## Algorithms and Complexity 2016 Extra Mästarprov 1: Algoritms

This test is given to students who failed to get E on the ordinary Mästarprov 1. It consists of two problems. If both problems are solved correctly (basically) the test gives grade E. Your solutions should be handed in latest May 27th 16.00. No collaboration is allowed.

1. You are given an undirected graph G with n nodes. The graph is represented with adjacency lists. Is the graph a tree or not? You want to decide it algorithmically in time O(n).

Observe that this can easily be done in time O(|E|), but you are asked to do better than this. Also observe that you know n from start but you don't know |E| explicitly. (Can be computed though.)

Describe an O(n)-algorithm that solves the problem.

## $\mathbf{2}.$

We have a string  $s = s_1, s_2, ..., s_n$  of normal letters. Some substrings of s might form readable words. We want to use Dynamic Programming to compute the size of the maximal set of readable words in the string. Let M[k] be the size of the maximal set of readable words in  $s_1, s_2, ..., s_k$ . We assume that we can use a dictionary function read[w] which decides if w is a readable word in time O(1). (Read returns TRUE or FALSE.) Explain why the following recursion formula solves the problem:

$$\begin{cases} M[0] = 0\\ M[k] = max(M[k-1], max_i(M[i]+1))\\ \text{where } i \text{ runs over all indices } < k \text{ such that } read[s[i+1], ...s[k]] = TRUE \end{cases}$$

Construct an algorithm (using the recursion) that solves the problem.