Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr
Jan Griebsch, Stefan Pfingstl

SS 2005
Aufgabenblatt 8
6. Juni 2005

Praktikum Diskrete Optimierung

(Abgabetermin: **Montag, den 13.6.2005,** $14.^{00}$Uhr)

Suchen in Texten

Aufgabe 1 Suffix-Bäume

Schreiben Sie ein Programm, das für einen Eingabetext X der Länge n mit Ukkonen's Algorithmus in linearer Zeit $O(n)$ einen Suffix-Baum konstruiert und damit für Muster Y der Länge m in Zeit $O(m)$ prüft, ob das Muster im Text vorkommt. Falls das Muster vorkommt, soll die Anzahl der Vorkommen sowie die Position des ersten und des letzten Vorkommens ausgegeben werden. (Beachte, dass es durchaus $O(n)$ Vorkommen eines Muster geben kann und man deshalb für die Ausgabe nicht den ganzen Teilbaum durchsuchen darf. Man wird hier eine Vorverarbeitung durchführen müssen.)

Ihr Programm muß ohne Zuhilfenahme der LEDA-Bibliothek implementiert werden. Die Laufzeit für die Konstruktion des Suffix-Baums sowie die Laufzeit für die Bearbeitung jedes einzelnen Musters soll ausgegeben werden. Prüfen Sie anhand der ausgegebenen Zeiten, ob die Laufzeit Ihrer Implementierung tatsächlich linear mit der Textlänge bzw. mit der Musterlänge wächst.

Der Eingabetext besteht aus beliebigen ASCII-Zeichen (Typ `char`), die Alphabet-Größe ist also 256. Das Zeichen `$` ist das Schlußzeichen und kommt ausschließlich am Ende des Textes vor. Die Beispieleingaben stehen in den Text-Dateien `text1` bis `text6` (sowie die zugehörigen korrekten Ausgaben in `text1.sol` bis `text6.sol`) im WWW zur Verfügung. Jede solche Datei enthält zuerst einen Text, abgeschlossen mit `$`. Darauf folgt ein Zeilenvorschub und dann eine Reihe von Mustern, die jeweils von einem `$` und einem Zeilenvorschub gefolgt sind. Ihr Programm bekommt den Namen der Eingabedatei als Kommandozeilen-Parameter, die Ausgabe soll auf den Bildschirm erfolgen.

Hinweise

Beachten Sie das Beispiel für eine korrekte Ausgabe des Algorithmus auf der Rückseite dieses Aufgaben-Blattes.

Beispiel für Eingabe und Ausgabe des Algorithmus

Eingabe:

```
Heute programmieren wir
einen Algorithmus zur
Suche in Texten.$
te$
wir
ein$
xx$
```

Ausgabe:

```
Suffix-Baum: Zeit 0.5 Sek
Muster 1: 2 Vorkommen, erste Pos. 4, letzte Pos. 59, Zeit: 0.1 Sek
Muster 2: 1 Vorkommen, Pos. 21, Zeit: 0.2 Sek
Muster 3: 0 Vorkommen, Zeit: 0.1 Sek
```