

**Babenko, Maxim A.; Starikovskaya, Tatiana A.**

**Computing longest common substrings via suffix arrays.** (English) [Zbl 1142.68592](#)

Hirsch, Edward A. (ed.) et al., Computer science – theory and applications. Third international computer science symposium in Russia, CSR 2008 Moscow, Russia, June 7–12, 2008. Proceedings. Berlin: Springer (ISBN 978-3-540-79708-1/pbk). Lecture Notes in Computer Science 5010, 64-75 (2008).

Summary: Given a set of  $N$  strings  $A = \{\alpha_1, \dots, \alpha_n\}$  of total length  $n$  over alphabet  $\Sigma$  one may ask to find, for each  $2 \leq K \leq N$ , the longest substring  $\beta$  that appears in at least  $K$  strings in  $A$ . It is known that this problem can be solved in  $O(n)$  time with the help of suffix trees. However, the resulting algorithm is rather complicated (in particular, it involves answering certain least common ancestor queries in  $O(1)$  time). Also, its running time and memory consumption may depend on  $|\Sigma|$ .

This paper presents an alternative, remarkably simple approach to the above problem, which relies on the notion of suffix arrays. Once the suffix array of some auxiliary  $O(n)$ -length string is computed, one needs a simple  $O(n)$ -time postprocessing to find the requested longest substring. Since a number of efficient and simple linear-time algorithms for constructing suffix arrays has been recently developed (with constant not depending on  $|\Sigma|$ ), our approach seems to be quite practical.

For the entire collection see [\[Zbl 1136.68005\]](#).

**MSC:**

[68W05](#) Nonnumerical algorithms

Cited in **2** Documents

**Full Text:** [DOI](#)