

# 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  $\mu$ 
```

---

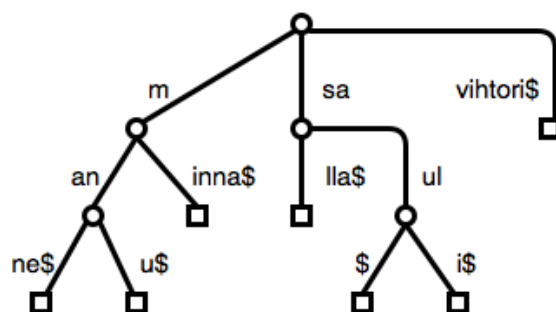
## Exercise 2

Let  $\mathcal{R} = \{\text{manne}, \text{manu}, \text{minna}, \text{salla}, \text{saul}, \text{sauli}, \text{vihtori}\}$ .

- (a) Give the compact trie of  $\mathcal{R}$ .
- (b) Give the balanced compact ternary trie of  $\mathcal{R}$ .

## Solution

(a)



(b)

See the drawing.

## Exercise 3

## Exercise 4

Prove

- (a) Lemma 1.14: For  $i \in [2..n]$ ,  $LCP_{\mathcal{R}}[i] = lcp(S_i, \{S_1, \dots, S_{i-1}\})$ .
- (b) Lemma 1.15:  $\Sigma LCP(\mathcal{R}) \leq \Sigma lcp(\mathcal{R}) \leq 2 \cdot \Sigma LCP(\mathcal{R})$ .

## Solution

(a)

We need an auxiliary lemma first:

**Lemma 1** (Distance lemma). *If  $S_1 < S_2 < S_3$ ,  $lcp(S_1, S_3) \leq lcp(S_2, S_3)$ .*

*Proof.* The proof is by contradiction: Let  $l_{13} = lcp(S_1, S_3)$  and  $l_{23} = lcp(S_2, S_3)$ . Assume the opposite that  $l_{13} > l_{23}$ . Now

$$\begin{aligned} S_1 &= a_1 a_2 \dots a_{l_{23}} \dots a_{l_{13}} b_1 b_2 \dots, \\ S_2 &= a_1 a_2 \dots a_{l_{23}} c_1 c_2 \dots, \\ S_3 &= a_1 a_2 \dots a_{l_{23}} \dots a_{l_{13}} d_1 d_2 \dots \end{aligned}$$

Since  $c_1 > a_{l_{23}+1}$ , we must have also  $S_2 > S_3$ , which is a contradiction.  $\square$

**Example:**

$$\begin{aligned} S_1 &:aaaab \\ S_2 &:aaaba \\ S_3 &:aaabc \end{aligned}$$

Now assume that  $i \in [2..n]$ . We have that

$$lcp(S_i, \{S_1, \dots, S_{i-1}\}) = \max\{lcp(S_i, S_{i-1}), lcp(S_i, \{S_1, \dots, S_{i-2}\})\}.$$

Because by distance lemma for any  $j = 1, 2, \dots, i-2$ ,  $lcp(S_j, S_i)$  cannot exceed  $lcp(S_i, S_{i-1})$ , we must have that

$$lcp(S_i, \{S_1, \dots, S_{i-1}\}) = lcp(S_i, S_{i-1}) = LCP_{\mathcal{R}}[i],$$

as expected.