

String Processing Algorithms 2015 - Week 2

Exercises

Rodion Efremov

October 30, 2015

Exercise 1

Outline algorithms that find the most frequent symbol in a give string.

- (a) for ordered alphabet, and
- (b) for integer alphabet.

The algorithms should be as fast as possible. What are their (worst case) time complexities? Consider also the case where $\sigma \gg n$.

Solution

Algorithm 1: MOSTFREQUENTSMBOL(S)

```
1 let  $f$  be an empty map  $f: \Sigma \rightarrow \mathbb{N}$ 
2  $\mu = \text{nil}$ 
3  $L_\mu = 0$ 
4 for  $i = 1$  to  $|S|$  do
5   if  $S[i]$  is not mapped in  $f$  then
6      $f(S[i]) = 1$ 
7     if  $L_\mu = 0$  then
8        $L_\mu = 1$ 
9        $\mu = S[i]$ 
10  else
11     $f(S[i]) = f(S[i]) + 1$ 
12    if  $L_\mu < f(S[i])$  then
13       $L_\mu = f(S[i])$ 
14       $\mu = S[i]$ 
15 return  $\mu$ 
```

Exercise 2

A full binary tree is a binary tree where every node is either a leaf or has two children. Show that every full binary tree with n leaves has exactly $2n - 1$ nodes.
Hint: Use induction.

Solution

Exercise 3