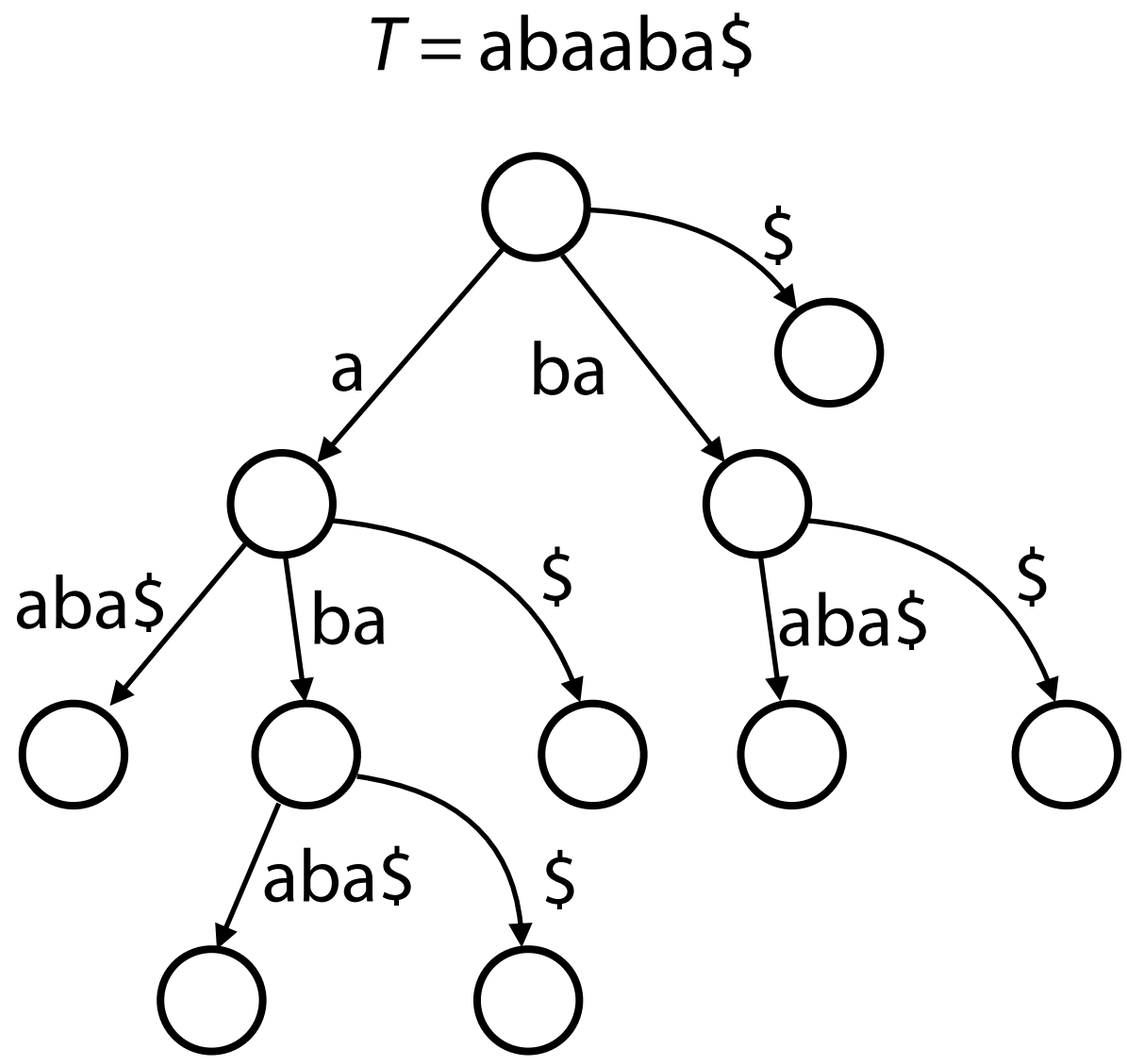
Suffix Trees: basic querying

Ben Langmead

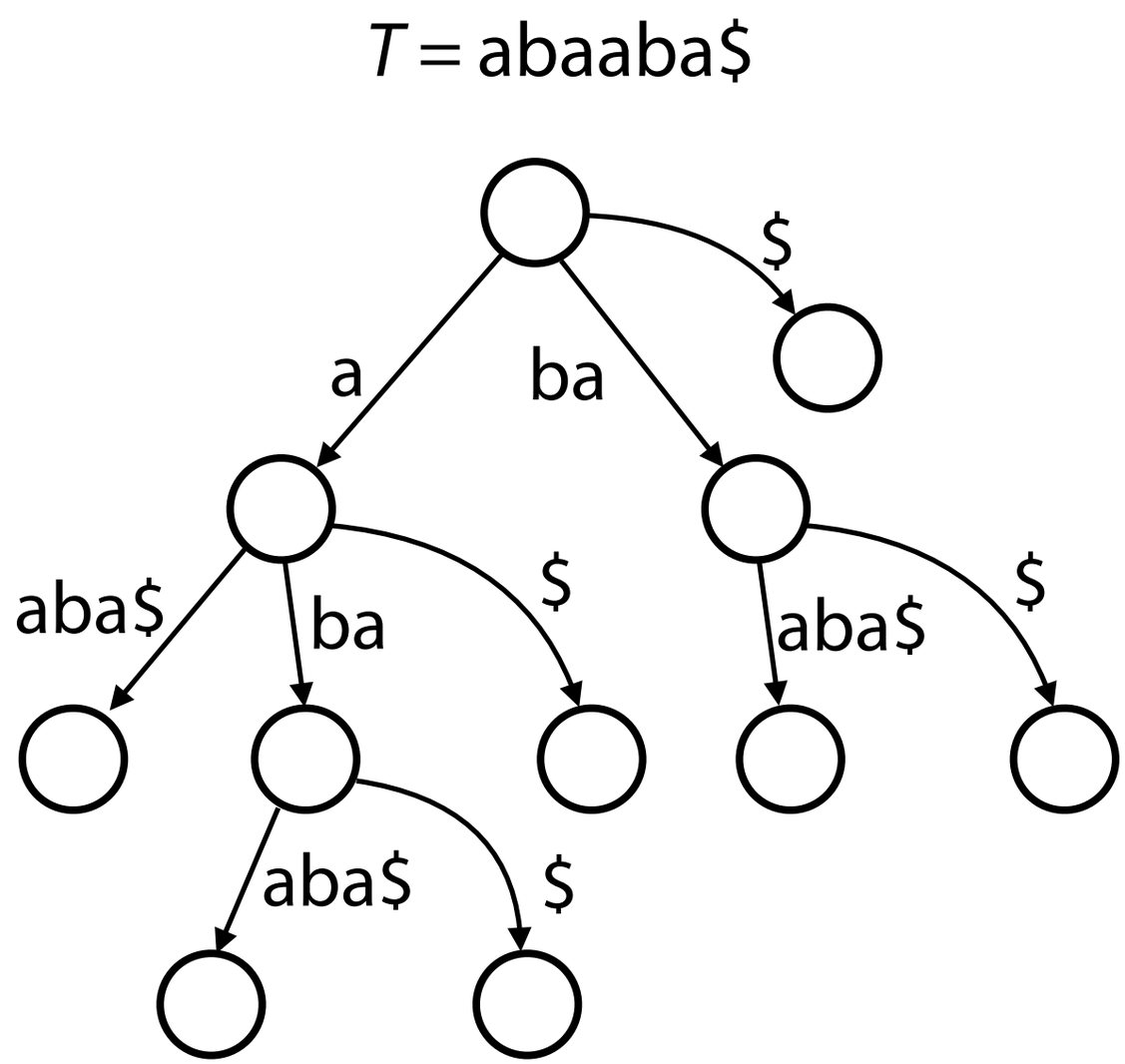
Suffix tree



Suffix tree

How do we check whether a string *S* is a substring of *T*?

Same procedure as for suffix trie, but dealing with coalesced edges

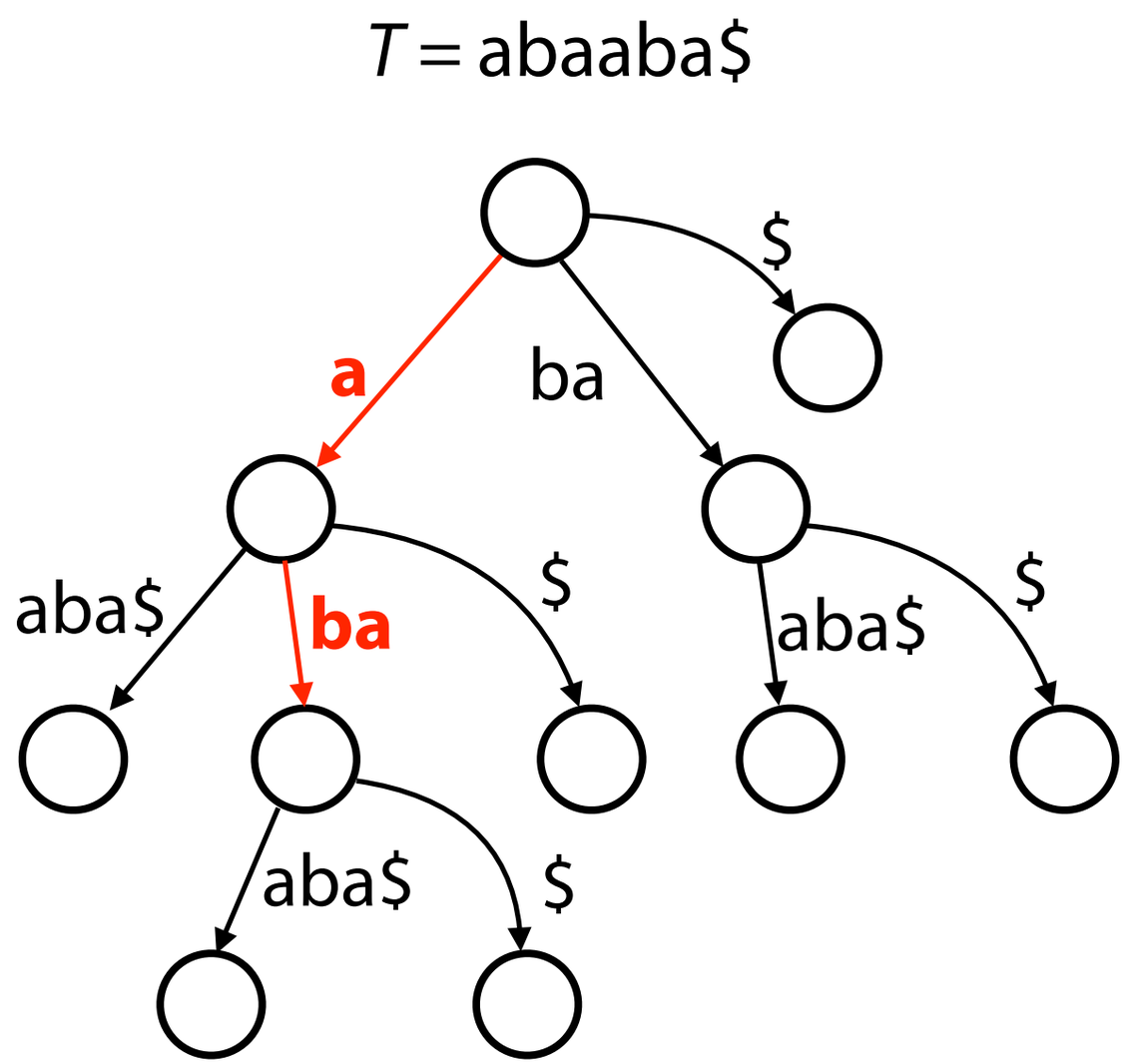


aba

Suffix tree

How do we check whether a string *S* is a substring of *T*?

Same procedure as for suffix trie, but dealing with coalesced edges

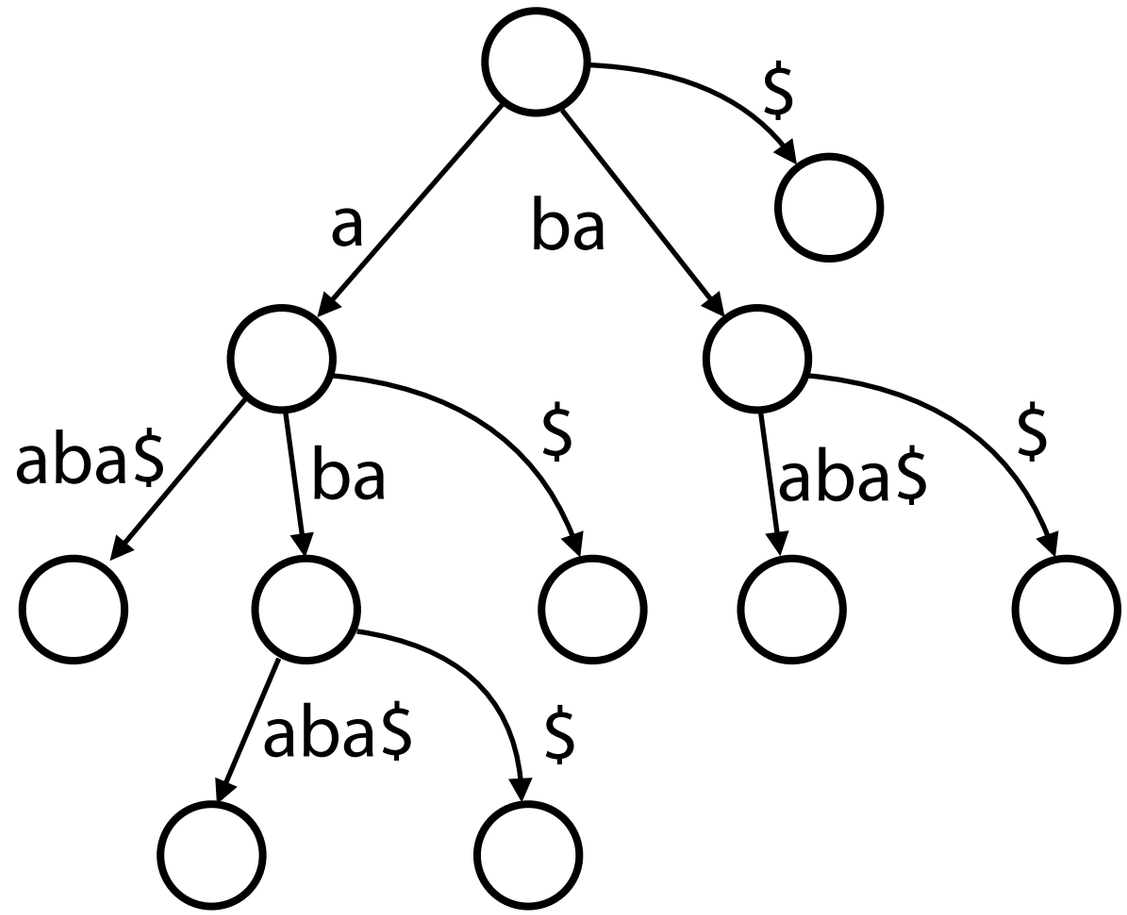


aba yes

Suffix tree

How do we check whether a string *S* is a substring of *T*?

Same procedure as for suffix trie, but dealing with coalesced edges



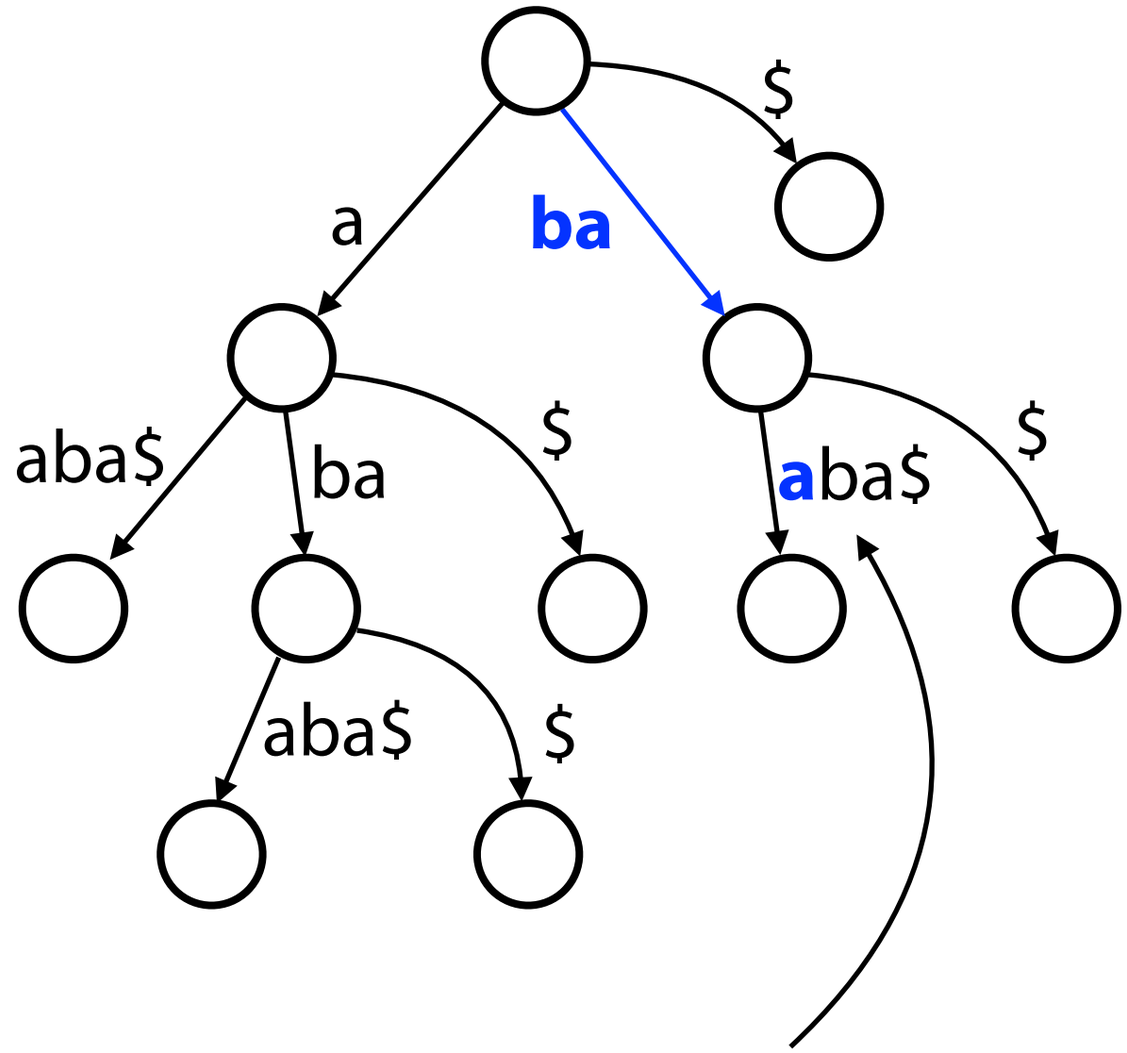
aba yes

baa

Suffix tree

How do we check whether a string *S* is a substring of *T*?

Same procedure as for suffix trie, but dealing with coalesced edges



aba yes

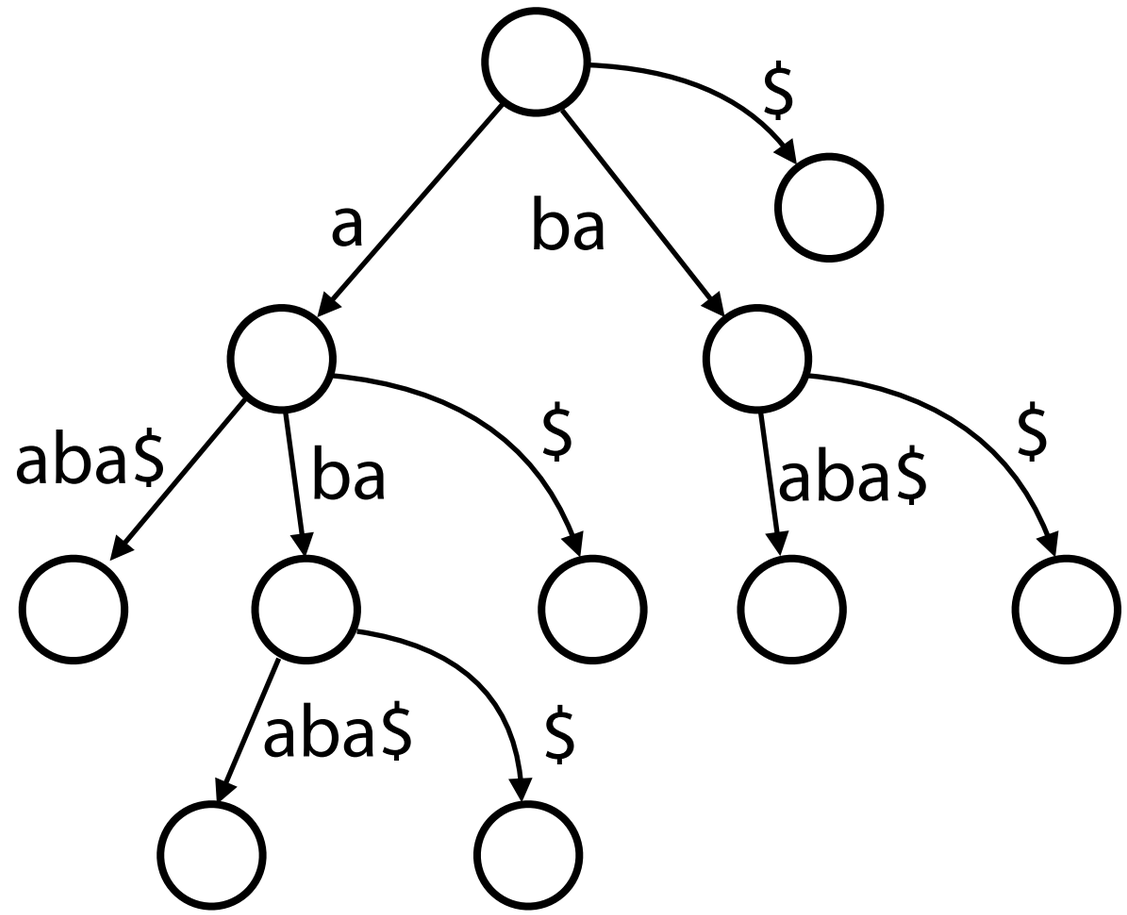
baa yes

Notice our walk ended in the *middle* of an edge label

Suffix tree

How do we check whether a string *S* is a substring of *T*?

Same procedure as for suffix trie, but dealing with coalesced edges



aba yes

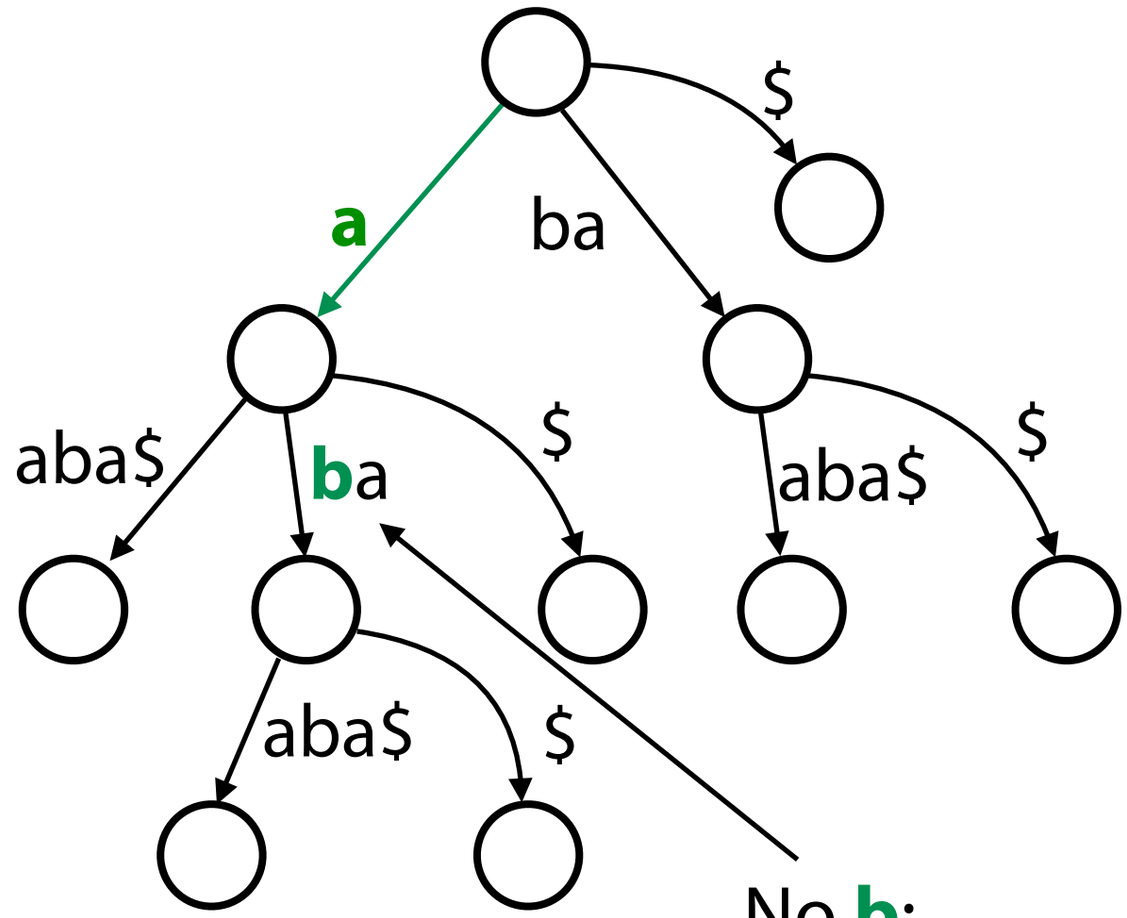
baa yes

abb

Suffix tree

How do we check whether a string *S* is a substring of *T*?

Same procedure as for suffix trie, but dealing with coalesced edges



aba yes

baa yes

abb no

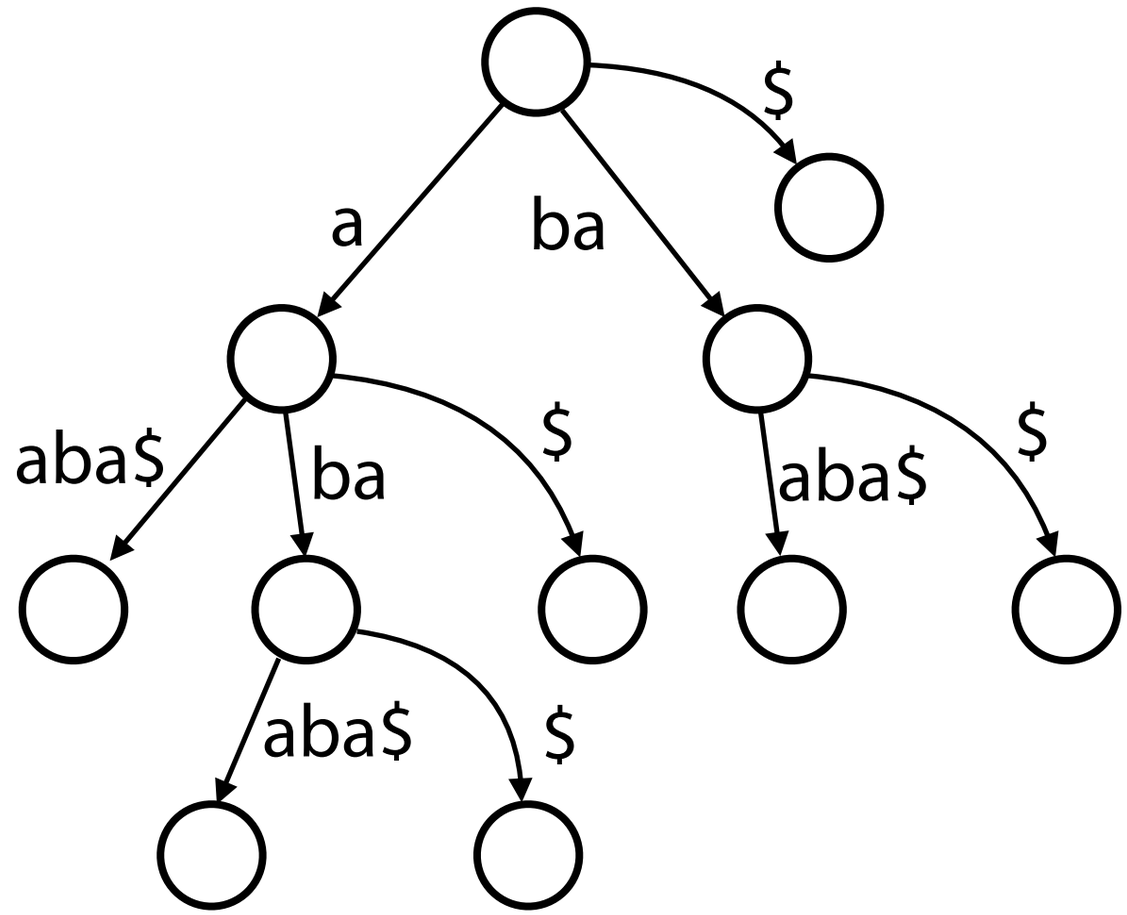
In middle of edge again!

Suffix tree

Time required to match a query string of length *n?*

Still O(*n*), like suffix trie

Some steps advance only along an edge, others advance to a new node; both are O(1)

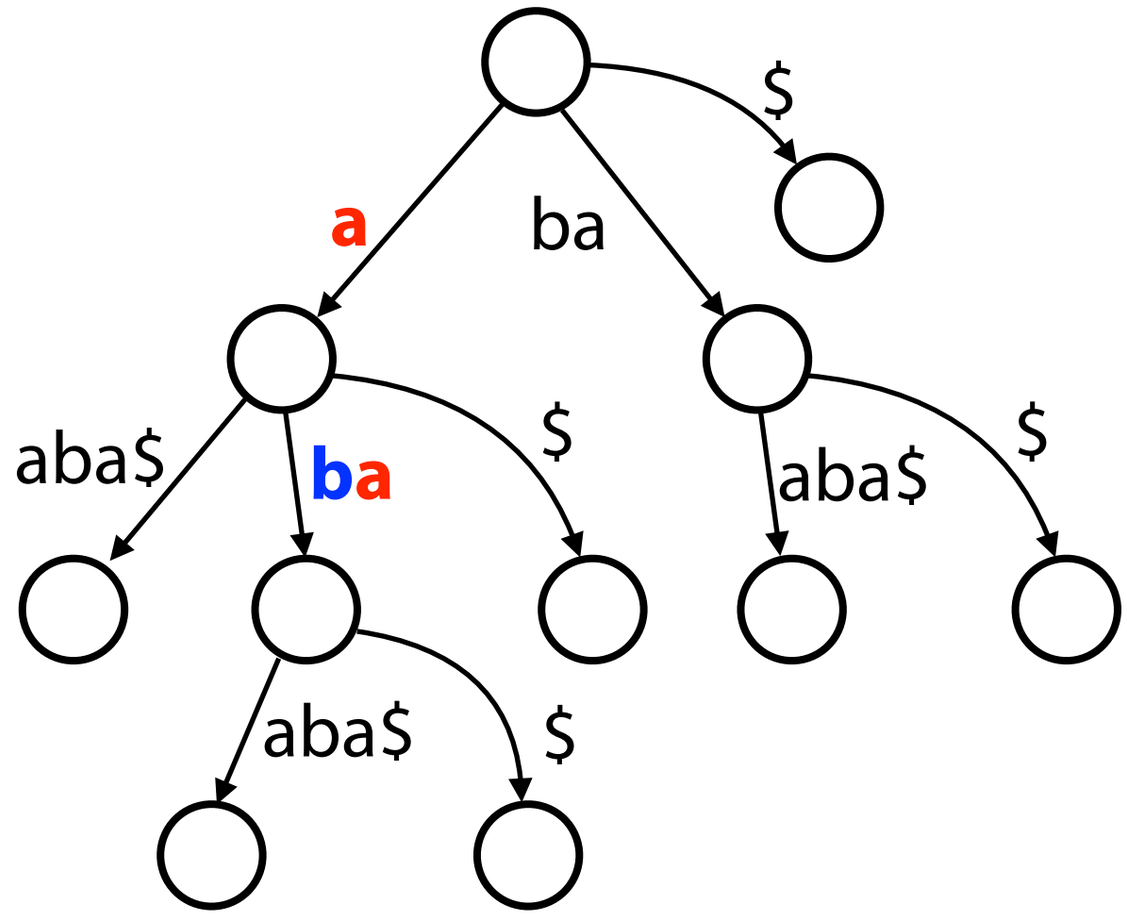


Suffix tree

Time required to match a query string of length *n?*

Still O(*n*), like suffix trie

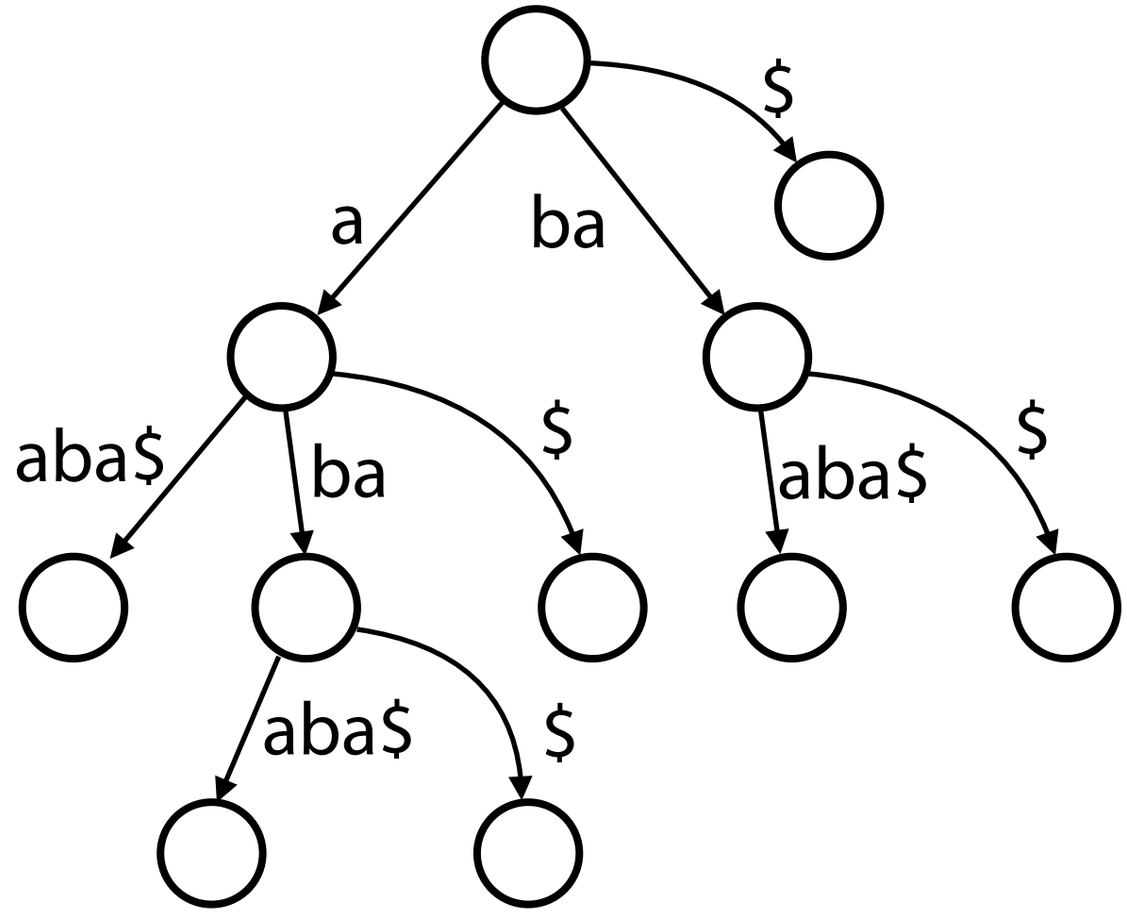
Some steps advance only along an edge, others advance to a new node; both are O(1)



Suffix tree

How do we count the **number of times** a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then count leaves below

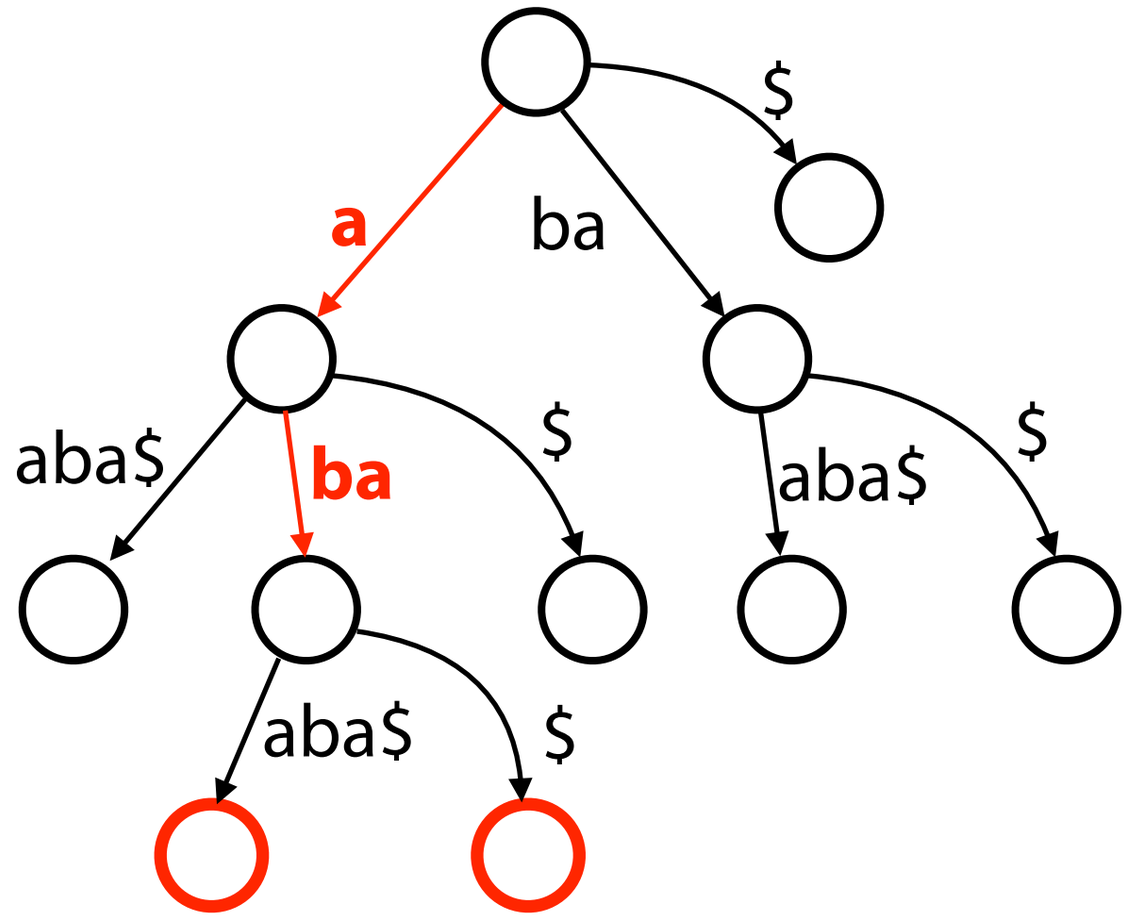


aba

Suffix tree

How do we count the **number of times** a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then count leaves below

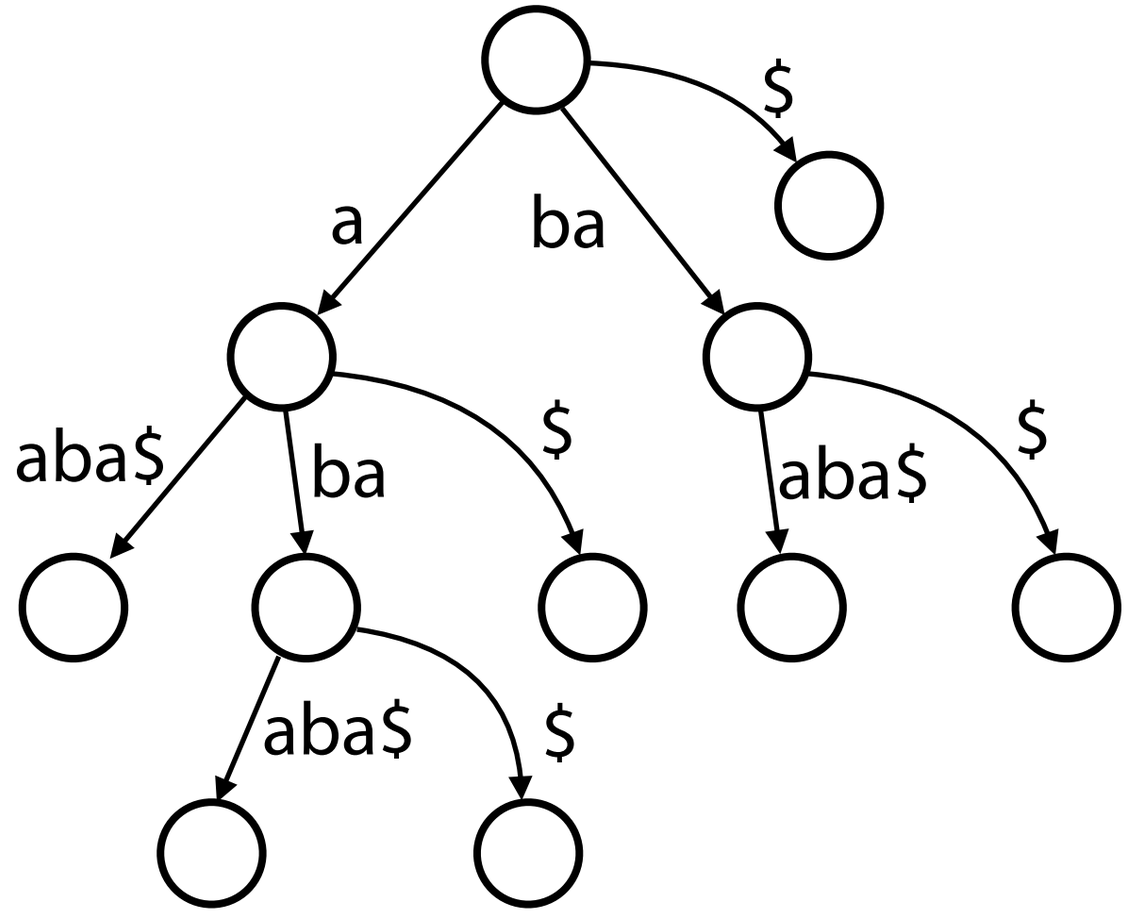


aba 2

Suffix tree

How do we count the **number of times** a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then count leaves below



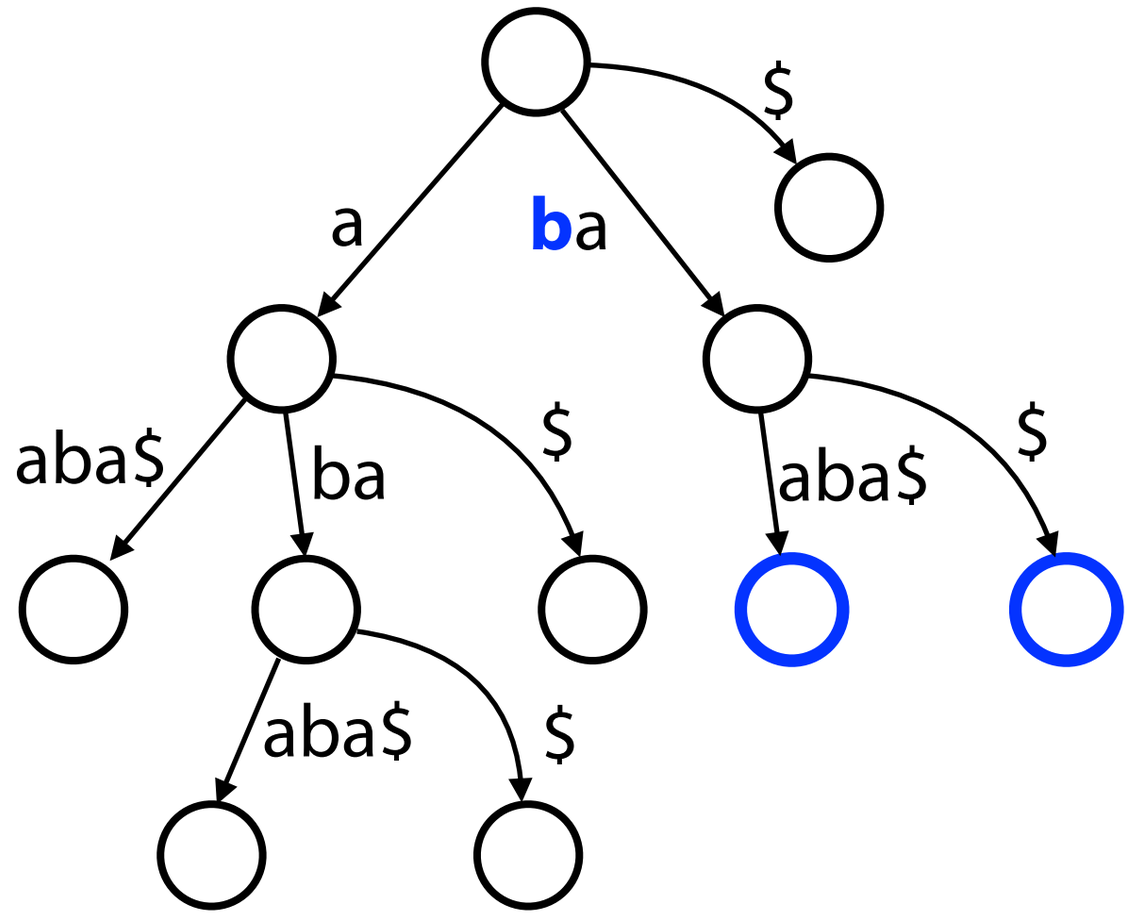
aba 2

b

Suffix tree

How do we count the **number of times** a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then count leaves below

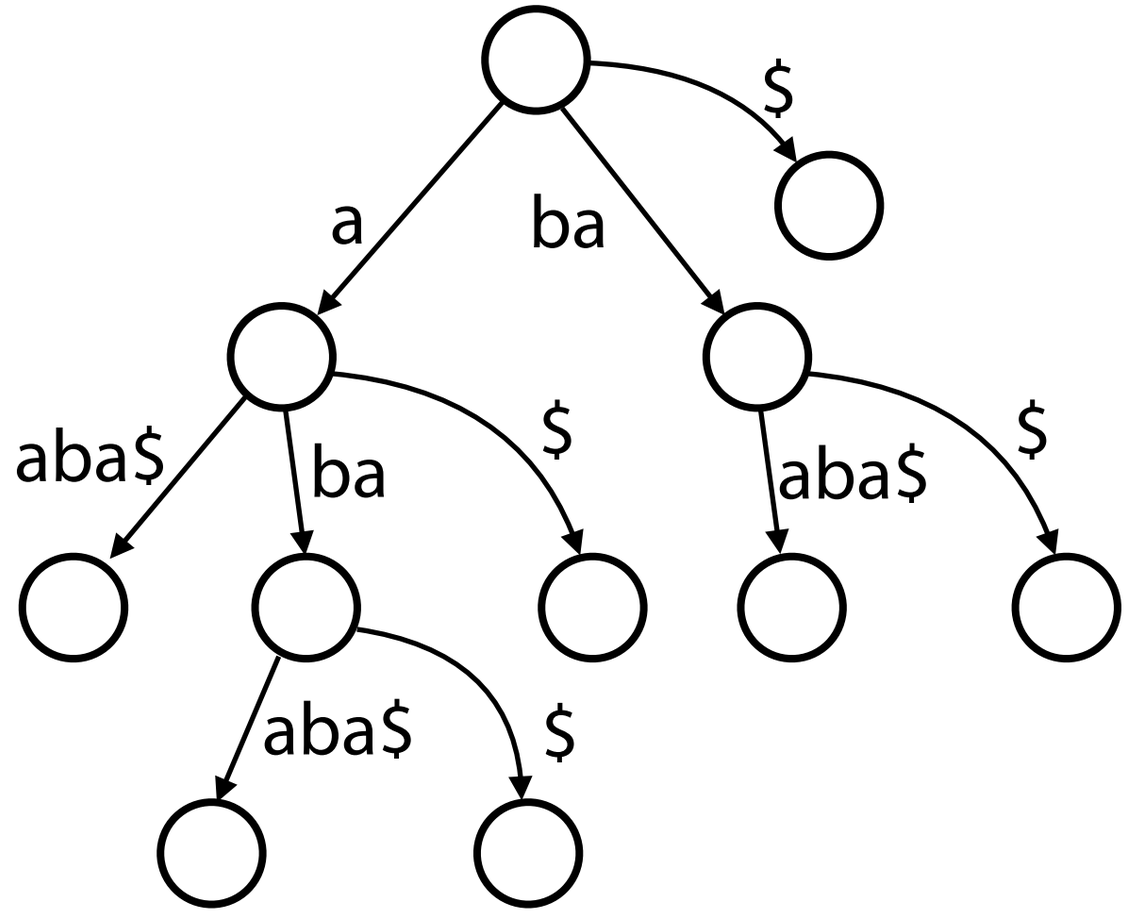


aba	2
b	2

Suffix tree

How do we count the **number of times** a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then count leaves below



aba 2

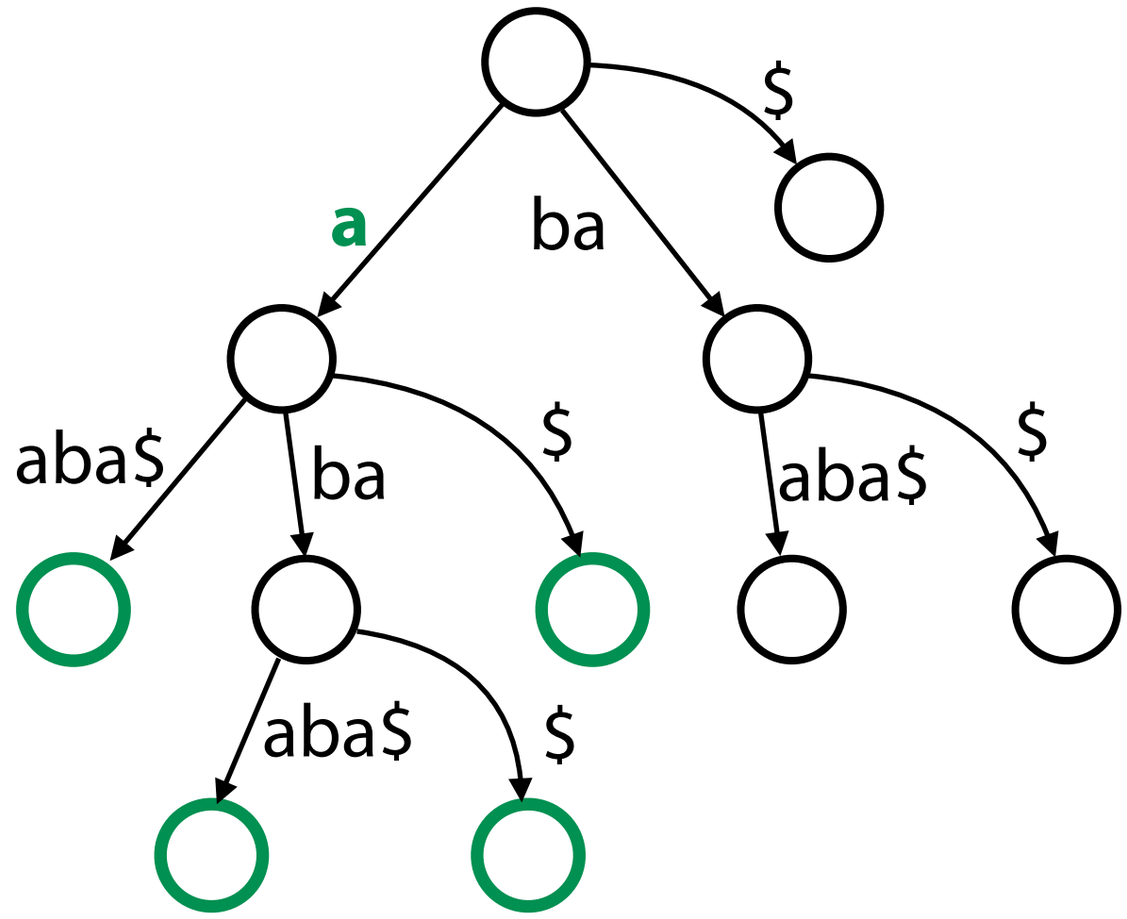
b 2

a

Suffix tree

How do we count the **number of times** a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then count leaves below

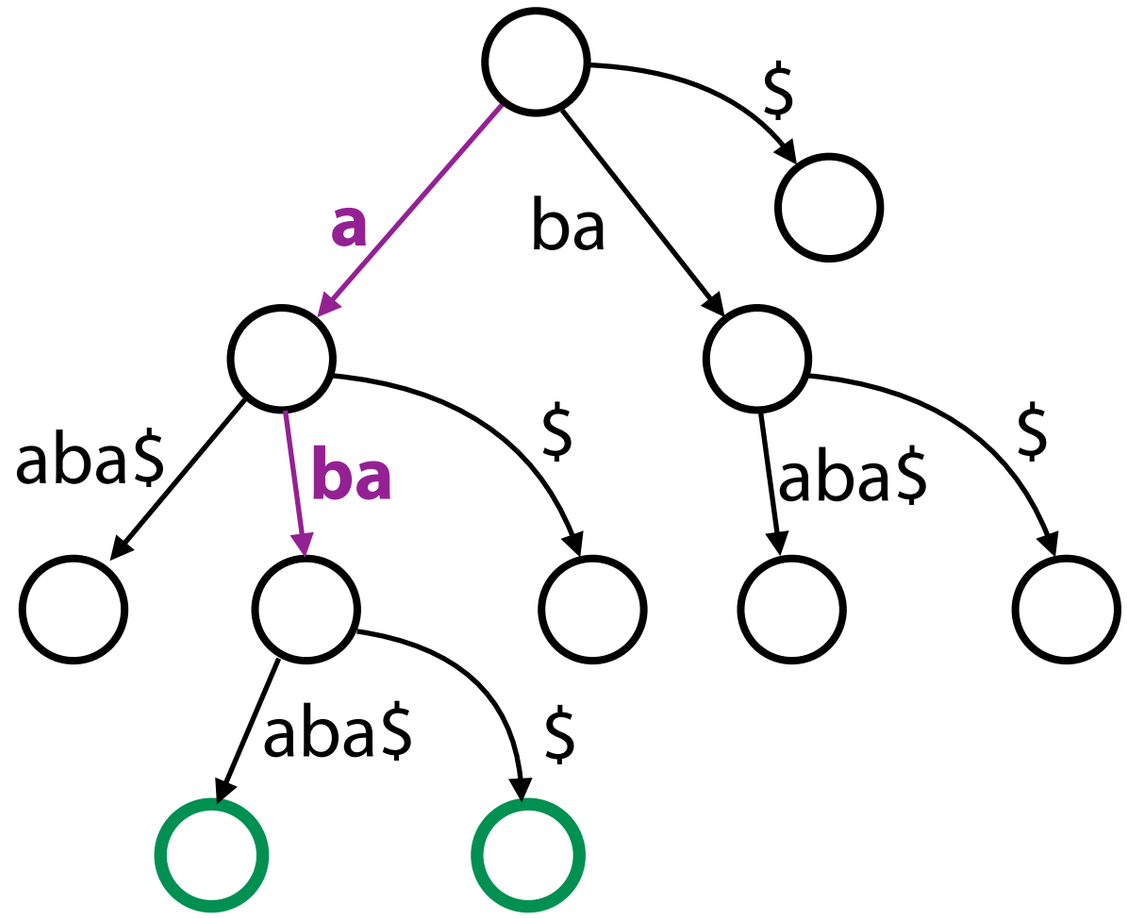


aba	2
b	2
a	4

Suffix tree

Walk down according to *S*, then
count leaves below

How much work?



Two parts:

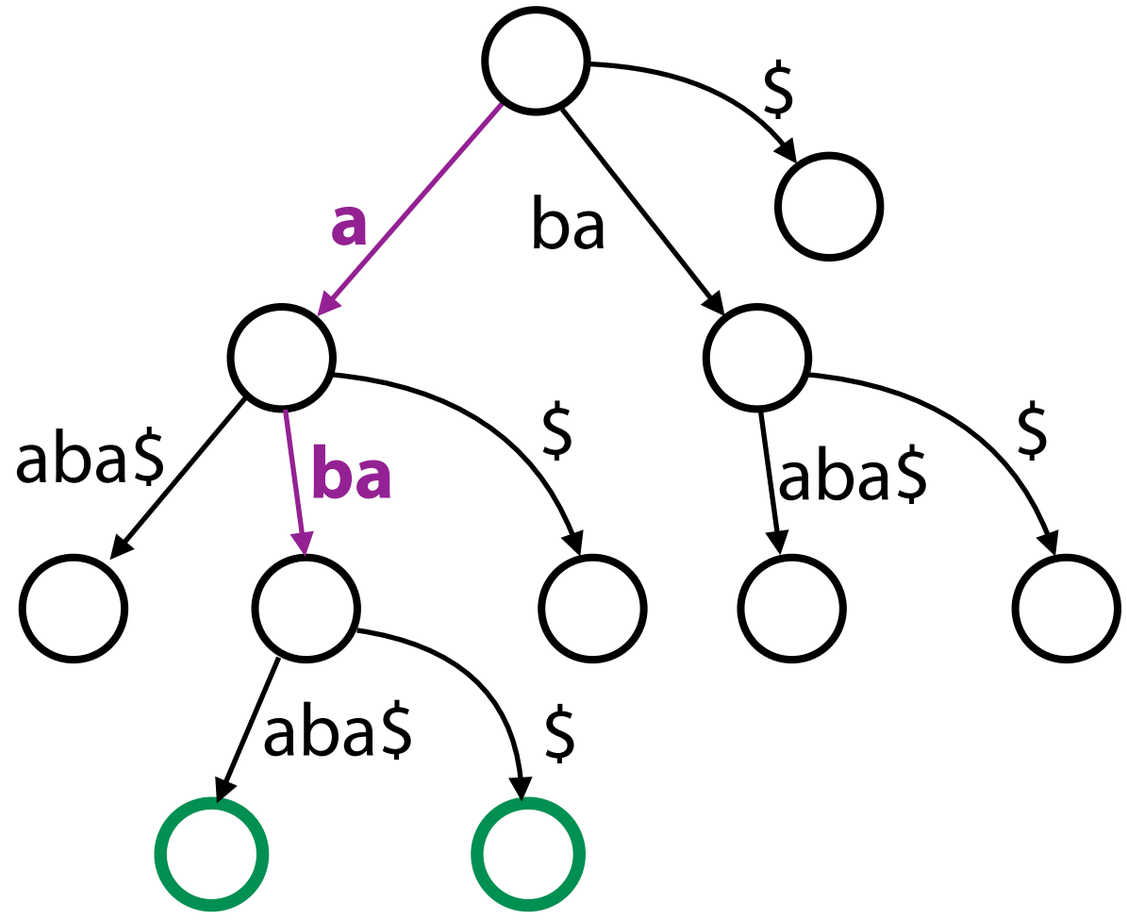
Walk down according to *S*	Count leaves below
O(*n*) (by our usual argument)	???

Suffix tree

Let k = # leaves below

The work of counting is simply the work of **traversing** the subtree

This work is proportional to the # nodes in a subtree with k leaves



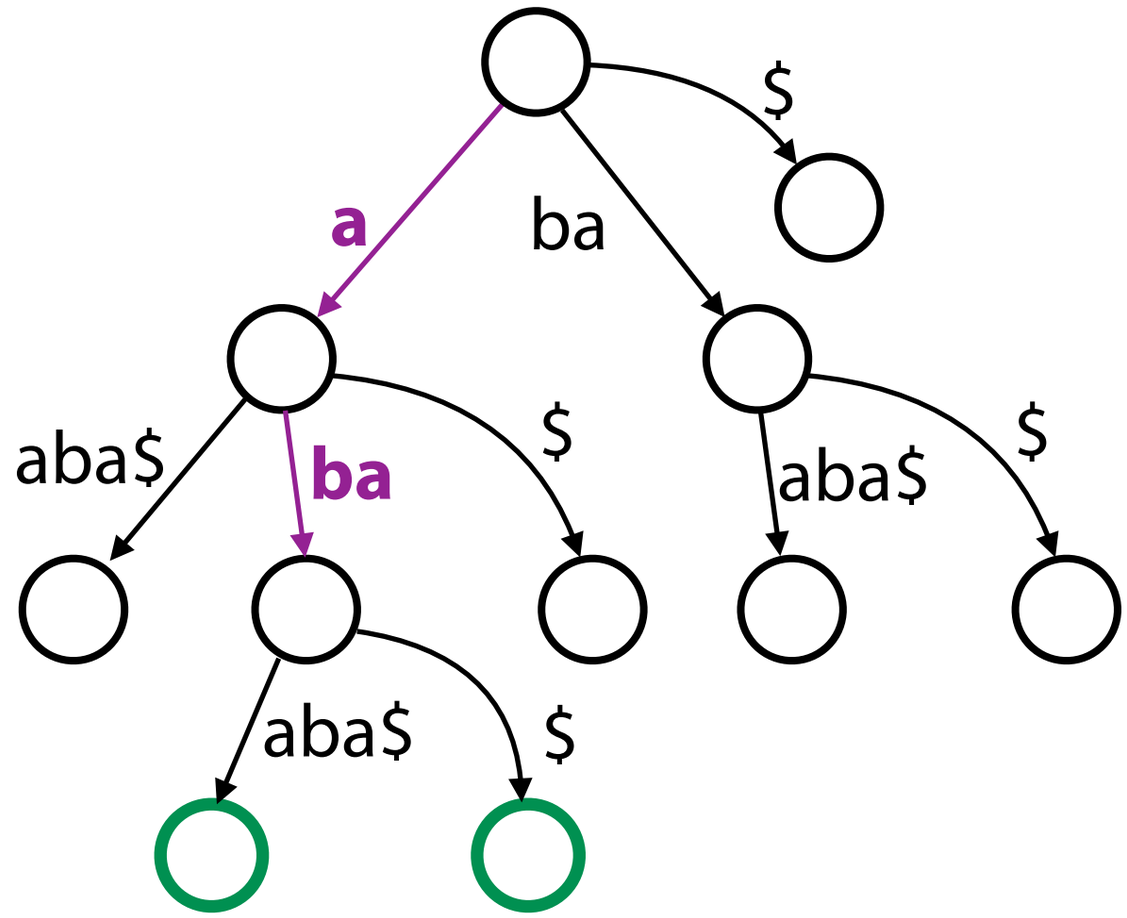
O(k)

- Nodes in subtree, by same no-only-child principle we used to argue suffix tree has O(m) nodes
- Time to traverse the subtree

Suffix tree

Walk down according to *S*, then
count leaves below

How much work?



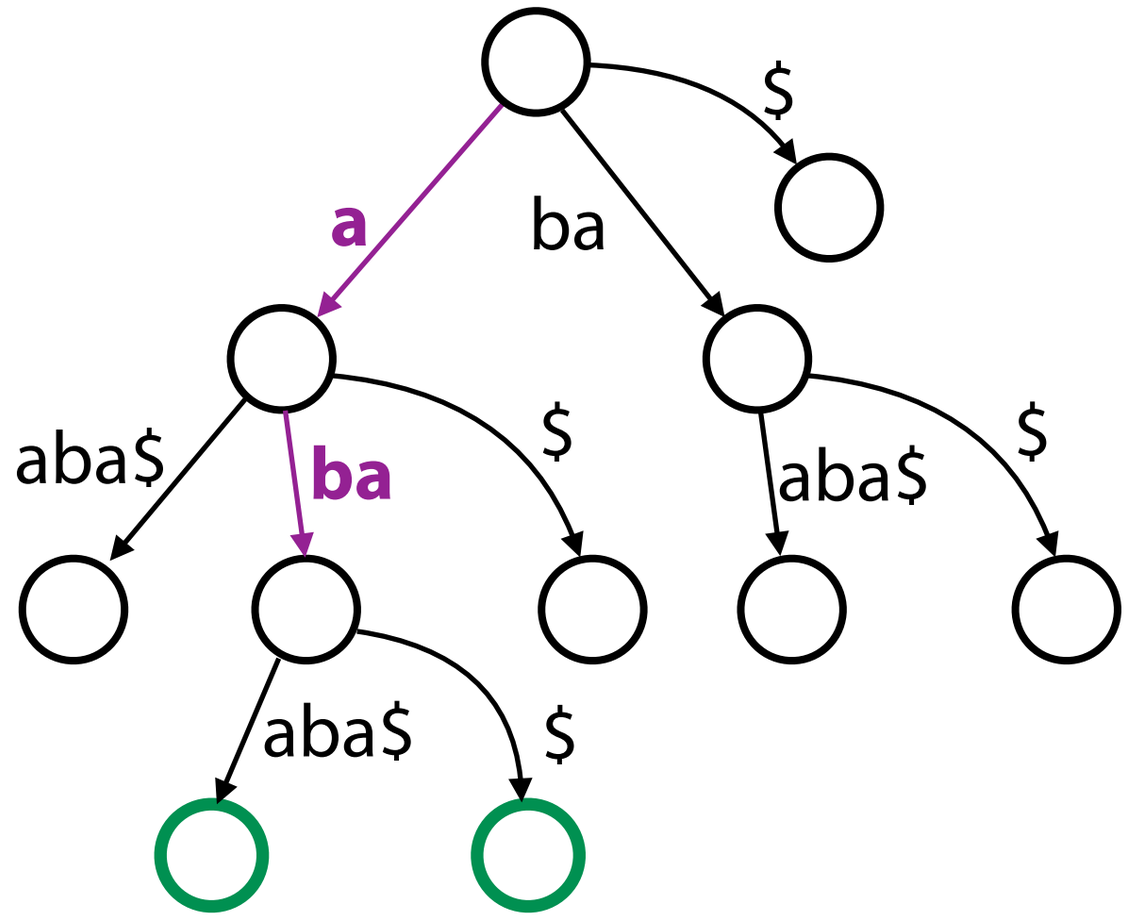
Two parts:

Walk down according to *S*	Count leaves below	**Overall**
O(*n*)	O(*k*)	

Suffix tree

Walk down according to *S*, then
count leaves below

How much work?

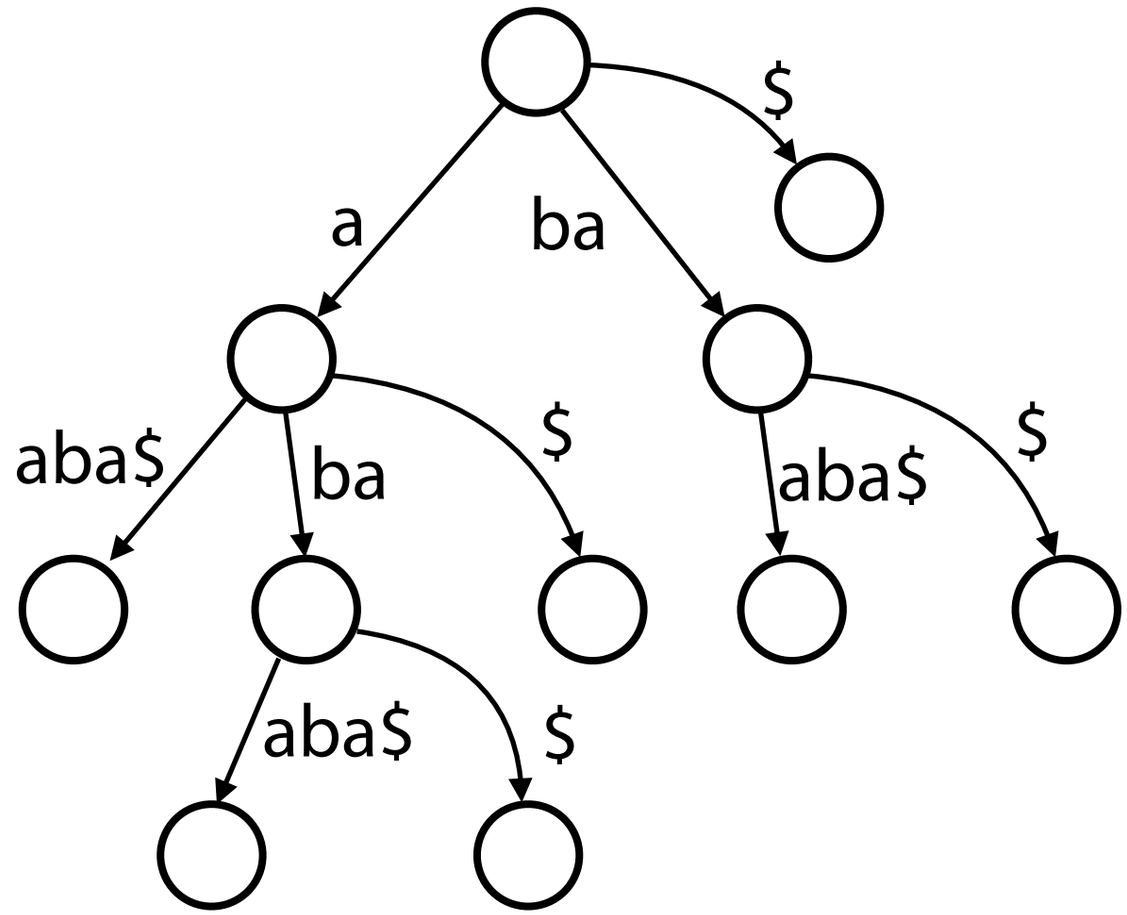


Two parts:

Walk down according to *S*	Count leaves below	**Overall**
O(*n*)	O(*k*)	O(*n* + *k*)

Suffix tree

How do we **report the offsets** where a string *S* occurs as a substring of *T*?

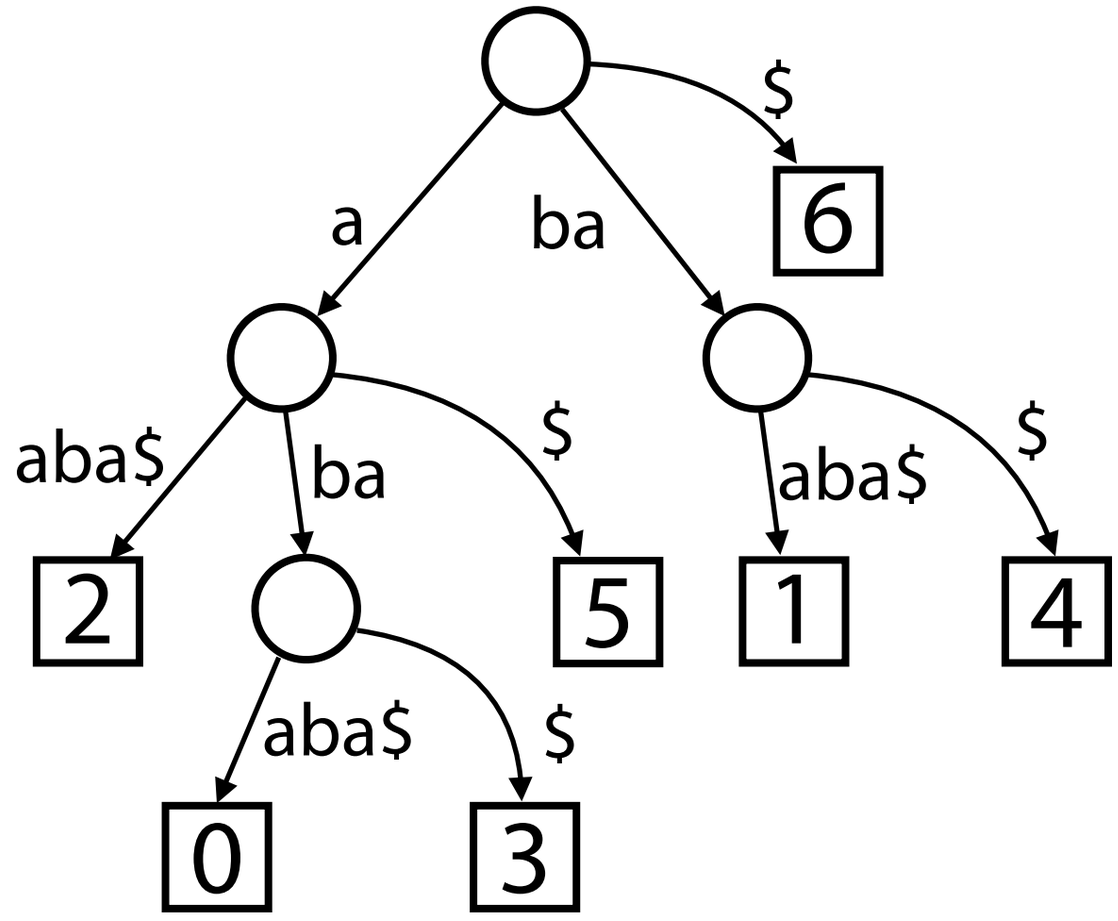


Suffix tree

How do we **report the offsets** where a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then report offsets in leaves below

aba

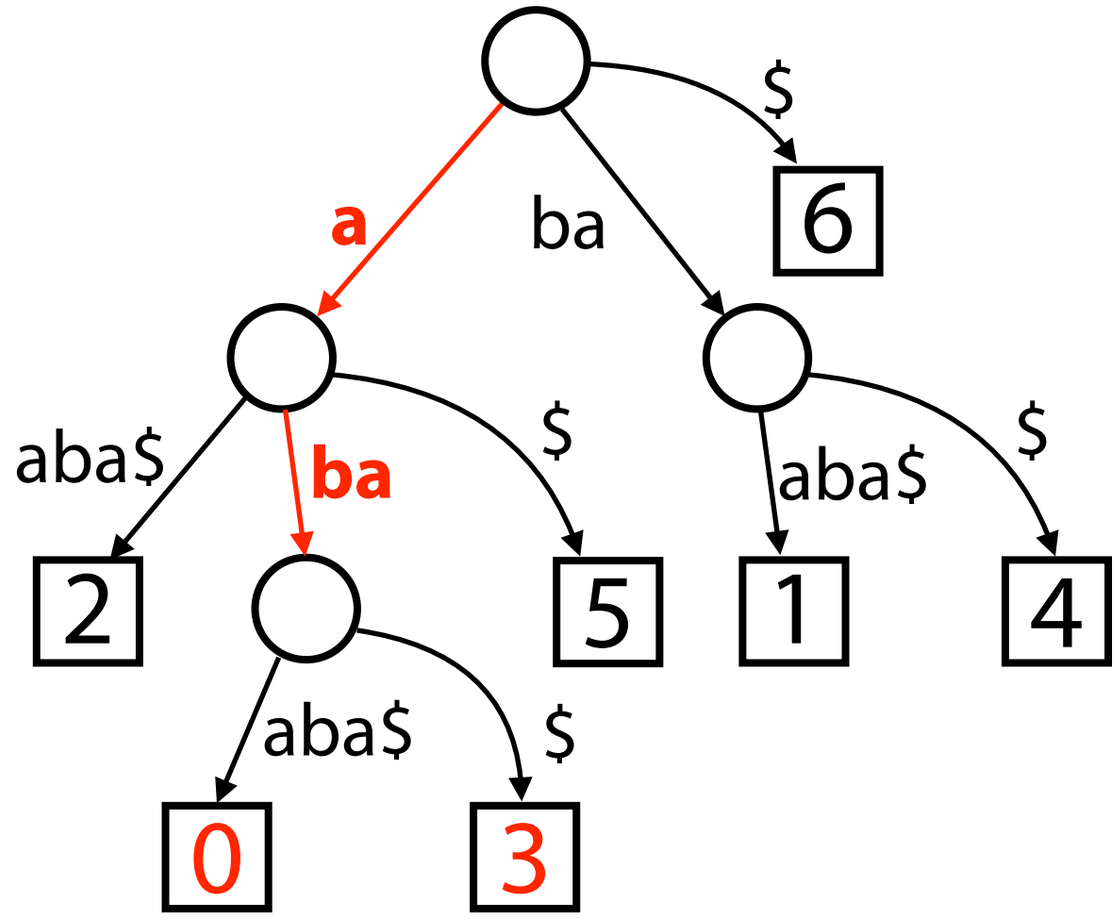


Suffix tree

How do we **report the offsets** where a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then report offsets in leaves below

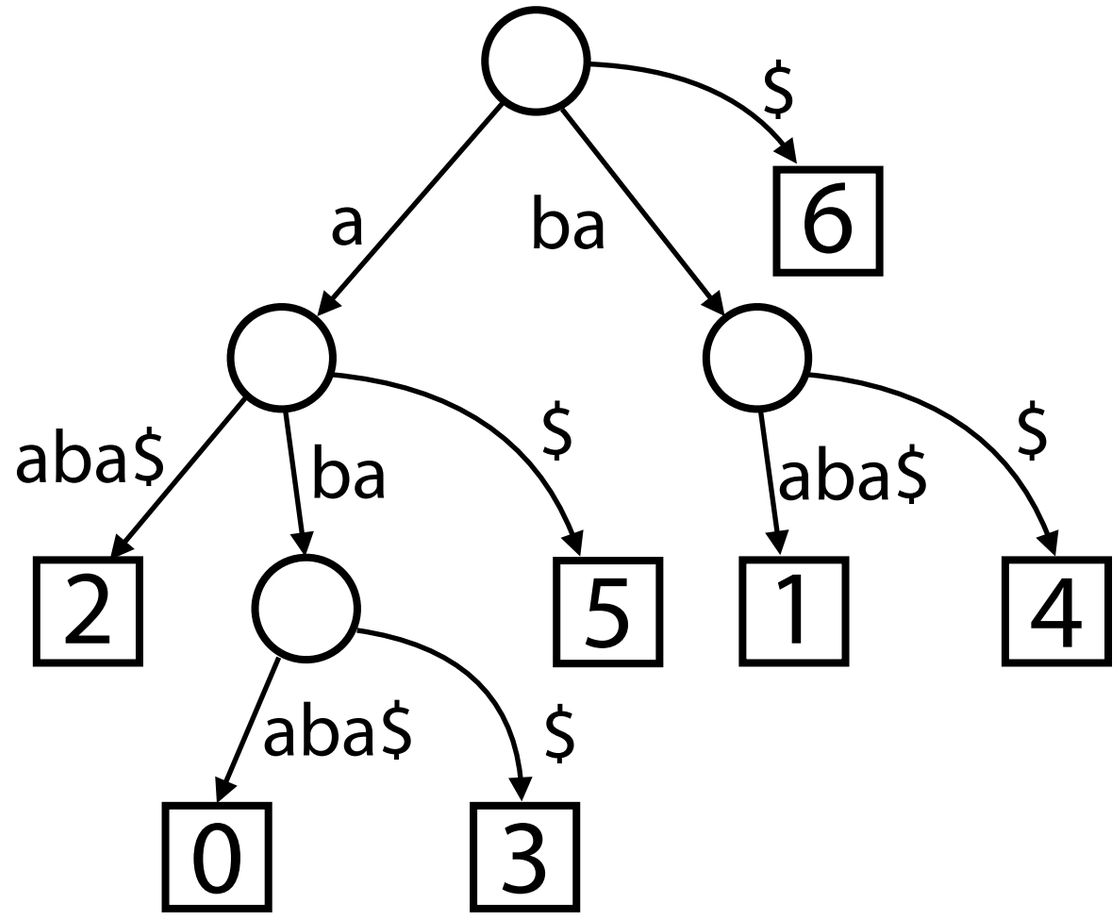
aba [0, 3]



Suffix tree

How do we **report the offsets** where a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then report offsets in leaves below



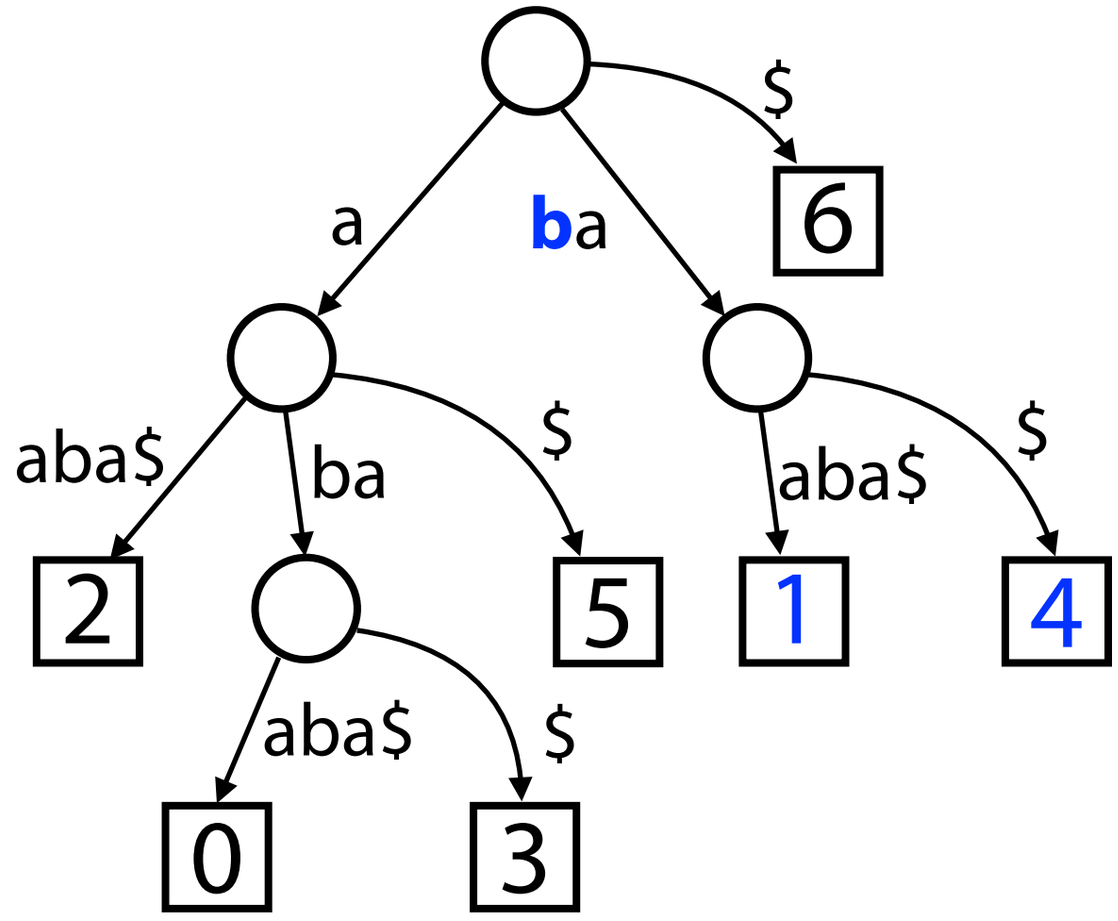
aba [0, 3]

b

Suffix tree

How do we **report the offsets** where a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then report offsets in leaves below



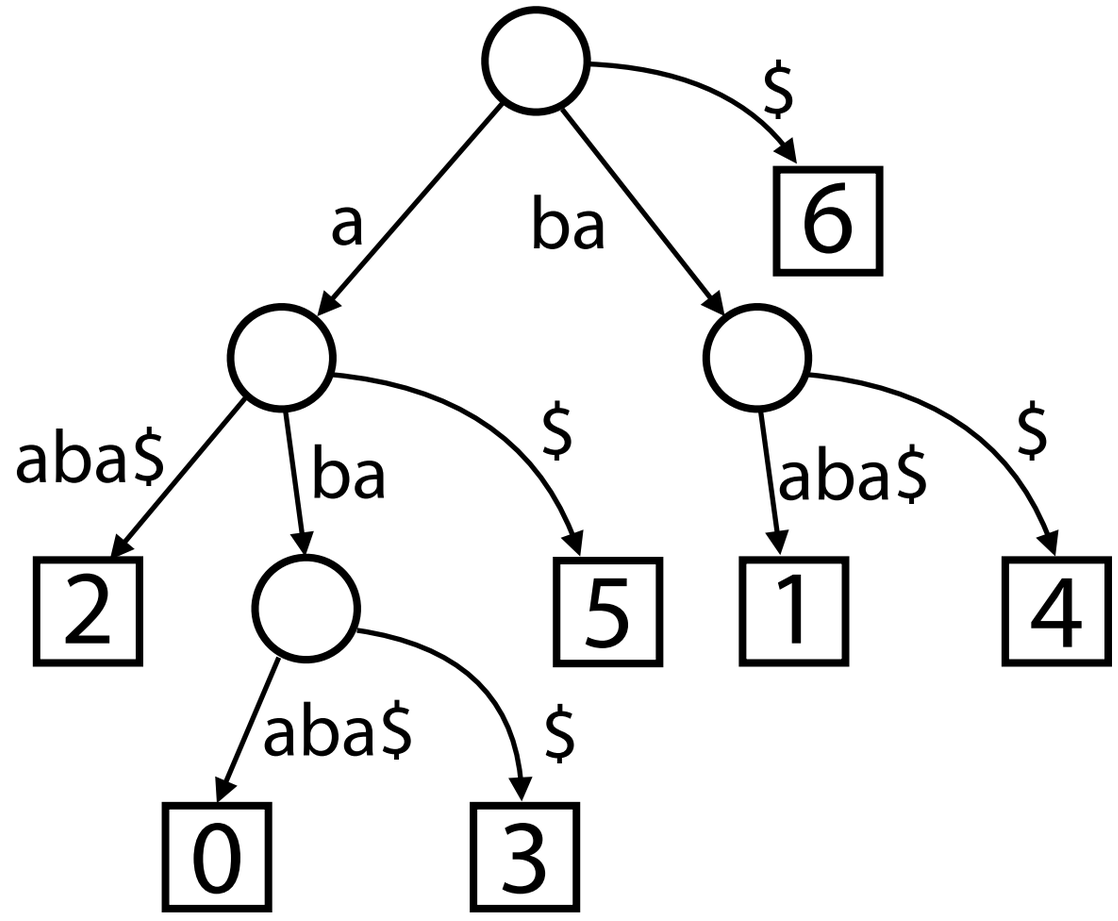
aba [0, 3]

b [1, 4]

Suffix tree

How do we **report the offsets** where a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then report offsets in leaves below



aba [0, 3]

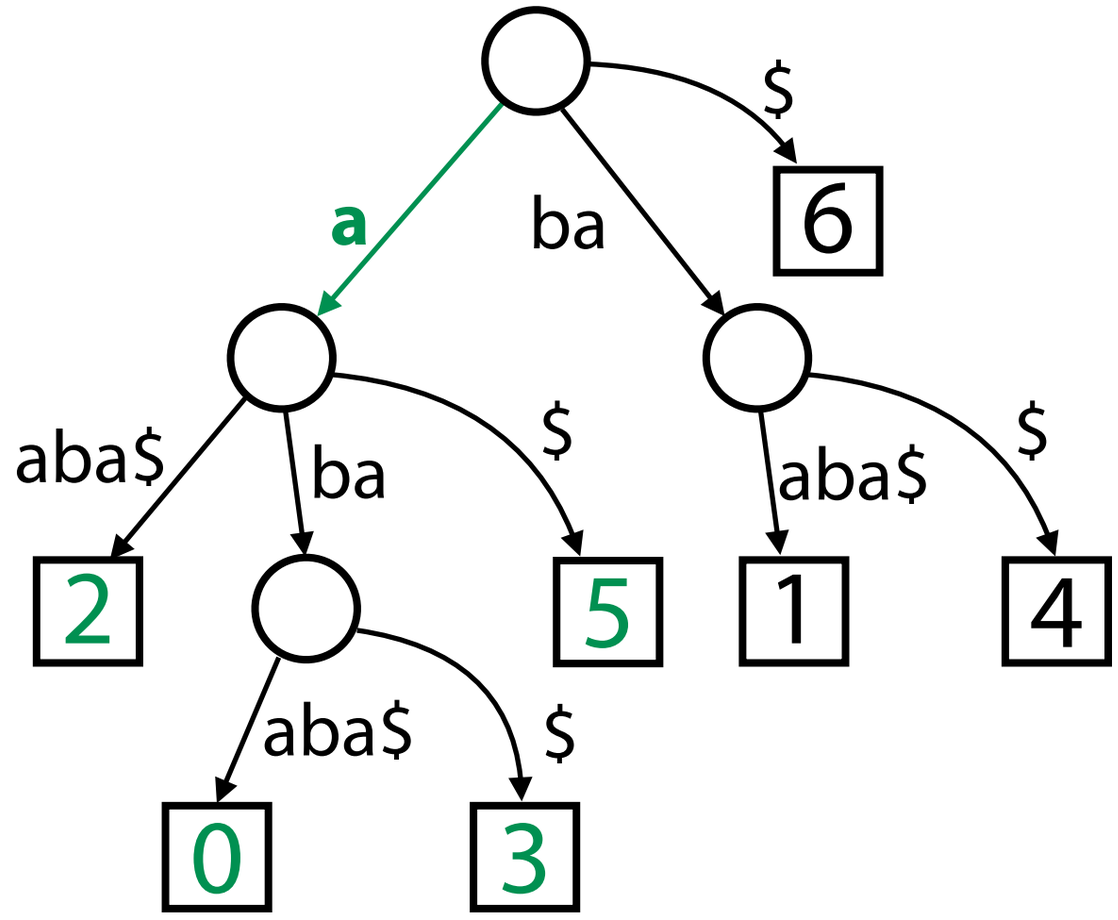
b [1, 4]

a

Suffix tree

How do we **report the offsets** where a string *S* occurs as a substring of *T*?

Same procedure as for suffix trie: walk down according to *S*, then report offsets in leaves below



aba [0, 3]

b [1, 4]

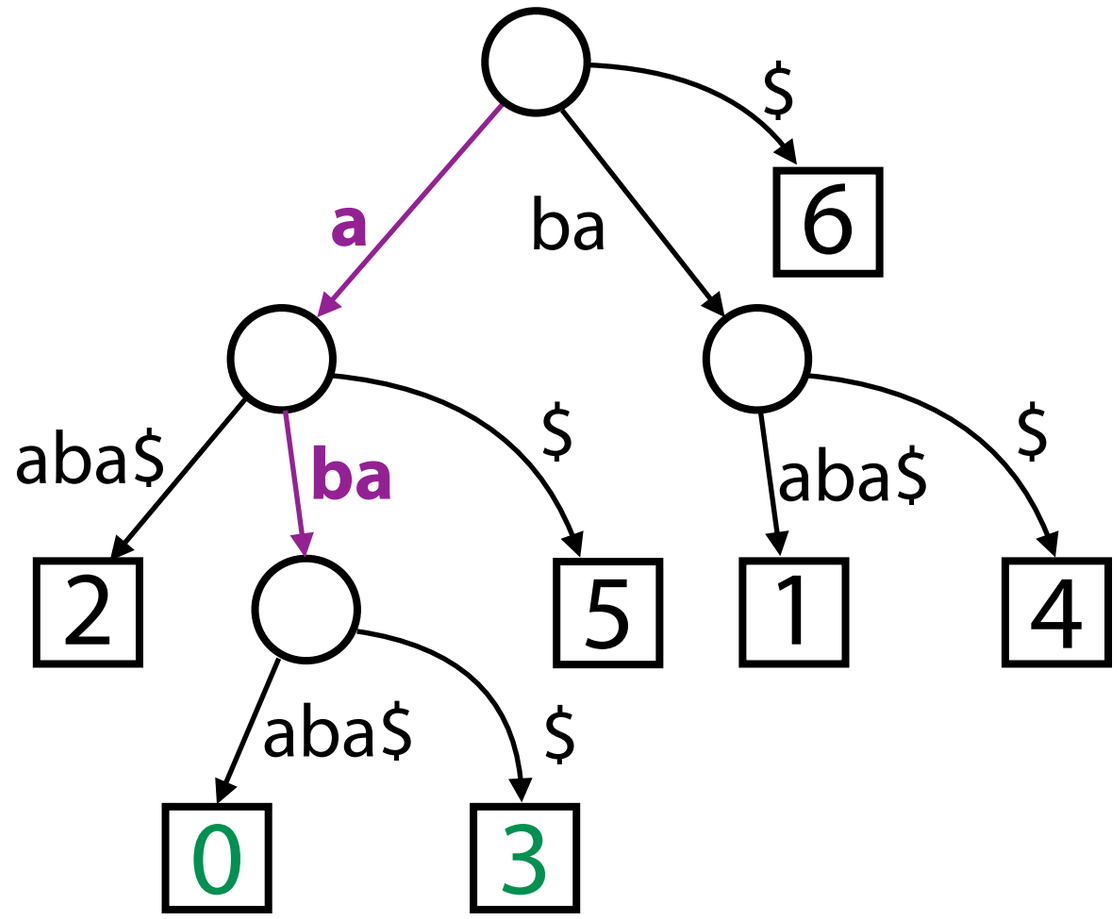
a [2, 0, 3, 5]

Suffix tree

Walk down according to *S*, then
report offsets in leaves below

How much work?

Same as counting!



Walk down according to *S*	Count leaves below	**Overall**
O(*n*)	O(*k*)	O(*n* + *k*)

Suffix tree bounds

Time: Does P occur?	$O(n)$
Time: Count k occurrences of P	$O(n + k)$
Time: Report k locations of P	$O(n + k)$ ← *Good!*
Space	$O(m)$

$m = |T|,\ n = |P|,\ k =$ # occurrences of P in T