1. Project in Text Indexing (WS 2015/16)

http://algo2.iti.kit.edu/2553.php

gog@kit.edu

**Aufgabe 1  (Better SA sampling)**

The sample suffix array $SA'$ as presented in the lecture takes $n/s \log n$ bits of space, where $s$ is the sampling parameter and $n$ the size of the text. Show how the space can be reduced to $n/s \log(n/s)$ while the access operation still remains in $O(s \cdot t_{LF})$.

**Aufgabe 2  (ISA sampling)**

The suffix array (SA) is a permutation of the numbers $0, \ldots, n-1$. The inverse suffix array (ISA) is the inverse permutation of SA. I.e. $SA[ISA[i]] = i$. Devise a sampling scheme similar for ISA which just stores $n/s$ ISA values and still supports access to every $ISA[i]$ value in $O(s \cdot t_{LF})$ worst case time. The scheme should use less space than the SA sampling scheme.

**Aufgabe 3  (Cyclic rotation)**

Write a program which uses an index structure to check if two strings of the same length $n$ are cyclic rotations of each other. E.g. ababba and abbaab are cyclic rotations or each other but not ababba and bbabaa. The index structure should be build for the first string and the algorithm should perform at most $n$ matching steps.