

# Mandatory Exercise: Predecessors

Philip Bille

**1 The Subsequence Problem** A string  $P$  is a *subsequence* of string  $T$  if we can obtain  $P$  from  $T$  by removing 0 or more characters in  $T$ . For instance, `aba` is a subsequence of `bcadfbba`. Let  $T$  be a string of length  $n$  with characters from an alphabet of size  $\sigma$ . We are interested in efficient data structures for  $T$  that supports the following query:

- `subsequence( $P$ )`: return true if  $P$  is a subsequence of  $T$  and false otherwise.

Solve the following exercises.

- 1.1** Give a data structure that answers queries in  $O(|P|)$  time and uses little space. *Hint*: a good solution depends on both the size of the alphabet and the length of  $T$ .
- 1.2** Give a data structure that uses  $O(n)$  space and supports fast queries. The query time should depend on  $P$ .