

Biosequence Algorithms, Spring 2005  
Exercise 2, February 8, 2005, at 12.15–14 in MT2

1. Simulate Boyer-Moore matching to locate occurrences of pattern “maamamma” in text “jo hommaamme maamamman”
  - (a) applying the bad character shift rule alone
  - (b) applying the good suffix shift rule alone
  - (c) selecting the maximum shift given by the bad character rule and the good suffix rule.

In each case, indicate the shifts and the character comparisons performed.

2. (Gusfield, Ex. 2.9) Explain why Theorem 2.2.4 holds, and apply it to design a linear-time algorithm to accumulate the  $l(i)$  values (or  $l'(i)$  using the notation of the textbook) in linear time. (Assume that the  $N_j(P)$  values have been computed.)
3. The *extended* version of the Boyer-Moore bad character rule is as follows: When a mismatch occurs between characters  $P[i]$  and  $T[h] = x$ , shift  $P$  to the right so that the closest occurrence of  $x$  to the left of  $i$  in  $P$  gets aligned with  $T[h]$  (Gusfield, page 18). The extended bad character rule can be implemented using either  $\Theta(|\Sigma|n)$  or  $\Theta(n)$  auxiliary space. Describe the structures and algorithms for preprocessing and looking up the shift values, and discuss trade-offs of the alternative implementations.
4. Draw the Aho-Corasick automaton for the set of patterns

$$\mathcal{P} = \{AT, AC, CAT, CATGAT, GACTAC\} ,$$

indicating values of the *goto*, *fail* and *output* links.

5. Describe a simple record structure for implementing nodes of an Aho-Corasick automaton. How much space would a single node require? (The standard implementation of a pointer takes 4 bytes.) How much space would be needed for the Aho-Corasick automaton built from all DNA patterns of length 10?