
Praktikum Diskrete Optimierung

Letzter Abgabetermin: Montag, den 1. Juli 2013, 14:00 Uhr

Aufgabe 1 (Suffix-Arrays)

Schreiben Sie ein Programm, das für einen Eingabetext X der Länge n direkt einen Suffix-Array in linear-logarithmischer Zeit $O(n \log n)$ konstruiert und dabei möglichst wenig Speicher verbraucht. Schreiben Sie weiterhin den Such-Algorithmus, um damit für Muster Y der Länge m in Zeit $O(m \log n)$ zu prüfen, ob das Muster im Text vorkommt. Falls das Muster vorkommt, soll die Anzahl der Vorkommen sowie die Position des lexikographisch kleinsten und des lexikographisch größten Vorkommens ausgegeben werden. Damit sind die Positionen gemeint, deren Suffix lexikographisch am kleinsten bzw. am größten ist. Dies entspricht den Einträgen, die man mit der binären Suche im Array **Pos** findet.

Ihr Programm muss ohne Zuhilfenahme der LEDA-Bibliothek implementiert werden. Die Laufzeit für die Konstruktion des Suffix-Arrays 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 `$` ist das Schlusszeichen und kommt ausschließlich am Ende des Textes vor. Die Beispielergebnisse stehen in den Text-Dateien `text1.txt` bis `text6.txt` bzw. in den Muster-Dateien `text1.pat` bis `text6.pat` (sowie die zugehörigen korrekten Ausgaben in `text1.sol` bis `text6.sol`) auf der Übungsseite zur Verfügung. Jede Text-datei enthält einen Text, abgeschlossen mit `$`. Die Muster-Dateien enthalten eine Reihe von Mustern, die jeweils von einem `$` und einem Zeilenvorschub gefolgt sind. Beachten Sie, dass die Muster ebenfalls Zeilenumbrüche enthalten können. Ihr Programm bekommt den Namen der Eingabedatei und der Muster-Datei 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-Array: Zeit 0.000238 Sek  
Muster 1: 2 Vorkommen  
lex. kleinste Pos. 4, lex. groesste Pos. 59  
Zeit: 0.000012 Sek  
Muster 2: 1 Vorkommen  
lex. kleinste Pos. 21, lex. groesste Pos. 21  
Zeit: 0.000006 Sek  
Muster 3: 0 Vorkommen  
Zeit: 0.000002 Sek
```