Marburg University

Department of Computer Science
Prof. Dr. Sebastian Wild

Date: 2026-06-19
Version: 2026-06-20 15:36

Exercise Sheet 9 for Advanced Data Structures (Summer 2026)

Hand In: *Until 2026-06-26 18:00, on ILIAS.*

Problem 1

50 points

You and your $N-1$ friends are on a team building exercise, and are playing a game to test your teamworking and planning skills. The game is being held by a judge, J., who is not one of your friends. The game works as follows.

1. Before the game begins, you and your friends meet and plan. You start by giving J. a number T, which will come into play later.
2. J. separates all N of you, putting each of you in a different room. Each room is completely isolated, neither receiving nor sending any information.
3. J. sets up the judging room, which contains an infinite sequence of switches labelled $0, 1, \ldots$. Each switch is either *on* or *off*. All switches are initially *off*.
4. J. iterates through the list of N participants in some arbitrary order. On the turn of participant i, the following happen:
   a) J. takes i from their room to the judging room.
   b) i looks at T of the switches (and no others).
   c) i can change the state of the T switches they looked at arbitrarily (but not that of any others).
   d) i must declare whether or not they are the last participant in J.'s order.
5. If any participant declared incorrectly (i.e. the last participant did not say they were the last participant, or any other participant did), then you and all your friends lose. Otherwise, you win.

We are interested in the smallest T necessary to win (it is easy to check that, as T grows, the game gets easier).

a) Show that it is possible to win with $T = 10 + \log_2 N$.

b) Show that it is possible to win even if $T = 100 \log_2 \log_2 N$.

Problem 2

50 points

You must create a data structure which maintains a string S of length n with characters from the set Σ. The data structure must answer the following queries:

Update. Given $i \in \{1, \ldots, n\}$ and $c \in \Sigma$, set $S[i] = c$.

Query. Given $i, j \in \{1, \ldots, n\}$ with $i \leq j$ and $T \in \Sigma^*$, decide whether T appears as a (perhaps not contiguous) subsequence[1] within $S[i..j]$.

Your data structure should use $O(n)$ space, take $O(n)$ time to be constructed, and support update operations in $O(\log \log n)$ and query operations in $O(|T| \log \log n)$.

[1] Recall, a string $t_1, \ldots, t_k$ appears within $s_1, \ldots, s_n$ as a subsequence if and only if there exist integers $1 \leq i_1 < \cdots < i_k \leq n$ such that $t_j = s_{i_j}$.