International Workshop On
Combinatorial Algorithms
IWOCA

http://www.iwoca.org—Problems section.

Arithmetic Progressions of String Data Structures

Dominik Köppl
Kyushu University, Japan Society for the Promotion of Science
dominik.koeppl@inf.kyushu-u.ac.jp

presented at IWOCA 2019
23-07-2019

Checking the performance or integrity of string algorithms or data structures on word sequences can be of interest in testbeds if properties of the employed word sequences are well understood. For instance, one could check the validity of the computed Burrows-Wheeler transform (BWT) on a word of the Fibonacci sequence as the shape of its BWT is known [4, 2, 6].

Other studies based on the Fibonacci sequence are the suffix tree [5] or the Lempel-Ziv 77 (LZ77) factorization [1]. In [3], the suffix array and its inverse of each even word of the Fibonacci word sequence are studied as an arithmetic progression. In this study, the authors did not append the artificial `$` delimiter, allowing suffixes to be prefixes of other suffixes. This small fact makes the definition of $\mathrm{BWT}[i] = T[\mathsf{SA}[i] - 1]$ for a text T with suffix array SA incompatible with the traditional BWT defined on the lexicographic sorting of all rotations of the string T. Despite this incompatibility, in the suffix array based definition we can still observe a regularity for even Fibonacci words [3, Sect. 5]. The authors of [3, Remark 1] also showed that similar characteristics can be observed for other, more peculiar word sequences. However, it remains unknown whether we can formulate a class of word sequences, for which we can give the shape of the suffix array as an arithmetic progression (in the setting with or without the `$` delimiter). During the open problem session at IWOCA'19, the problem has been solved for binary alphabets thanks to joint efforts [7]. The problem remains still open for ternary or general alphabets. For tackling this problem, it could be helpful to use the software from `https://github.com/koeppl/strinalyze` written in C++ that can generate different kinds of word sequences and test string data structures for special shapes.

References

[1] J. Berstel and A. Savelli. Crochemore factorization of Sturmian and other infinite words. In *Proc. MFCS*, volume 4162 of *LNCS*, pages 157–166, 2006.

[2] M. Christodoulakis, C. S. Iliopoulos, and Y. J. P. Ardila. Simple algorithm for sorting the Fibonacci string rotations. In *Proc. SOFSEM*, volume 3831 of *LNCS*, pages 218–225, 2006.

[3] D. Köppl and T. I. Arithmetics on suffix arrays of Fibonacci words. In *Proc. WORDS*, volume 9304 of *LNCS*, pages 135–146, 2015.

[4] S. Mantaci, A. Restivo, and M. Sciortino. Burrows-Wheeler transform and Sturmian words. *Inf. Process. Lett.*, 86(5):241–246, 2003.

[5] W. Rytter. The structure of subword graphs and suffix trees of Fibonacci words. *Theor. Comput. Sci.*, 363(2):211–223, 2006.

[6] J. Simpson and S. J. Puglisi. Words with simple Burrows-Wheeler transforms. *Electr. J. Comb.*, 15(1), 2008.

[7] J. Daykin, M. Equi, G. Fici, V. Guerrini, D. Kübel and F. Stober. Characterization of binary strings whose suffix array is an arithmetic progression. *Open Problem Session IWOCA*, 2019.