58093 String Processing Algorithms (Autumn 2015)
Study Groups 1 (Thurday, November 5)

Group A: Burstsort

Members

- Denis, Devin
- Halin, Mikko
- He, Yan
- Jecmen, Jan
- Korento, Mika
- Mesimäki, Jerry
- Rantanen, Kari
- Riiheläinen, Anni
- Översti, Mikko

Advance reading. Read at least Section 1 of the following paper before the study group session.

> Ranjan Sinha, Anthony Wirth:
> **Engineering Burstsort: Toward fast in-place string sorting**.
> Journal of Experimental Algorithmics (JEA) 15 (2010): 2-5.
> `http://dx.doi.org/10.1145/1671970.1671978`

Topics for Discussion. Discuss at least the following topics in your group. Prepare to summarize the discussion for members of the other groups.

- The burst trie. How does it differ from the standard trie?

- The basic idea of burstsort. Why is it faster than MSD radix sort?

- The two improvements: sampling-based burstsort and copy-based burstsort.
  Why are they faster?

You may also discuss additional topics, for example:

- The main new ideas of the paper. How effective are they in practice?

Group B: Weak Prefix Search

Members

- Behalová, Karolína
- Falk, Sebastian
- Hurme, Teemu
- Karjalainen, Antti
- Korhonen, Tuukka
- Nikkari, Eeva
- Parisse, Alicia
- Toivanen, Aleksi
- Walve, Riku

Advance reading. Read at least Sections 1 and 2 of the following paper before the study group session.

Paolo Ferragina: **On the weak prefix-search problem**.
Theoretical Computer Science 483 (2013): 75-84.
`http://dx.doi.org/10.1016/j.tcs.2012.06.011`

Topics for discussion. Discuss at least the following topics in your group. Prepare to summarize the discussion for members of the other groups.

- The weak prefix search problem. How does it differ from the standard prefix search problem?
- The Patricia trie. How does it differ from the standard compact trie?
- How is Patricia trie used for solving the weak prefix search problem?

You may also discuss additional topics, for example:

- The basic ideas of the techniques in section 3.

Group C: Sparse Suffix Sorting

Members

- Bertron, Aurélien
- Concas, Francesco
- Hamberg, Jiri
- Hyvärinen, Ada
- Karvo, Tiina
- Korpinen, Kari
- Paavilainen, Topi
- Rajani, Chang
- Wilzbach, Sebastian

Advance reading. Read at least Sections 1–4 of the following paper before the study group session.

Tomohiro I, Juha Kärkkäinen, Dominik Kempa: **Faster Sparse Suffix Sorting**.
Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS),
pp. 386-396, 2014.
`http://dx.doi.org/10.4230/LIPIcs.STACS.2014.386`

Topics for discussion. Discuss at least the following topics in your group. Prepare to summarize the discussion for members of the other groups.

- The sparse suffix sorting problem. What is the problem of solving it using MSD radix sort?
- The ℓ-strict compact trie. How does it differ from the standard trie?
- The main ideas of the algorithm in Section 4.

You may also discuss additional topics, for example:

- The algorithm in Section 4 in more detail.

Group D: Comparison-driven Data structures for Strings

Members

- Abrar, Atthia
- Efremov, Rodion
- Haukka, Jani
- Holmes, Nicola
- Kaikkonen, Antti
- Niinimäki, Jouko
- Panchamukhi, Sandeep
- Räty, Olli
- Schettler, Jan

Advance reading. Read at least Section 2 of the following paper before the study group session.

> A. Amir, G. Franceschini, R. Grossi, T. Kopelowitz, M. Lewenstein, and N. Lewenstein:
> **Managing unbounded-length keys in comparison-driven data structures with applications to on-line indexing**.
> SIAM Journal on Computing 43 (2014): 1396-1416.
> `http://dx.doi.org/10.1137/110836377`

Topics for discussion. Discuss at least the following topics in your group. Prepare to summarize the discussion for members of the other groups.

- The DS_{lcp} list. How is it related to the lcp-comparison technique on the lectures?

- Consider lookup, insertion and deletion in a balanced binary tree with string keys. What is the difference with and without the DS_{lcp} list?

You may also discuss additional topics, for example:

- The implementation of DS_{lcp} list (Section 3).