

Algorithmen zur Sequenzanalyse

Wintersemester 2017/2018
Besprechung am 10.11.2017

Übungsblatt 2

Prof. Dr. E. Ohlebusch,
Institut für Theoretische Informatik

Aufgabe 2.1.

In Tabelle 1 ist das Suffix Array von $S = \text{annaassannasananas\$}$ gegeben. Berechnen Sie mit Hilfe des Kasai-Algorithmus das LCP-Array von S .

i	SA	$S_{SA[i]}$
1	19	\$
2	4	aassannasananas\$
3	13	anas\$
4	15	anas\$
5	1	annaassannasananas\$
6	8	annasananas\$
7	17	as\$
8	11	asananas\$
9	5	assannasananas\$
10	3	naassannasananas\$
11	14	nanas\$
12	16	nas\$
13	10	nasanas\$
14	2	mnaassannasananas\$
15	9	mnanasananas\$
16	18	s\$
17	12	sananas\$
18	7	sannasananas\$
19	6	ssannasananas\$

Tabelle 1: Das Suffix Array des Strings $S = \text{annaassannasananas\$}$.

Aufgabe 2.2.

Der Φ -Algorithmus berechnet aus einem String und dessen Suffix Array das LCP-Array. Im Gegensatz zum Kasai-Algorithmus wird jedoch das so genannte Φ -Array statt dem inversen Suffix Array benutzt. In Algorithmus 1 ist der Beginn des Φ -Algorithmus dargestellt. Vervollständigen Sie den Algorithmus, sodass das LCP-Array berechnet wird. Wenden Sie diesen Algorithmus auf das Suffixarray aus der vorherigen Aufgabe an.

Algorithmus 1 Der Φ -Algorithmus bekommt einen String S der Länge n und dessen Suffix Array SA übergeben. Achtung: Im Algorithmus wird von 0 an gezählt.

```
1: function  $\Phi$ -ALGORITHMUS( $S, SA$ )
2:    $\Phi[0] \leftarrow SA[n - 1]$ 
3:   for  $i \leftarrow 1$  to  $n - 1$  do
4:      $\Phi[SA[i]] \leftarrow SA[i - 1]$ 
5:    $\ell \leftarrow 0$ 
6:   for  $i \leftarrow 0$  to  $n - 1$  do
7:      $j \leftarrow \Phi[i]$ 
8:     while  $S[j + \ell] = S[i + \ell]$  do
9:        $\ell \leftarrow \ell + 1$ 
10:     $\Phi[i] \leftarrow \ell$ 
11:     $\ell \leftarrow \max(0, \ell - 1)$ 
12:    // ToDo
```

Aufgabe 2.3.

Geben Sie einen Algorithmus in Pseudocode an, der bei Eingabe zweier Strings S^1 und S^2 einen längsten gemeinsamen Teilstring von S^1 und S^2 berechnet und ausgibt. Der Algorithmus soll die worst-case Laufzeit von $O(n_1 + n_2)$ haben, wobei n_1 und n_2 die Längen von S^1 und S^2 sind.

Aufgabe 2.4.

Gegeben sei ein (langer) String S sowie dessen Suffixarray. Geben Sie einen Algorithmus in Pseudocode an, der bei Eingabe eines (kurzen) Strings P feststellt, ob P ein Teilstring von S ist. Wenn ja, so soll der Algorithmus alle Positionen ausgeben, an denen P in S vorkommt bzw. beginnt. Analysieren Sie die worst-case Laufzeit.