Algorithms I

Tutorial 9

November 15, 2016

Problem 1

(CLRS 32.4-5) Give a linear-time algorithm to determine if a text T is a cyclic rotation of another string T. For example, *arc* and *car* are cyclic rotations of each other.

Solution: Search T' in TT (i.e. T concatenated with itself).

Problem 2

How can you use the prefix function to find occurrences of T in S.

Solution: Let F be the prefix function of $S' = T + \$ + S$ where \$ does not occur in S or T. Let us number the indices in S' as $0, 1, 2, \ldots m + n$, where $m = |T|$ and $n = |S|$. For all $2m \leq i \leq n + m$, if $F[i] = m$, then there is a match of T starting at index $i - m$ in S.

Problem 3

Let T be a string of length m. Propose an $O(m)$-time algorithm to determine whether T can be represented as $T = \alpha\beta = \beta\alpha$ for two non-empty strings α and β.

Solution: Search for T in TT using the KMP string-matching algorithm. The first and the last positions are trivial matching positions. If there is any non-trivial matching position, we have a representation of T as in the problem.

Problem 4

Suppose that all characters in the pattern P are different. Show how to accelerate $NAIVE - STRING\text{-}MATCHER$ to run in time $O(n)$ on an n-character text T. asdf

Solution: Iterate over P and T simultaneously. Let i be the position in P and j in T. Whenever, $P_i = T_j$, we increment both i and j. And if $P_i \neq T_j$, there are two cases:

- if $i = 0$, increment j (Positions are 0-indexed)
- otherwise, set $i = 0$.

If at any iteration during the algorithm, $j = |P|$, we have found a match.

Problem 5

(CLRS 32.2-1) Working modulo $q = 11$, how many spurious hits does the Rabin-Karp matcher encounter in the text $T = 3141592653589793$ when looking for the pattern $P = 26$?

Solution: Three spurious hits, $15 \equiv 59 \equiv 92 \equiv 26 \equiv 4 \mod 11$