

Datenkompression

Sommersemester 2014

Übungsblatt 4

Prof. Dr. E. Ohlebusch

Institut für Theoretische Informatik

S. Arnold

Ausgegeben am 24.06.2014

Besprechung in der Übung am 03.07.2014

Aufgabe 4.1

Führen Sie den Induced-Sorting-Algorithmus für die Eingabe $S = \text{dcdbdcdadcdcbdc} \$$ aus, mit der üblichen Sortierung $\$ < \mathbf{a} < \mathbf{b} < \mathbf{c} < \mathbf{d}$. Welche Rekursionstiefe ergibt sich? Gibt es eine Eingabe S' , die nicht länger ist als S , für die der Induced-Sorting-Algorithmus aber mehr rekursive Aufrufe benötigt als für S ?

Aufgabe 4.2

Um die LZ77-Faktorisierung eines Textes der Länge n in linearer Zeit zu berechnen, können die Hilfsarrays PSV_{lex} und NSV_{lex} verwendet werden. In der Vorlesung wurde ein Algorithmus vorgestellt, der diese Hilfsarrays bestimmt.

- Geben Sie einen alternativen Algorithmus zur Berechnung von PSV_{lex} und NSV_{lex} an, der einen Stack benutzt. Ihr Algorithmus sollte lineare Laufzeit besitzen und darf auch auf das Suffix-Array SA zugreifen.
- Beweisen Sie die Korrektheit Ihres Algorithmus!
- Zeigen Sie, dass die Laufzeit Ihres Algorithmus tatsächlich in $\mathcal{O}(n)$ liegt!

Die folgenden Aufgaben beziehen sich auf den Skriptteil „Lempel-Ziv Factorization: LZ77 without Window“, der auf der Vorlesungshomepage und im Skriptdrucksystem verfügbar ist.

Aufgabe 4.3

Zeigen Sie, dass die Laufzeit von Algorithmus 5 (Berechnung der LZ-Faktorisierung) in $\mathcal{O}(n)$ liegt. Es wurde bereits gezeigt, dass die Arrays SA , ISA , PSV_{lex} und NSV_{lex} in linearer Zeit berechnet werden können.

Aufgabe 4.4

Führen Sie Algorithmus 6 (Berechnung von PSV_{text} und NSV_{text}) für das Suffix-Array in Abbildung 10 aus. Dabei sollte insbesondere ersichtlich werden, in welcher Reihenfolge die Array-Einträge in Abbildung 11 generiert werden. Welche Vorteile besitzt die Verwendung von PSV_{text} und NSV_{text} gegenüber PSV_{lex} und NSV_{lex} ?