Master Intership in Computer Science
NormaSTIC
GREYC – LITIS
Supervisors : Julien David and Thierry Lecroq

Lyndon Words and Cartesian Trees

Date: Spring 2024

Lyndon words are lexicographically smaller than all their non-empty proper suffixes. The Lyndon array of a word gives, for each position of the word, the length of the longest Lyndon word starting at that position. An example of Lyndon array is given in Figure 1.

The Cartesian tree [5] of a sequence s of m numbers is defined as follows:

- the root is the index i of the smallest element in s (the index of the first minimal element if it is not unique);
- the left subtree of the root is the Cartesian tree of $s[0 \mathinner{..} i-1]$;
- the right subtree of the root is the Cartesian tree of $s[i+1 \mathinner{..} m-1]$;

An example of Cartesien tree is given in Figure 2.

The exist several linear representations of Cartesian trees [2, 3, 4].

The Lyndon array of a word x can be computed with the Cartesian tree of the inverse suffix array of x [1].

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$x[i]$	a	b	b	a	b	a	b	a	a	b	a	b	b	a	b	a
$Lyn[i]$	3	1	1	2	1	2	1	8	5	1	3	1	1	2	1	1

Figure 1: Lyndon array of `abbababaababbaba`.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s[i]$	7	15	12	4	10	1	5	13	6	14	11	3	9	0	2	8

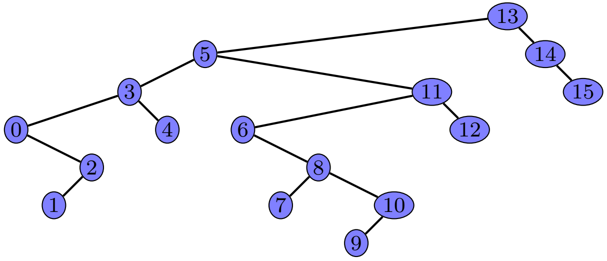

Figure 2: Cartesian tree of $s = 7, 15, 12, 4, 10, 1, 5, 13, 6, 14, 11, 3, 9, 0, 2, 8$.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	a	b	b	a	b	a	a	b	a	b	b	a	b	a	a	b
SA	13	5	14	11	3	6	8	0	15	12	4	10	2	7	9	1
ISA	7	15	12	4	10	1	5	13	6	14	11	3	9	0	2	8

Figure 3: Suffix array (SA) and inverse suffix array (ISA) of `abbababaababbaba`.

For instance, Figure 3 presents the suffix array and the inverse suffix array of `abbababaababbaba`. One can see that the inverse suffix array of `abbababaababbaba` is equal to sequence s of Figure 2. The Lyndon array of Figure 1 can then be easily computed with the help of the Cartesian tree of Figure 2.

The goals of this internship are:

- list all the known methods for computing Lyndon arrays;
- list all the known linear representations of Cartesian trees;
- select the linear representations than can enable to efficiently compute Lyndon arrays.

This internship fit into the activities of the Lyndex projet funded by the NormaSTIC CNRS FR 3638 federation. The results of this internship will feed the web site of the project `lyndex.org`.

Contacts:

`julien.david@unicaen.fr`

`Thierry.Lecroq@univ-rouen.fr`

References

[1] C. Hohlweg and C. Reutenauer. Lyndon words, permutations and trees. *Theoret. Comput. Sci.*, 307(1):173–178, 2003.

[2] E. Ohlebusch. *Bioinformatics Algorithms: Sequence Analysis, Genome Rearrangements, and Phylogenetic Reconstruction.* Oldenbusch Verlag, 2013.

[3] S. G. Park, A. Amir, G. M. Landau, and K. Park. Cartesian tree matching and indexing. *In 30th CPM*, 16:1–14, 2019.

[4] S. Song, G. Gu, C. Ryu, S. Faro, T. Lecroq, and K. Park. Fast algorithms for single and multiple pattern Cartesian tree matching. *Theoret. Comput. Sci.*, 849:47–63, 2021.

[5] J. Vuillemin. A unifying look at data structures. *Communications of the ACM*, 23(4):229–239, 1980.