Second meeting [UPDATED] (28-09-2022): The string indexing with compressed pattern problem

Mathias Søndergaard, s174426

September 28, 2022

Due to illness, the second meeting was postponed to 29-09-2022. This document describes the period between 20-09-2022 and today (28-09-2022). For details regarding the period before 20-09-2022, see previous meeting documents.

What has been done since last time

The plan was followed, which involved improving the prepossessing, specifically the consecutive suffix compression problem. The problem can be described as follows: Let S be a string of size n. What is a fast way to compress all suffixes of S? A naive solution would be to compress each suffix, without using any context of the previous suffixes. In [1], a reference is made to [2], in which a data structure for the generalized substring compression problem is described. However, this data structure is overly complicated, leaving it very hard to implement. Thus, I devised a new algorithm, "Lazy consecutive suffix compression". This algorithm uses simple and fast data structures, including: SA, LCP (done by RMQ on the LCP array) and a few dictionaries. Moreover, when compressing suffix S_i, the algorithm tries to reuse as much information from $LZ(S_{i+1})$. Before, I was able to handle strings of size 15.000, now I can handle strings of size 50.000 (and probably larger). The algorithm has a bad theoretical worst-case time, but is fast in practice. Finally, I have been writing a bit - mostly about this algorithm, but also preliminaries.

Plans for the next weeks

Write!

Questions for this meeting

No new question. If time allows we can talk about the Lazy consecutive suffix compression algorithm.

References

[1] Philip Bille, Inge Li Gørtz, and Teresa Anna Steiner. “String Indexing with Compressed Patterns”. In: (2020).

[2] Orgad Keller et al. “Generalized substring compression”. In: (2014).