

## Advanced Algorithms Weekplan 2

### Lecture

This lecture is about suffix trees which is the basic data structure for indexing strings. Additionally, we discuss some applications of suffix trees to various string problems.

### Curriculum for week 2

“Algorithms on Strings, Trees, and Sequences” (G) by Dan Gusfield.

- Chap. 5: All sections
- Chap. 6: Section 6.4.
- Chap. 7: All sections excluding 7.2.1, 7.3, 7.5, and 7.7.

### Exercises

Prepare solutions to the following exercises for the exercise session next week.

1. Suppose that we preprocess a string  $T[1..m]$  such that the following type of query can be handled efficiently:

**Count**( $P[1..m]$ ): Return the number of the occurrences of  $P$  in  $T$ .

Give a data structure that supports **Count**( $P[1..m]$ ) in  $O(m)$  time. *Hint*: use a suffix tree.

2. Searching a suffix tree with a pattern string  $P[1..m]$  uses time  $O(m + z)$ , where  $z$  is the number of occurrences of  $P$ . However, this only holds if we assume that the size of the alphabet is constant. Discuss ways to modify suffix trees to efficiently handle larger alphabets.
3. The generalized suffix tree for a set  $\{S_1, \dots, S_k\}$  is constructed by building a suffix tree for the string  $C = S_1\$1 \cdots S_k\$k$ . However, this also indexes substrings of  $C$  which are not substrings of any  $S_i$ . For instance, the string  $C$  will be indexed. Show how to remove all such substrings.
4. Consider a generalized suffix tree  $R$  for strings  $S_1$  and  $S_2$ . Give an algorithm to mark each node  $v$  in  $R$  with 1 (2) if there is a leaf in the subtree of  $v$  representing a suffix of 1 (2).
5. **Directed Acyclic Word Graphs** Let  $R$  be a suffix tree. Suppose that there are two nodes  $p$  and  $q$  such that the edge labeled subtrees below  $p$  and  $q$  are *isomorphic*. That is, for every path from  $p$  there is a path from  $q$  with the same path-labels and vice versa. Show how to use this property of  $p$  and  $q$  to save space.

### Mandatory assignment

The mandatory hand-in assignment consist of the following two exercises. They should be handed in no later than Sep. 19 at the start of the lecture. You are encouraged to discuss the exercises with your classmates but you have to hand in your solutions individually.

1. Show the suffix tree for the string `mississippi`.
2. **Subsequence Indexing** Show how to preprocess a string  $T[1..n]$  such that the following query can be answered efficiently:

**Subsequence**( $P[1..m]$ ): If  $P$  is a subsequence of  $T$  return yes. Otherwise return no.

For a definition of subsequence see weekplan 1. *Hint*: Build a DFA.