

Algorithmen und Programmierung III

Abgabe 12.12.2014, 12 Uhr

Aufgabe 1

5 Punkte

Geben Sie einen effizienten Algorithmus an, der für einen Text T und ein Muster P entscheidet ob P eine *Teilfolge* (nicht unbedingt ein zusammenhängendes Teilwort) der Zeichenfolge T ist. Analysieren Sie, wie immer, die Laufzeit Ihres Algorithmus.

Aufgabe 2

9 Punkte

Implementieren Sie in Java folgende Algorithmen zum Stringmatching, wobei für ein Muster alle Stellen (Zeichennr.) im Text ausgegeben werden sollen, wo es auftritt:

- (a) Brute-force
- (b) Rabin-Karp
- (c) Boyer-Moore (ohne Good-Suffix-Heuristik)

Vergleichen Sie die Algorithmen experimentell mit verschiedenen Alphabetgrößen einschließlich $\Sigma = \{0, 1\}$ und beschreiben Sie Ihre Beobachtungen. Suchen Sie auf jeden Fall auch nach “Moby” und “captain” im text unter

<http://www.gutenberg.org/files/2701/2701.txt>

Aufgabe 3

6 Punkte

Bei der zweidimensionalen Mustererkennung sind ein $n \times n$ -Feld T und ein $m \times m$ -Feld P von Zeichen eines Alphabets Σ gegeben und es sollen alle Vorkommen von P in T gefunden werden.

- (a) Beschreiben und analysieren Sie einen Brute-Force-Algorithmus.
- (b) Wie würden Sie den Rabin-Karp-Algorithmus modifizieren, um dieses Problem zu lösen? Analysieren Sie die Laufzeit im schlechtesten Fall.