Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Dr. Simon Gog

4. Project in Text Indexing (WS 2016/17)

http://algo2.iti.kit.edu/3175.php
gog@kit.edu

Exercise 1 (*Substring select*)

Given a text T of length n over an alphabet of size σ and a query q of length m. A *substring select query* $select(i, q, T)$ returns the position of the i-th occurrence of q in T. Show how a substring select query can be answered in $O((m+1)\log^2 n)$ time using at most $n\lceil\log n\rceil + \sigma\lceil\log n\rceil + o(n\log n)$ bits of space.
Hint: Use the result of 2.4.

Exercise 2 (*Top-k range reporting*)

Given a set of n points $p_0, \ldots, p_{n-1}$ of the form $p_i = (i, Y[i])$ with $0 \le Y[i] < \log n$. We associate a weight w_i with each point p_i.
Devise a data structure which takes

(a) $O(n \log\log n)$

(b) $n\log\log n + 2n + o(n\log\log n)$

bits of space (on top of the space for the weights) and can answer top-k range reporting queries in $O((\log\log n)^2 + k\log\log n)$ time for (a) and $O(\log n + k\log n)$ time for (b).

Exercise 3 (*Longest palindromic substring*)

A string s is called palindromic if it reads the same forward or backward. E.g. *ada*, *gog*, or *hannah* are palindromic strings. Given a text T of size n over an alphabet of size σ. Design a linear time algorithm to calculate the longest palindromic substring of s. If there is more than one longest palindromic substring we are interested in the lexicographically smallest one.