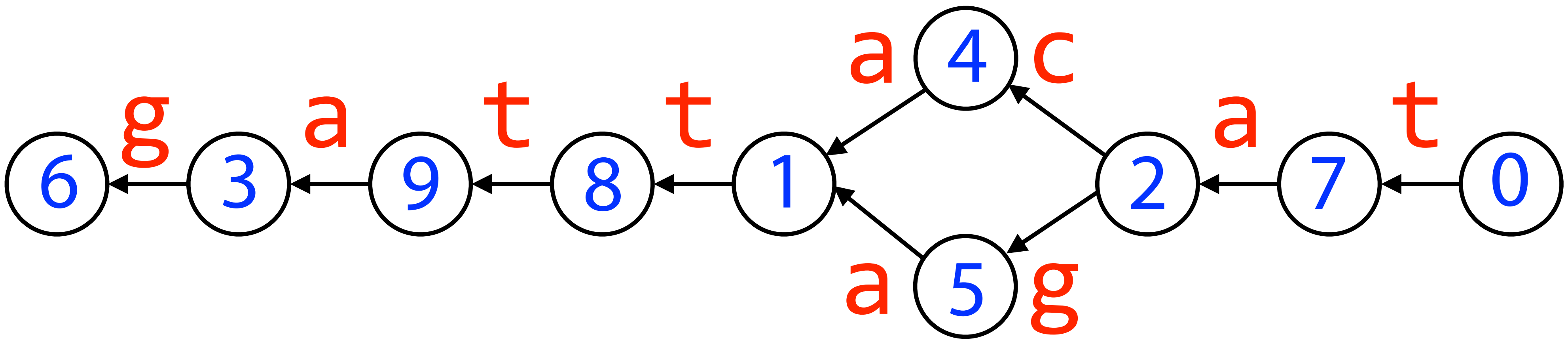
Wheeler graphs, part 2

Ben Langmead

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Department of Computer Science

Wheeler graphs



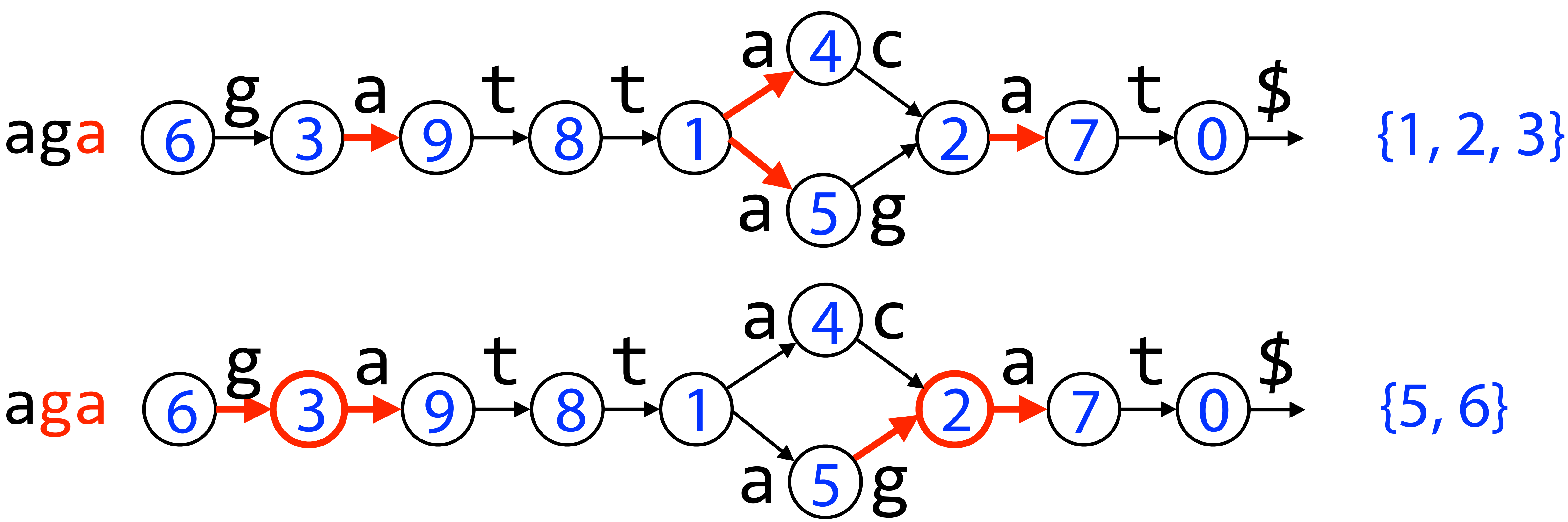
Can we show Wheeler Graphs are “special,” giving us analogs to FM index queries?

We start with “path coherence,” the graph version of ***consecutivity***

Wheeler graphs

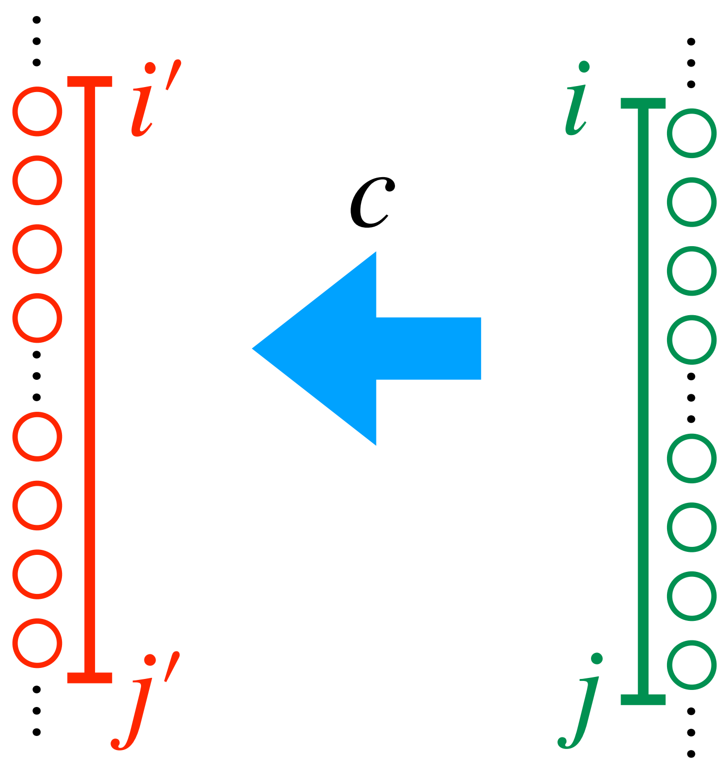
A graph is ***path coherent*** if there is a total order of the nodes such that:

For any consecutive range $[i, j]$ of nodes and string α, the nodes reached by following edges matching α also form a consecutive range



Wheeler graphs

Consider a single step where our initial set of nodes are in consecutive range $[i, j]$ and, after advancing on a single character $c \in \Sigma$, $[i', j']$ is the smallest range containing our next set of nodes

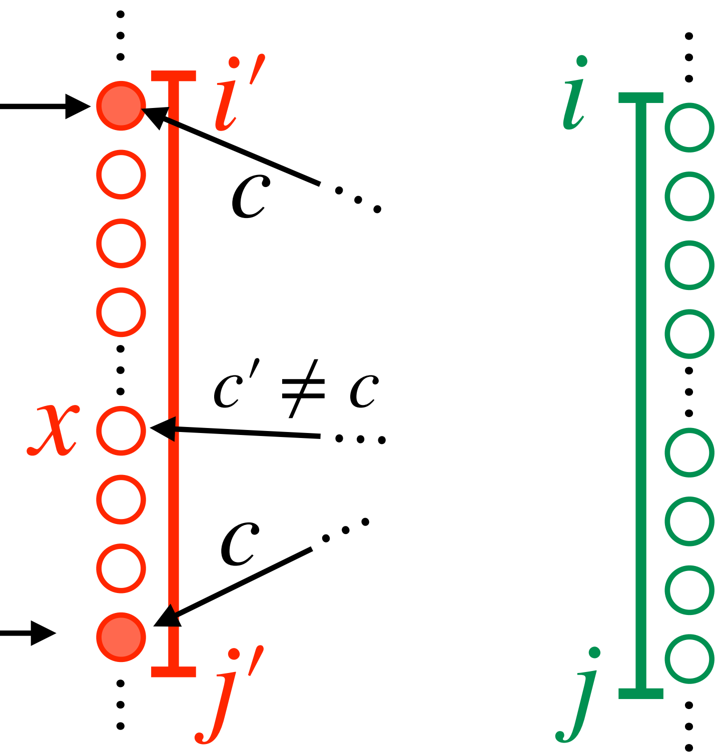


We want to show that the nodes in $[i', j']$ consist *only* of the c-successors of nodes $[i, j]$

Wheeler graphs

As defined, i' is reachable via an edge labeled c from a node in $[i, j]$

Same for j'



Consider node x, where $i' < x < j'$ with incoming edge labeled c'. Suppose $c' \neq c$.

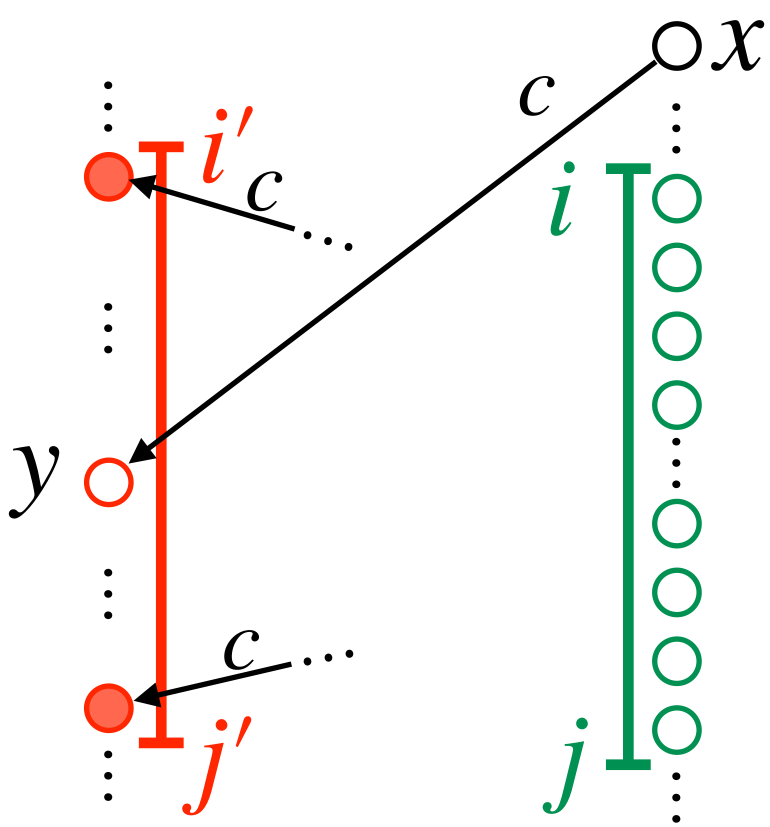
Recall: $a \prec a' \implies v < v'$

Since $x \not< i'$, we have $c' \nprec c$

Since $j' \not< x$, we have $c \nprec c'$

We have $c' \succcurlyeq c$, $c \succcurlyeq c'$, and $c' \neq c$, giving a contradiction

Wheeler graphs



Could node $x \notin [i, j]$ be a $c-$predecessor of a node y, $i' < y < j'$?

No. Proof is by contradiction, similar to previous argument and using:

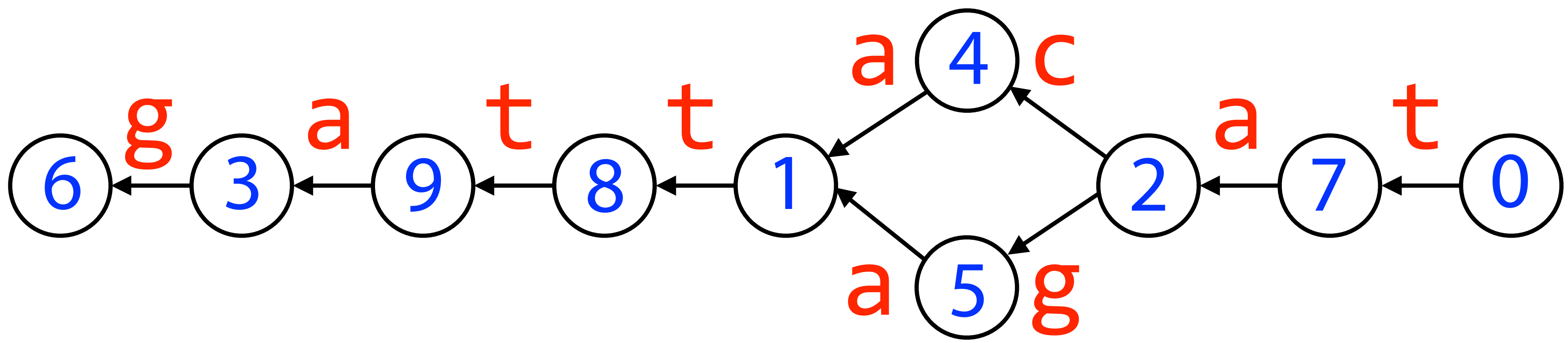
$$(a = a') \wedge (u < u') \implies v \leq v'$$

Wheeler graphs

For any consecutive range $[i, j]$ of nodes and string α, the nodes reached by following edges matching α also form a consecutive range

Proof takes previous arguments and extends them to string α inductively

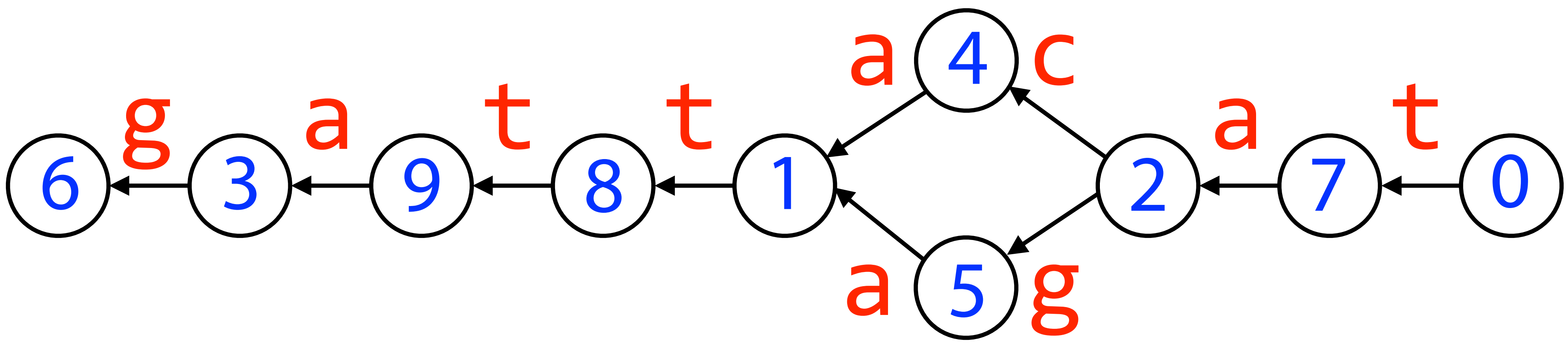
Wheeler graphs



How would we represent a Wheeler graph with bitvectors?

Need to represent *structure* as well as *node and edge labels*

Wheeler graphs

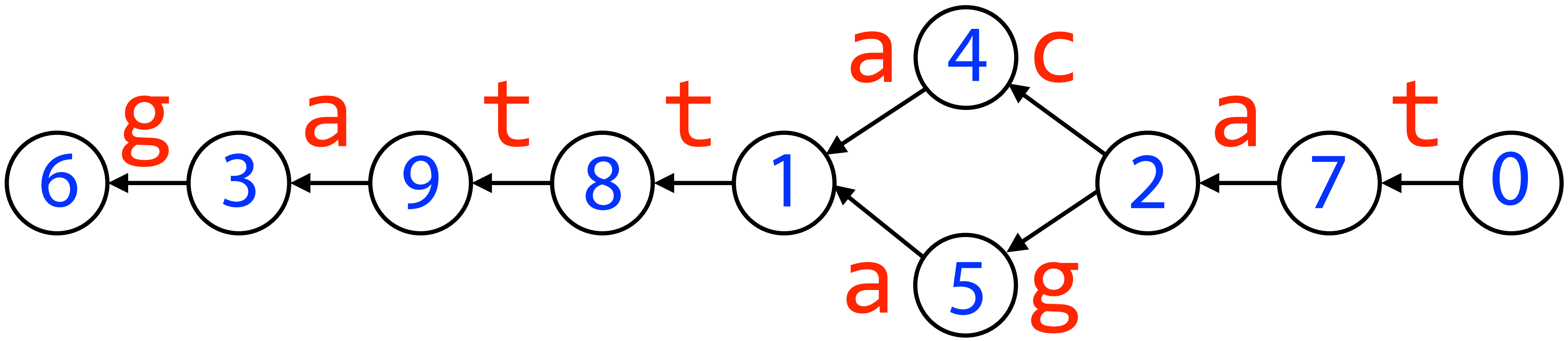


Idea 1: Encode in- and outdegree of each node in unary

Idea 2: Concatenate in order by node

Here, O = 01010010101011010101
0 1 2 3 4 56 7 8 9

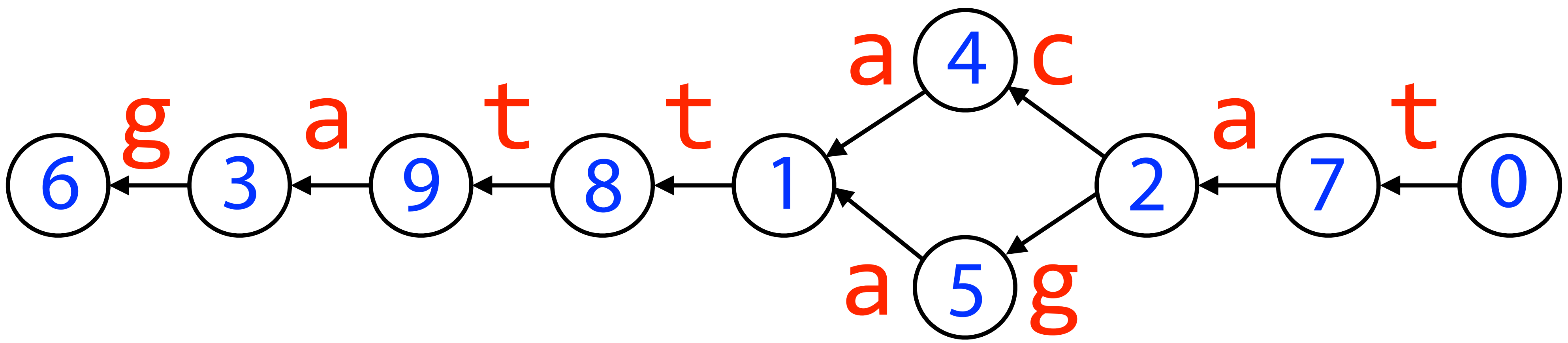
Wheeler graphs



$I =$ 10010101010101010101

0 1 2 3 4 5 6 7 8 9

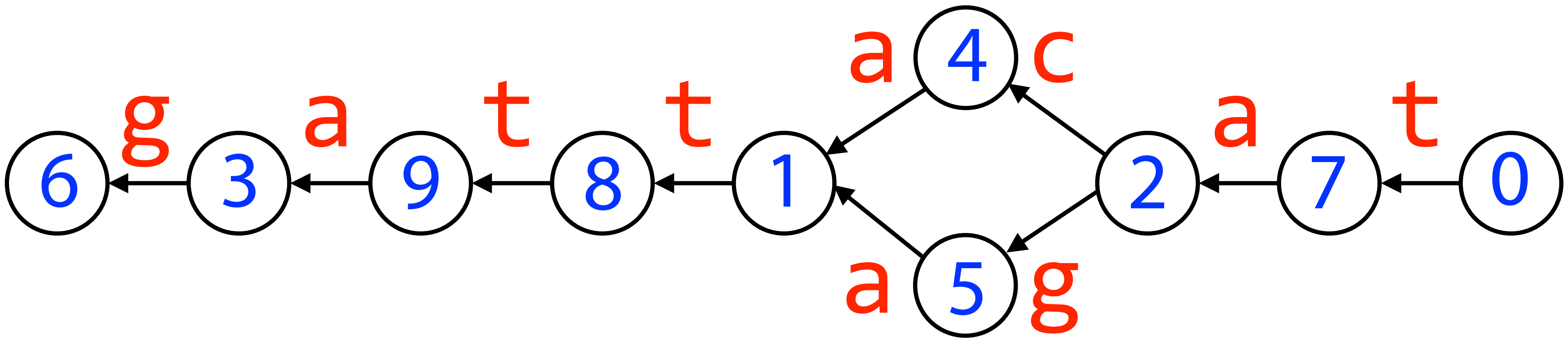
Wheeler graphs



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

$O =$ 01010010101011010101
$L =$ t t cg g a a a t a

Wheeler graphs



$I =$ 10010101010101010101

$O =$ 01010010101010110101

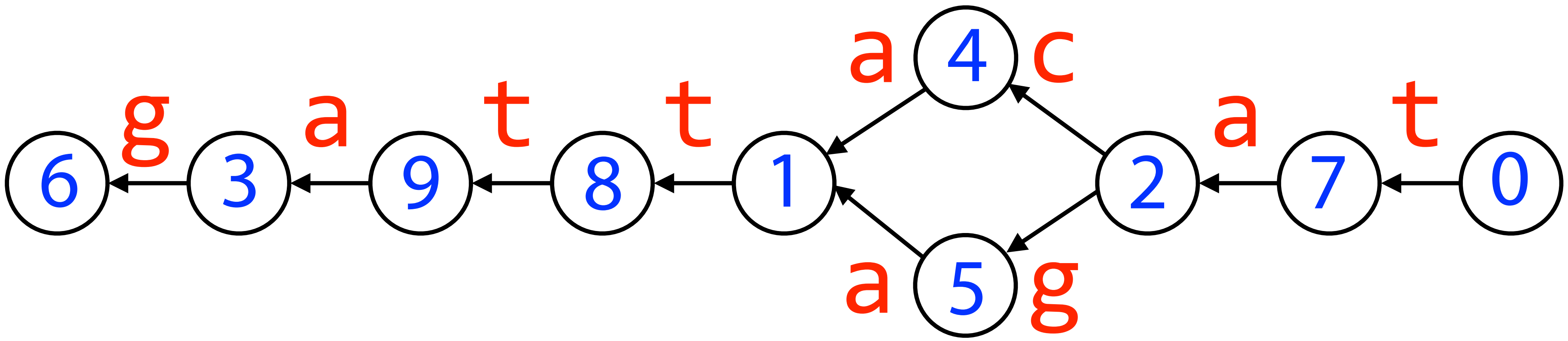
$L =$ ttcggaaata

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

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

How long is L? (# edges) chars

Wheeler graphs



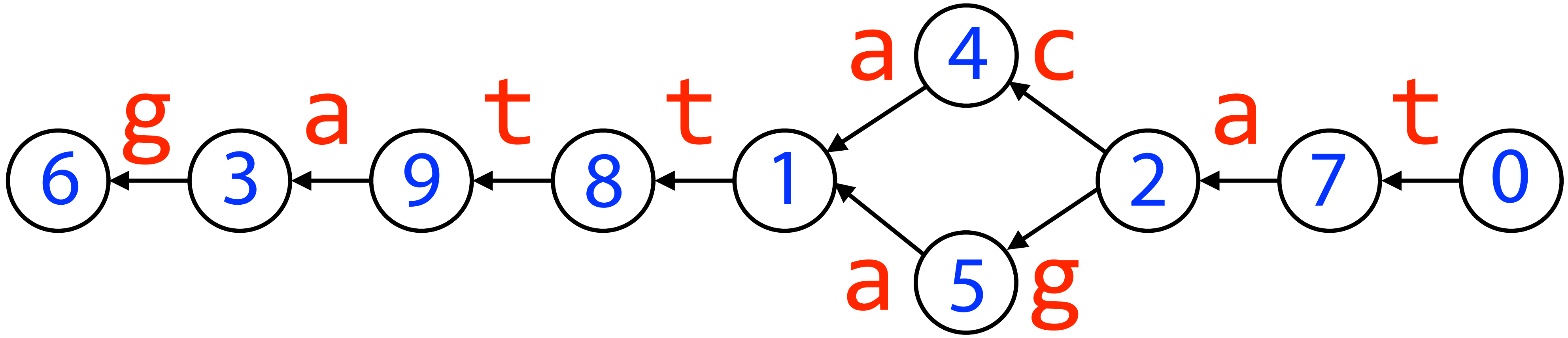
$$I = 10010101010101010101$$

$$O = 01010010101011010101$$

$$L = \texttt{ttcggaaata}$$

How to find in- or outdegree of a node?

Wheeler graphs



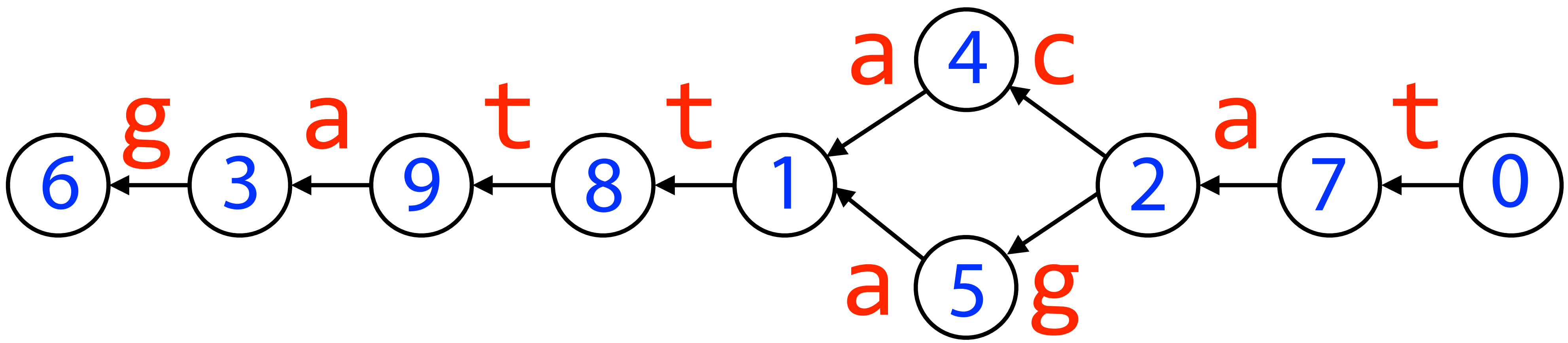
$$I = \texttt{10010101010101010101}$$

$$G.\text{indegree}(i) = I.\text{select}_1(i) - I.\text{select}_1(i-1) - 1$$

(let I. select(-1) = 0)

$$G.\text{indegree}(1) = I.\text{select}_1(1) - I.\text{select}_1(0) - 1$$
$$= 3 - 0 - 1 = 2$$

Wheeler graphs



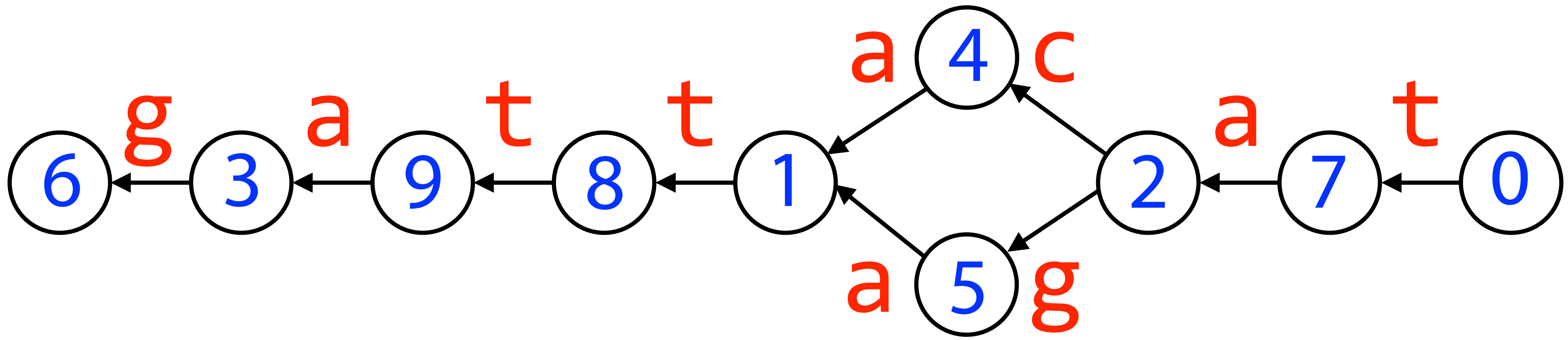
$$O = 0101001010101011010101$$

$$G.\text{outdegree}(i) = O.\text{select}_1(i) - O.\text{select}_1(i-1) - 1$$

(let O . select(-1) = 0)

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

Wheeler graphs



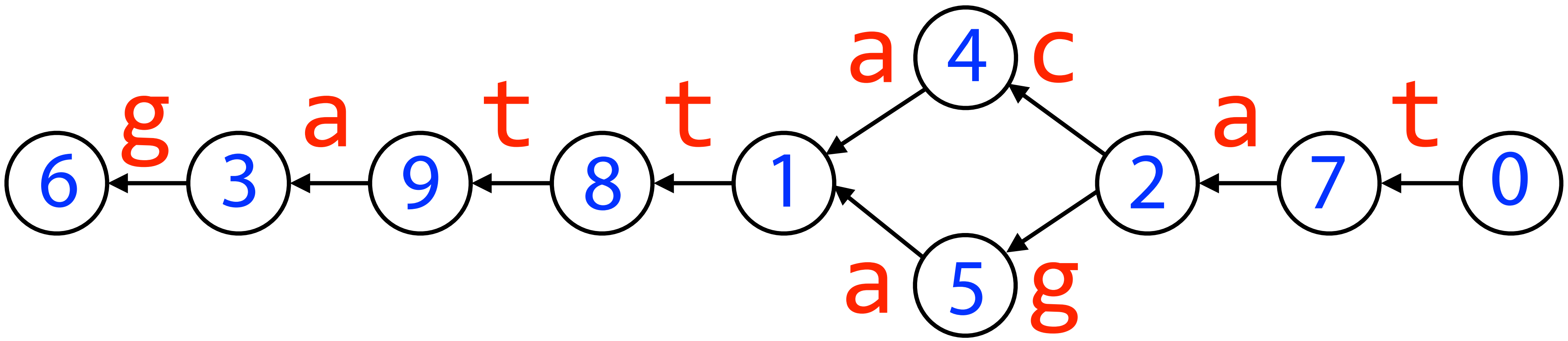
$$I = 1001010101010101010101$$

$$O = 0101001010101011010101$$

$$L = \texttt{ttcggaaata}$$

How to access the labels of the edges outgoing from a node?

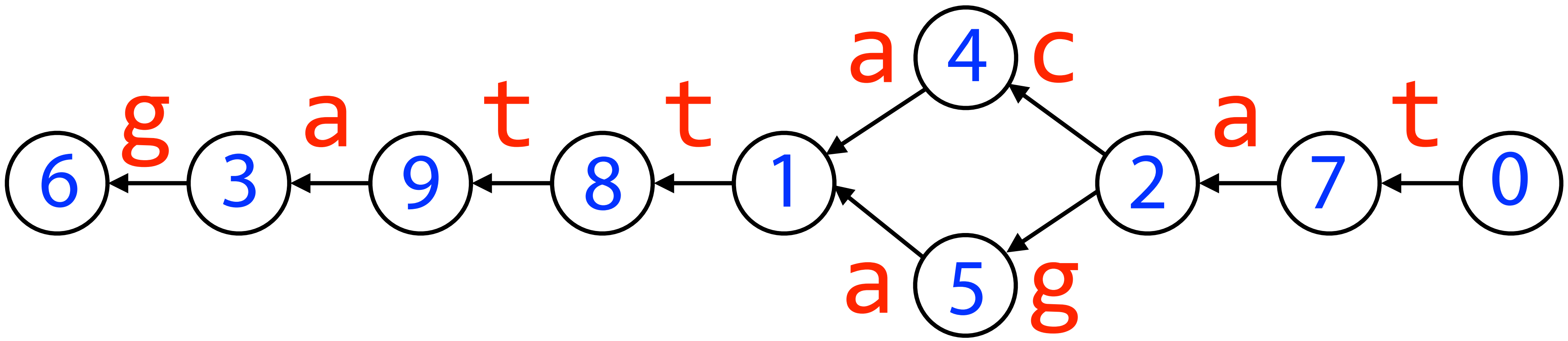
Wheeler graphs



$$O = \texttt{01010010101011010101}$$
$$L = \texttt{ttcggaaata}$$

$$\text{off} = O\,.\,\text{rank}_0(O\,.\,\text{select}_1(i-1))$$

Wheeler graphs



$O =$ 01010010101011010101

$L =$ ttcggaaata

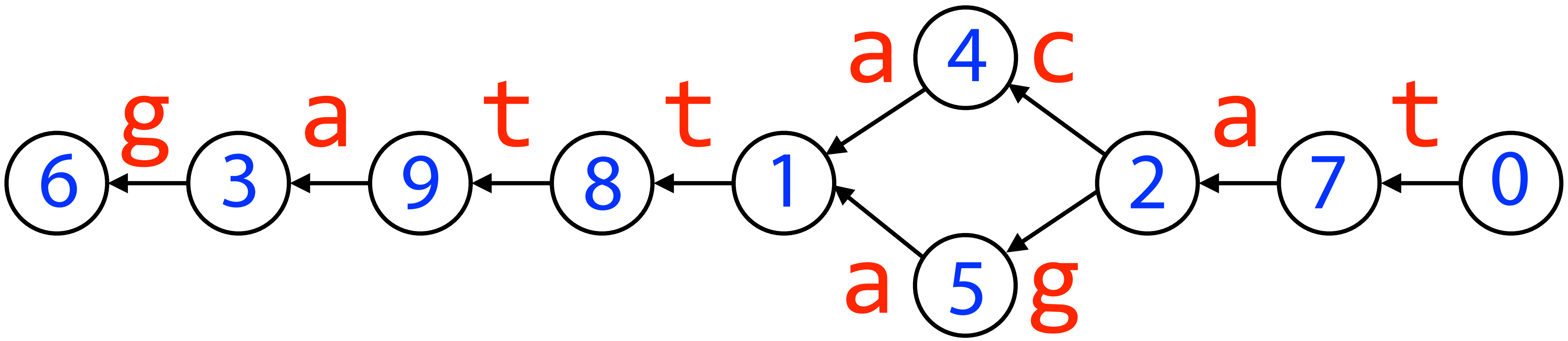
num $= G$. outdegree(i)

off $= O$. rank$_0(O$. select$_1(i-1))$

for j in {0, 1, …, num-1}:

  L . access(off $+ j)$

Wheeler graphs



$O =$ `010100101010110101`

$L =$ `ttcggaaata`

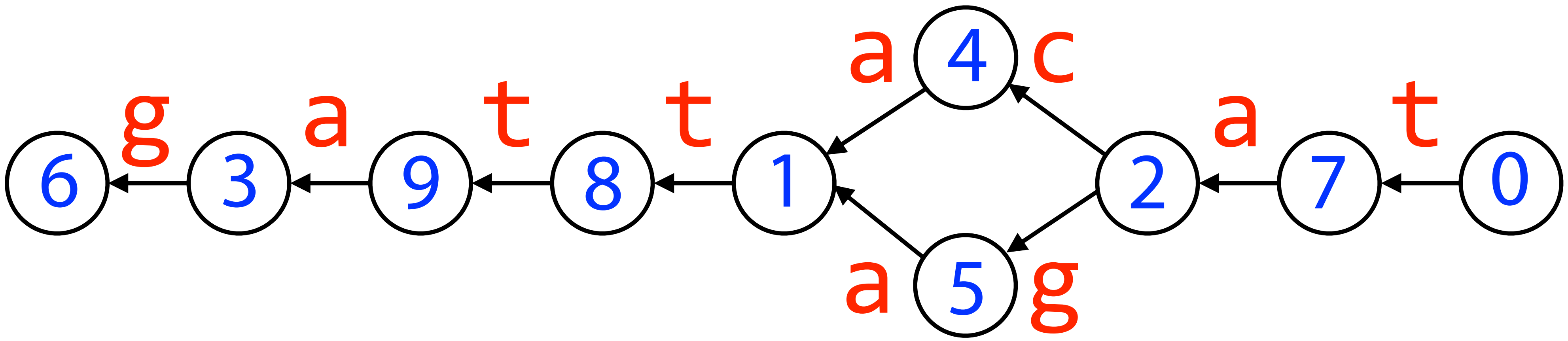
→ num $= G\,.\,\text{outdegree}(2) =$ **2**

off $= O\,.\,\text{rank}_0(O\,.\,\text{select}_1(2-1))$

for j in {0, 1, …, num-1}:

  $L\,.\,\text{access}(\text{off} + j)$

Wheeler graphs



$O =$ `0101001010101011010101`

$L =$ `ttcggaaata`

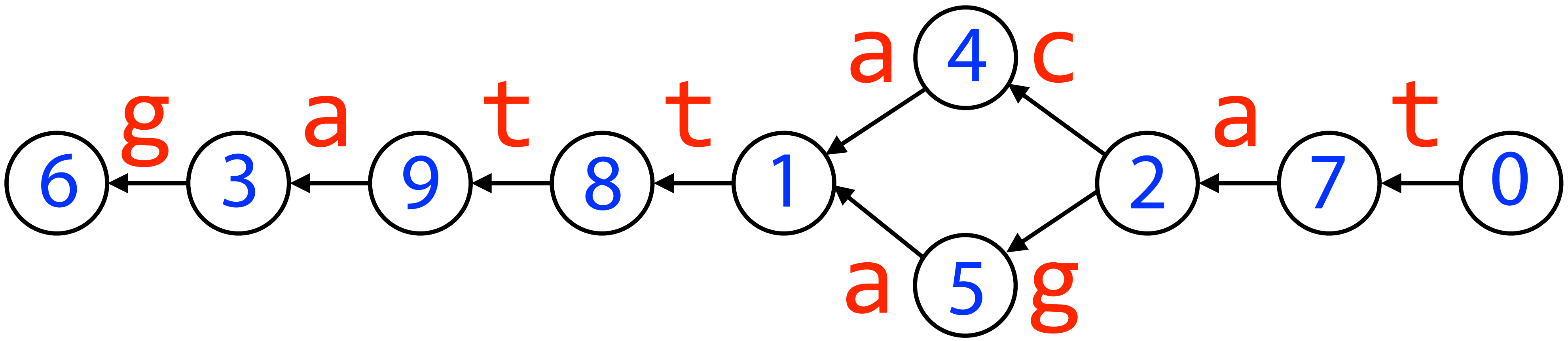
num $= 2$

off $= O\,.\,\text{rank}_0(O\,.\,\text{select}_1(2-1) = 3)$

for j in {0, 1, …, num-1}:

$L\,.\,\text{access}(\text{off} + j)$

Wheeler graphs



$O = $ 01010010101011010101

$L = $ ttcggaaata

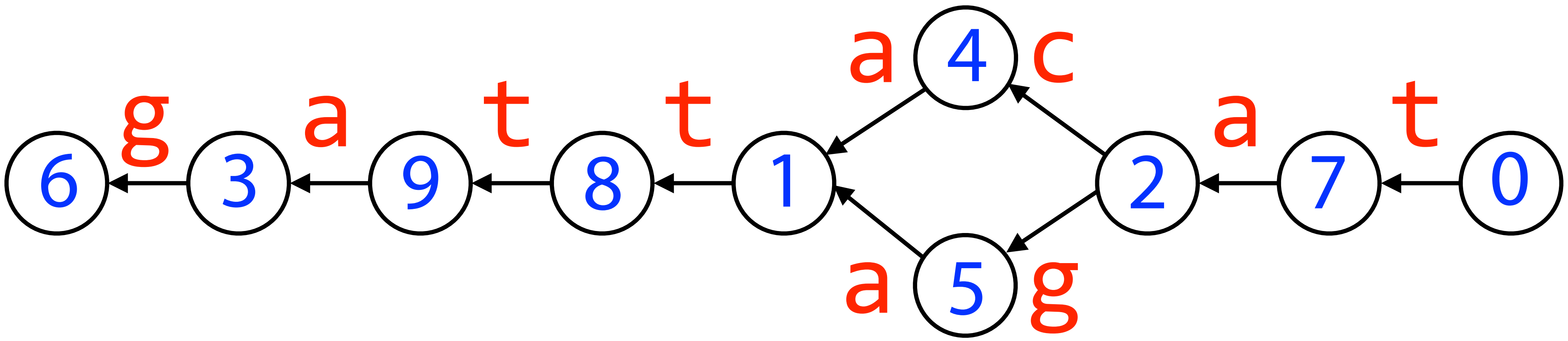
num $= 2$

off $= O\,.\,\text{rank}_0(3) = 2$

for j in {0, 1, …, num-1}:

$L\,.\,\text{access}(\text{off} + j)$

Wheeler graphs



$$O = \texttt{010100101010110101 01}$$

$$L = \texttt{ttcggaaata}$$

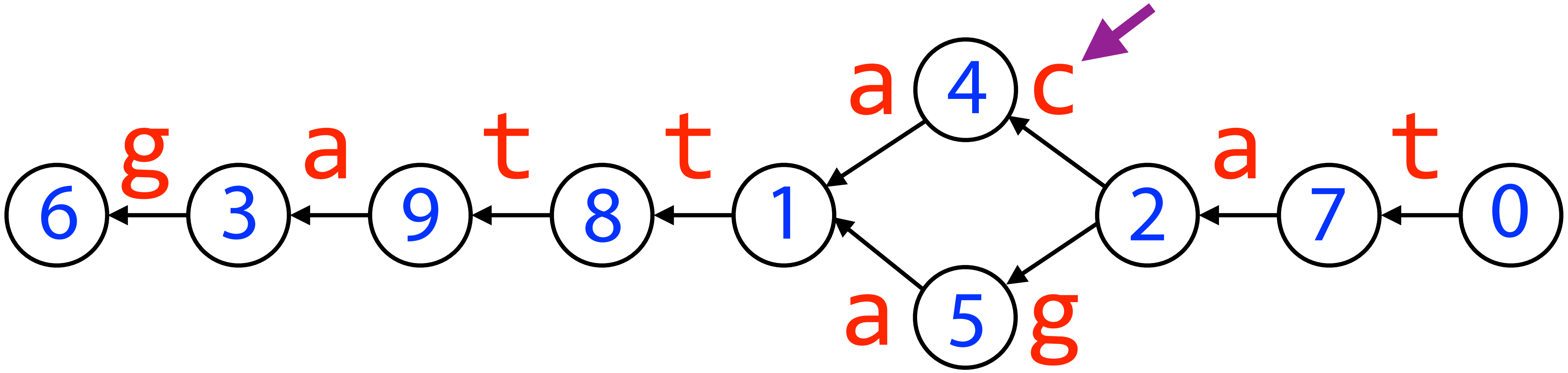
num $= 2$

off $= 2$

→ for j in $\{0, 1\}$:

  L . access(off $+ j$)

Wheeler graphs



$O =$ 01010010101011010101

$L =$ ttcggaaata

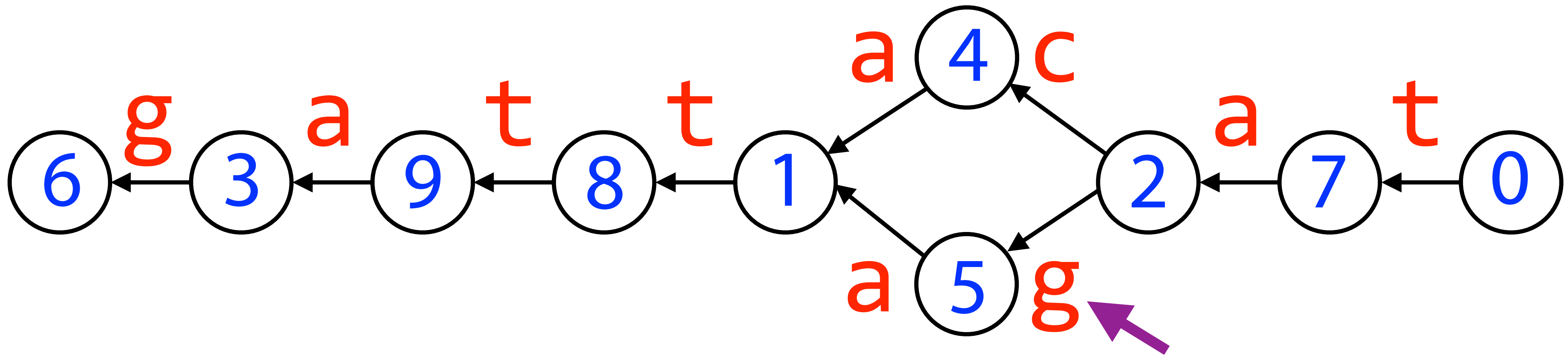
num $= 2$

off $= 2$

for j in {0, 1}:

L.access$(2 + 0) =$ c

Wheeler graphs



$$O = \texttt{01010010101011010101}$$

$$L = \texttt{ttcggaaata}$$

num $= 2$

off $= 2$

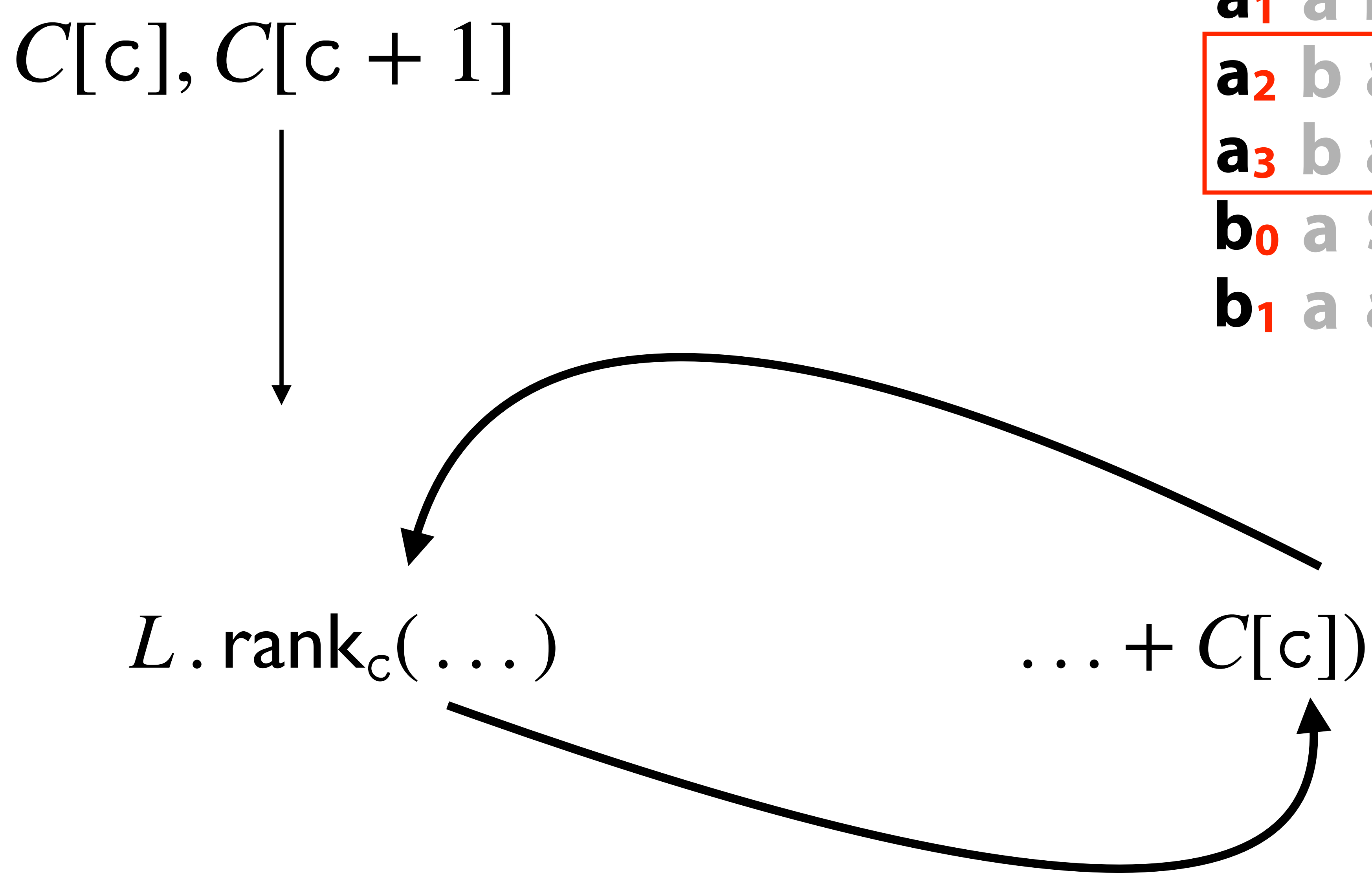
for j in {0, 1}:

$L.\text{access}(2 + 1) = \text{g}$

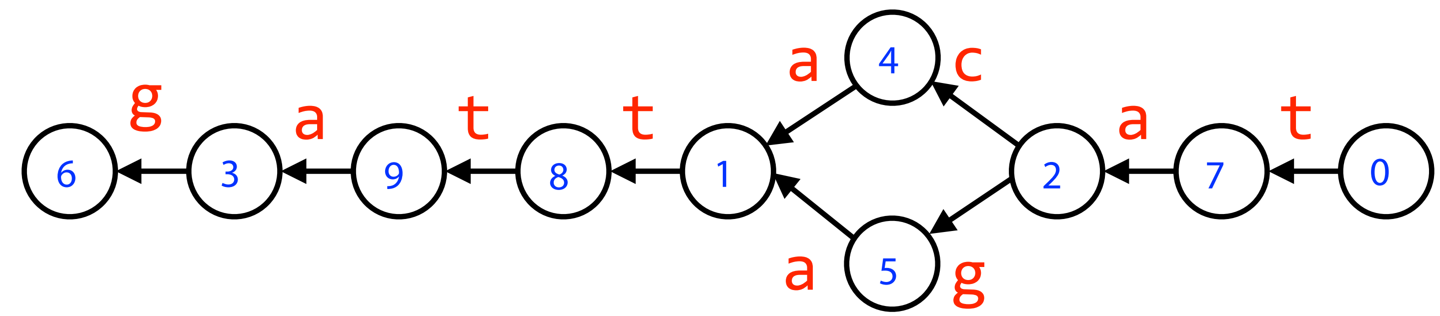
Wheeler graphs

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



Wheeler graph match query:

I : 1001010101010101010101

O : 0101001010101011010101

L : ttcggaaata

$C[\mathtt{c}], C[\mathtt{c}+1]$

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

$I.\text{rank}_1(I.\text{select}_0(\ldots + C[\mathtt{c}]))$

$L.\text{rank}_\mathtt{c}(\ldots)$