

Frank Hoffmann
Klaus Kriegel
Romain Grunert
Ludmila Scharf
André Schulz

Ws 2007/08

30. November 2007
Abgabe: 10. Dezember 2007
vor der Vorlesung

7. Übung zur Vorlesung Höhere Algorithmik

1. Aufgabe (5 Punkte).

Sei L eine Liste von n ganzen Zahlen, die nicht notwendigerweise paarweise verschieden sein müssen.

- (a) Bestimmen Sie mit Dynamischer Programmierung eine streng monoton steigende Teilfolge \mathcal{I} maximaler Länge $l = l(L)$. Was ist die Laufzeit?
- (b) Sei U eine Partition der Listeneinträge in schwach monoton fallende Teilsequenzen. (*Achtung:* Mehrfacheinträge kommen auch so oft vor.) Angenommen, es gibt eine streng monoton wachsende Teilfolge I , die genauso lang ist wie die Anzahl der Folgen in so einem U . **Zeigen Sie:** Dann ist I schon streng monoton steigend mit maximaler Länge und U ist minimale Partition.
- (c) Zeigen Sie, wie man eine solche minimale Partition “greedy” aufbauen kann, wenn man sequentiell die Einträge aus L ausliest. Finden Sie dann eine längste aufsteigende Teilsequenz. Ihr Algorithmus sollte Laufzeit $O(n \log n)$ haben.

2. Aufgabe (5 Punkte).

Setzen Sie im folgenden String an die durch `_` gekennzeichneten Stellen die Anfangsbuchstaben Ihrer Nachnamen als Kleinbuchstaben ein. (Wer die Übungszettel allein bearbeitet, verwendet die Anfangsbuchstaben seines Vor- und Nachnamens.)

`an_an_as`

Konstruieren Sie mit dem Linearzeitalgorithmus aus der Vorlesung den Suffixbaum für den sich ergebenden String und erläutern Sie die Konstruktion. (Es genügt *nicht*, nur das Ergebnis anzugeben.)

3. Aufgabe (5 Punkte).

Betrachten Sie das folgende Problem: Gegeben sind ein String S und eine natürliche Zahl k . Gesucht ist ein kürzester Teilstring, der genau k -mal in S vorkommt. Beschreiben Sie, wie man dieses Problem mit Hilfe des Suffixbaums in $O(|S|)$ lösen kann. Begründen Sie Korrektheit und Laufzeitschranke Ihrer Lösung.

4. Aufgabe (5 Punkte).

In einem String S sind als Teilstrings alle Palindrome maximaler ungerader Länge gesucht. Betrachten Sie dazu den gemeinsamen Suffixbaum von S und dem rückwärts gelesenen String S^{rev} . Wie können aus dem Suffixbaum Palindrome ungerader Länge

abgelesen werden? Entwickeln Sie daraus einen Algorithmus, der in $O(|S|)$ alle Palindrome maximaler ungerader Länge in S findet. Argumentieren Sie, warum Ihr Algorithmus die gesuchte Lösung findet und die geforderte Laufzeit einhält.

Hinweis 07.12. (F. Hoffmann): Die Aufgabe ist m.E. unverhältnismäßig schwer. Der Lösungsweg, der mir vorschwebte, funktioniert so leider nicht. Sie können deshalb – falls notwendig – für die Lösung folgendes Hilfsmittel benutzen. Man kann in Linearzeit einen Suffixbaum so vorverarbeiten, dass man danach für jedes Anfragepaar b_1, b_2 von Blättern den tiefsten gemeinsamen Vorfahren $lca(b_1, b_2)$ im Baum in konstanter Zeit bestimmen kann. Ein Palindrom soll in der Ausgabe durch Index des ersten bzw. letzten Zeichens angegeben werden. Maximales Palindrom heißt, dass wenn man links und rechts davon das nächste Zeichen hinzunimmt kein Palindrom mehr erhält.