7. Aufgabenblatt zur Vorlesung WS 2012/13

Helmut Alt

Algorithmen und Programmierung III

Abgabe 14.12.2012, 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.

Aufgabe 2 *8 Punkte*

(a) Analysieren Sie die erwartete Laufzeit des Brute-Force-Algorithmus für String-Matching unter der Annahme, dass die Alphabetgröße $d \geq 2$ ist und dass jeder Buchstaben an jeder Stelle des Textes mit gleicher Wahrscheinlichkeit vorkommt.

Hinweis: Zeigen Sie zunächst, dass man bei jedem erneuten Anlegen des Musters an den Text nach erwartet konstant vielen Vergleichen ein Mismatch erhält, vgl. auch Analyse von Skiplists.

(b) Implementieren Sie den Brute-Force-Algorithmus für String-Matching, wobei für ein Muster alle Stellen (Zeichennr.) im Text ausgegeben werden sollen, wo es auftritt. Experimentieren Sie mit hinreichend großen realen Texten, um empirisch die Abhängigkeit der Laufzeit von der Mustergröße m und der Textgröße n festzustellen. Wird die Analyse aus (a) bestätigt (obwohl der Text nicht rein zufällig ist)?

Verwenden Sie z.B. den Text unter

`http://www.gutenberg.org/files/2701/2701.txt`.

Aufgabe 3 *7 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.