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: MostFrequentSymbol(S)

```
1  let f be an empty map f: Σ → ℕ
2  μ = nil
3  L_μ = 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_μ = 0 then
8              L_μ = 1
9              μ = S[i]
10     else
11         f(S[i]) = f(S[i]) + 1
12         if L_μ < f(S[i]) then
13             L_μ = f(S[i])
14             μ = S[i]
15 return μ
```

Exercise 2

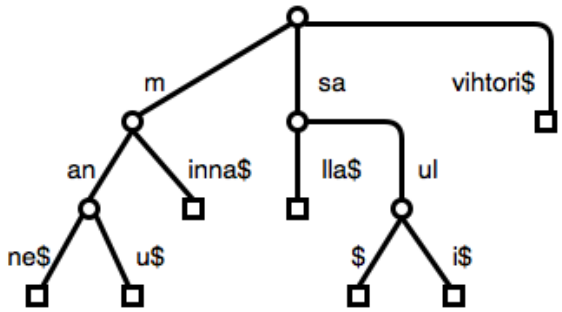
Let $\mathcal{R} = \{\texttt{manne}, \texttt{manu}, \texttt{minna}, \texttt{salla}, \texttt{saul}, \texttt{sauli}, \texttt{vihtori}\}$.

(a) Give the compact trie of $\mathcal{R}$.

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

Solution

(a)



Exercise 3