The height of the Lyndon tree

Philippe Chassaing and Lucas Mercier

Université de Lorraine
IECL

AOFA 2013

- Alphabet:$\{0, 1\}$

- Alphabet: $\{0, 1\}$
- Suffixes: If $w = w_1 w_2 \ldots w_n$, then
  $s_k(w) = w_k w_{k+1} \ldots w_n$.
  $s_1(w) = w$.

- Alphabet: $\{0, 1\}$
- Suffixes: If $w = w_1 w_2 \dots w_n$, then
  $s_k(w) = w_k w_{k+1} \dots w_n$.
  $s_1(w) = w$.
- w is a Lyndon word if $\forall k \leq |w|$, $w \leq s_k(w)$ (for the lexicographic order)

- Alphabet: $\{0, 1\}$
- Suffixes: If $w = w_1 w_2 \dots w_n$, then
  $s_k(w) = w_k w_{k+1} \dots w_n$.
  $s_1(w) = w$.
- w is a Lyndon word if $\forall k \leq |w|$, $w \leq s_k(w)$ (for the lexicographic order)
- Standard right factor of w:
  smallest $s_k(w)$, $2 \leq k \leq n$.

Lyndon tree

Split at the standard right factor, and iterate.

000110010111001

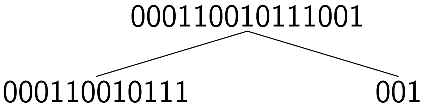
Lyndon tree

Split at the standard right factor, and iterate.

000110010111001

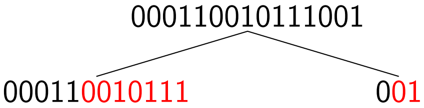
Lyndon tree

Split at the standard right factor, and iterate.



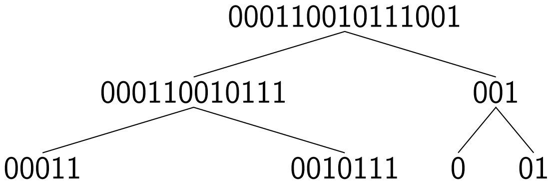
Lyndon tree

Split at the standard right factor, and iterate.



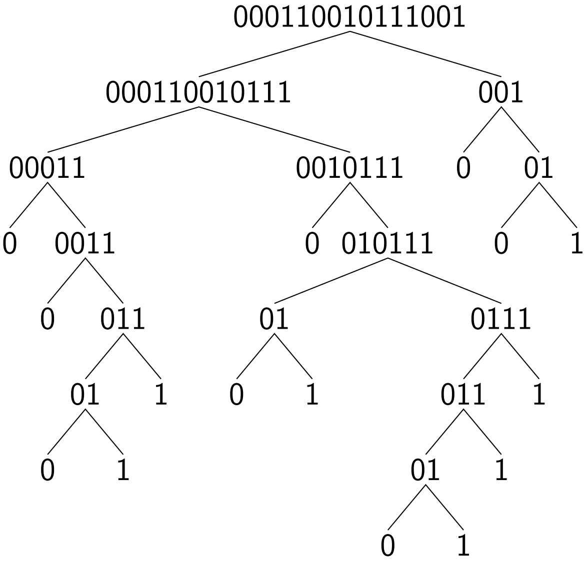
Lyndon tree

Split at the standard right factor, and iterate.



Lyndon tree

Split at the standard right factor, and iterate.



Let L_n be a Lyndon word chosen uniformly among the Lyndon words of length n, and let H_n be the height of the associated Lyndon tree.

Let L_n be a Lyndon word chosen uniformly among the Lyndon words of length n, and let H_n be the height of the associated Lyndon tree.

Theorem

$$\frac{H_n}{\ln n} \xrightarrow[n\to\infty]{p} \gamma$$

where $\gamma = \sup_\theta f(\theta) \simeq 5.092\ldots$

Let L_n be a Lyndon word chosen uniformly among the Lyndon words of length n, and let H_n be the height of the associated Lyndon tree.

Theorem

$$\frac{H_n}{\ln n} \xrightarrow[n\to\infty]{p} \gamma$$

where $\gamma = \sup_\theta f(\theta) \simeq 5.092\ldots$

No recursive structure.

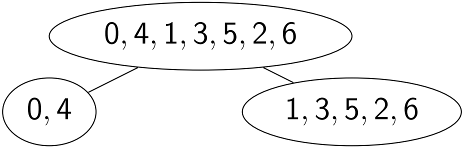
Binary search tree of a sequence

$$4, 1, 3, 5, 2, 6$$

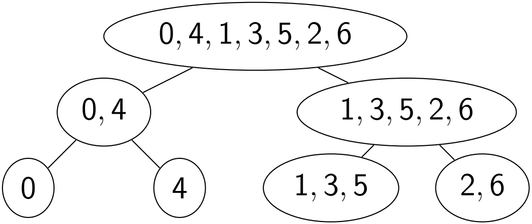
Binary search tree of a sequence

$0, 4, 1, 3, 5, 2, 6$

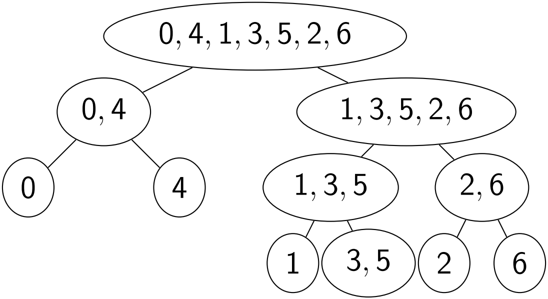
Binary search tree of a sequence



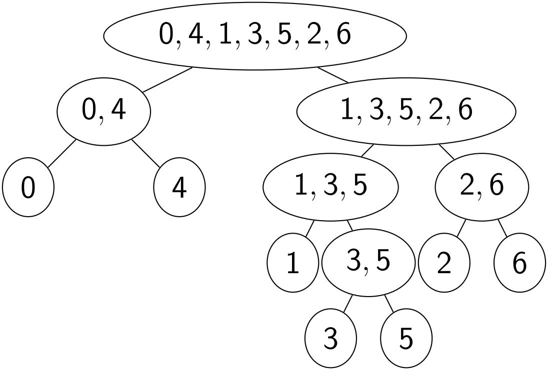
Binary search tree of a sequence



Binary search tree of a sequence



Binary search tree of a sequence



- Ordering the suffixes with the lexicographic order induces a permutation $\sigma(w)$ on $\{1, \ldots, n\}$.
  Example:$w = 0110$.
  $(s_1(w), s_2(w), s_3(w), s_4(w)) = (0110, 110, 10, 0)$
  gives $\sigma(w) = (2, 4, 3, 1)$.

- Ordering the suffixes with the lexicographic order induces a permutation $\sigma(w)$ on $\{1, \ldots, n\}$.
  Example: $w = 0110$.
  $(s_1(w), s_2(w), s_3(w), s_4(w)) = (0110, 110, 10, 0)$
  gives $\sigma(w) = (2, 4, 3, 1)$.
- A word w is a Lyndon word if $\sigma(w)_1 = 1$

- Ordering the suffixes with the lexicographic order induces a permutation $\sigma(w)$ on $\{1, \ldots, n\}$.
  Example:$w = 0110$.
  $(s_1(w), s_2(w), s_3(w), s_4(w)) = (0110, 110, 10, 0)$
  gives $\sigma(w) = (2, 4, 3, 1)$.
- A word w is a Lyndon word if $\sigma(w)_1 = 1$
- The binary search tree associated to $\sigma(w)$ is the Lyndon tree of w.

- Ordering the suffixes with the lexicographic order induces a permutation $\sigma(w)$ on $\{1, \dots, n\}$.
  Example:$w = 0110$.
  $(s_1(w), s_2(w), s_3(w), s_4(w)) = (0110, 110, 10, 0)$
  gives $\sigma(w) = (2, 4, 3, 1)$.
- A word w is a Lyndon word if $\sigma(w)_1 = 1$
- The binary search tree associated to $\sigma(w)$ is the Lyndon tree of w.
- But $\sigma(w)$ is not uniform. Suffixes are not independent.

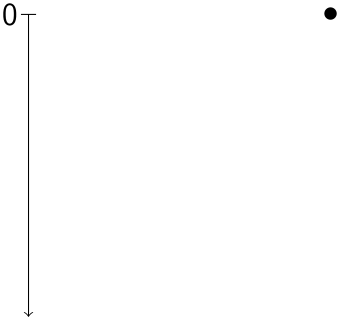
Suffixes far away are almost independent.

Suffixes far away are almost independent.
To find the smallest suffixes, we look for the longest runs of 0s. The position of the longest runs (and therefore smallest suffixes) is uniform. The top of the tree *looks like* BST_n

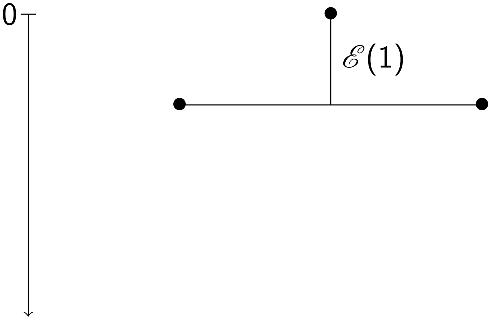
Suffixes far away are almost independent.
To find the smallest suffixes, we look for the longest runs of 0s. The position of the longest runs (and therefore smallest suffixes) is uniform. The top of the tree *looks like* BST_n
Echoes: if $s_k(w)$ starts with a l consecutive 0s, $s_{k+1}(w)$ starts with $l-1$ consecutive 0s.

Suffixes far away are almost independent.
To find the smallest suffixes, we look for the longest runs of 0s. The position of the longest runs (and therefore smallest suffixes) is uniform. The top of the tree *looks like* BST_n
Echoes: if $s_k(w)$ starts with a l consecutive 0s, $s_{k+1}(w)$ starts with $l-1$ consecutive 0s.
At the bottom of the tree, suffixes are no longer independent...

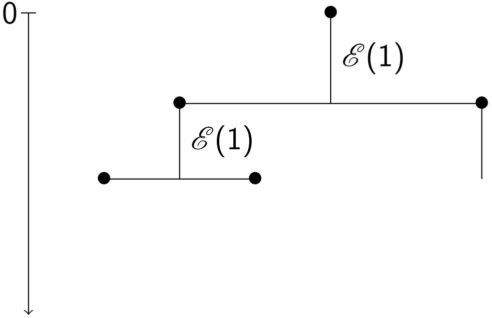
Pseudo-height of a Yule tree.



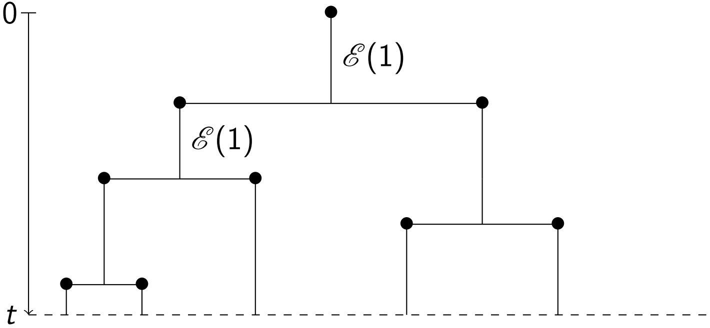
Pseudo-height of a Yule tree.



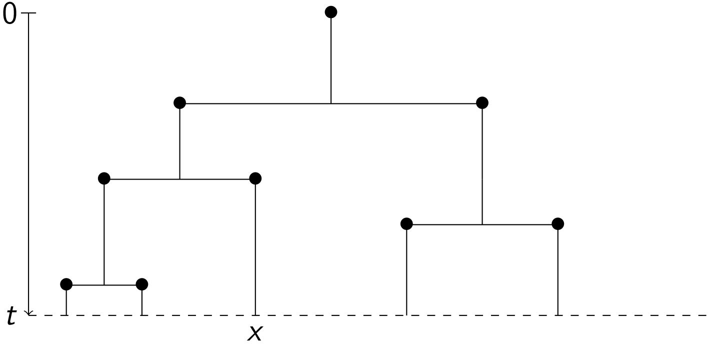
Pseudo-height of a Yule tree.



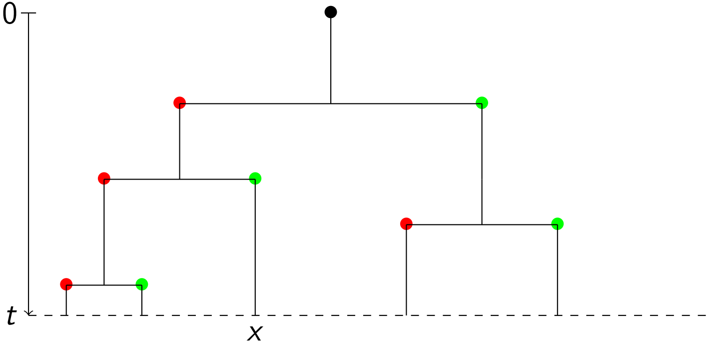
Pseudo-height of a Yule tree.



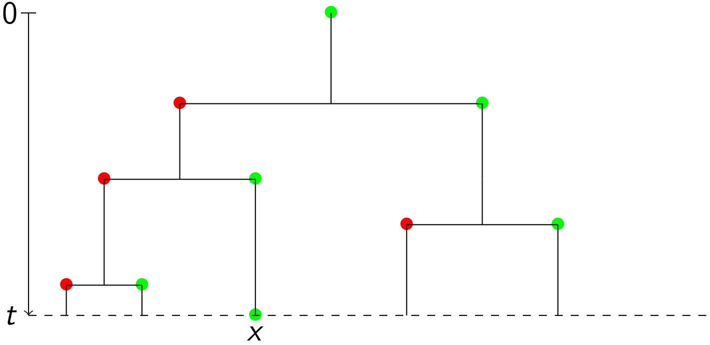
Pseudo-height of a Yule tree.



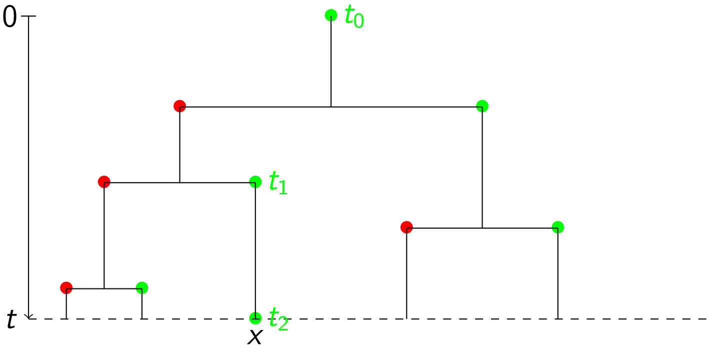
Pseudo-height of a Yule tree.



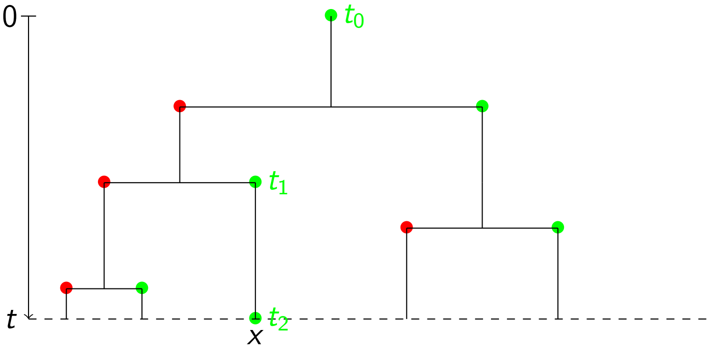
Pseudo-height of a Yule tree.



Pseudo-height of a Yule tree.



Pseudo-height of a Yule tree.



$$h_x = \begin{cases} \text{the depth of } x \text{ in the Yule tree} \\ +\sum_i \lfloor t_{i+1} - t_i \rfloor \\ +Geom(\frac{1}{2}) \end{cases}$$

Thanks for your attention!