

Algorithmen zur Sequenzanalyse: Übungsblatt 5

Enno Ohlebusch
Timo Beller

Das Übungsblatt wird am 07.01.2014 besprochen.

Aufgabe 1

Sei S ein String der Länge n ohne Abschlusszeichen $\$$ und L die letzte Spalte der Matrix, die alle n Rotationen von S in lexikografischer Reihenfolge enthält. Sei SA das Suffix Array von S und der String $BWT[1..n]$ definiert durch:

$$BWT[i] = \begin{cases} S[SA[i] - 1] & \text{falls } SA[i] > 1 \\ S[n] & \text{falls } SA[i] = 1 \end{cases}$$

Finden Sie einen String S für den gilt $L \neq BWT$.

Aufgabe 2

Geben Sie einen Algorithmus an, der das ψ -Array berechnet. (Hinweis: Sie können den Algorithmus 14 modifizieren).

Aufgabe 3

Geben Sie einen Algorithmus an, der aus der BWT und dem ψ -Array den ursprünglichen String S berechnet. Sie dürfen dabei annehmen, dass die Position des $\$$ -Zeichens in der Variablen *index* steht. Modifizieren Sie Ihren Algorithmus so, dass zusätzlich das Suffix Array des Strings berechnet wird. Ist es möglich, das ψ -Array mit dem Suffix Array zu überschreiben um Speicherplatz zu sparen?

Aufgabe 4

Implementieren Sie die Move-To-Front-Transformation. Wenden Sie diese auf die Strings $S = \text{diesisteintestdiesisteintest\$}$ und $BWT = \text{tt\$titiitddeessiieeeeeiisssnn}$ an. Vergleichen Sie die mittlere Huffman-Codewortlänge der beiden Move-To-Front-Transformierten. Testen Sie, ob sich in der Praxis ein Array oder eine Liste besser zur Verwaltung der Buchstabenliste eignet.

Aufgabe 5

Algorithmus 1 zeigt eine alternative Transformation. Welche Laufzeit hat diese? Implementieren Sie Algorithmus 1 und vergleichen Sie ihn mit der originalen Move-To-Front-Transformation: Welcher Algorithmus ist schneller? Welche Ausgabe kann der Huffman-Code besser komprimieren?

Algorithm 1 Alternative zur Move-To-Front-Transformation von L .

Initialize an array ord containing for each characters from Σ its order in Σ

$R[i] \leftarrow ord[L[1]]$

for $i \leftarrow 2$ **to** n **do**

if $ord[L[i - 1]] > ord[L[i]]$ **then**

$R[i] \leftarrow ord[L[i]] + \sigma - ord[L[i - 1]]$

else

$R[i] \leftarrow ord[L[i]] - ord[L[i - 1]]$
