Compressed Suffix trees: Theory and Implementation

Research seminar

Milan Milivojčević

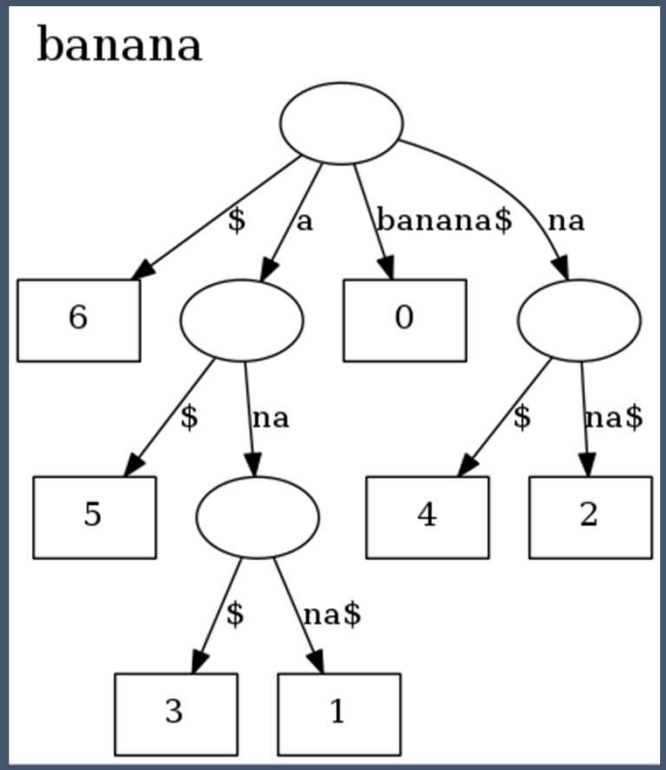
Mentor: Prof. Dr. Andrej Brodnik

What is Suffix tree?

• A suffix tree is a clever way to organize a word you can quickly find specific parts of it.

• It’s a tree structure where each branch represents a part of the word, and the complete words are found at the leaves.

• They are commonly used in areas like genomics, where large DNA sequences need to be analysed.



A suffix tree for a text supports the following operations:

1. root()
2. isleaf (v)
3. child(v, c)
4. sibling(v)
5. parent(v)
6. edge(v, d)
7. depth(v)
8. lca(v, w).
9. sl(v)

Motivation

Paper: Compressed Suffix Trees with Full Functionality

Author: Kunihiko Sadakane, Kyushu University

Proposed novel approach to construct compressed suffix trees that retain complete functionality while achieving significant space reduction.

Compressed Suffix Trees with Full Functionality

Kunihiko Sadakane
Department of Computer Science and Communication Engineering,
Kyushu University
Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581, Japan
sada@csce.kyushu-u.ac.jp

Abstract

We introduce new data structures for *compressed suffix trees* whose size are linear in the text size. The size is measured in *bits*; thus they occupy only $O(n \log |\mathcal{A}|)$ bits for a text of length n on an alphabet $\mathcal{A}$. This is a remarkable improvement on current suffix trees which require $O(n \log n)$ bits. Though some components of suffix trees have been compressed, there is no linear-size data structure for suffix trees with full functionality such as computing suffix links, string-depths and lowest common ancestors.

The data structure proposed in this paper is the first one that has linear size and supports all operations efficiently. Any algorithm running on a suffix tree can also be executed on our compressed suffix trees with a slight slowdown of a factor of polylog(n).

Plan

1. To implement both classical and compressed Suffix tree structure.

2. Test time and space complexities on ALGator system.

Data for testing: human genome or some lexicographical dictionary

Technology

- C++ programming language
- ALGator system

Reference

- Sadakane, K. (2007). Compressed suffix trees with full functionality. Theory of Computing Systems, 41(4), 589-607.

Thank you