Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Dr. Michal Mnuk

SS 1998
Übungsblatt 7
8. Juli 1998

Grundlegende Algorithmen

Abgabe bis 15. Juni 1998 in der Übung

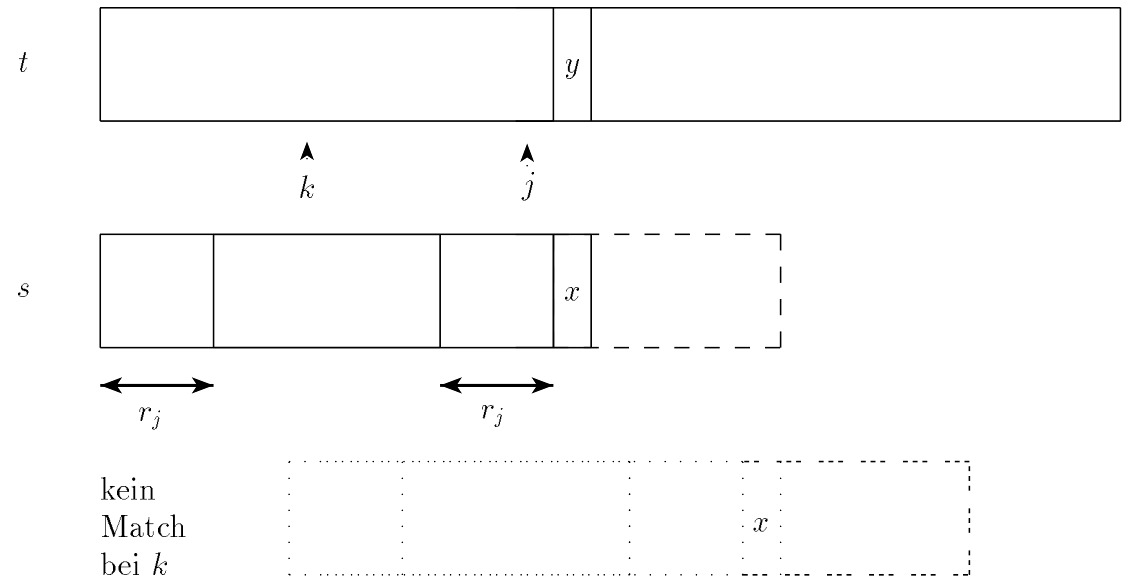
Aufgabe 20

Es geht hier um die entscheidende Einsicht, warum der in der Vorlesung vorgestellte KMP–Algorithmus das Auftreten eines Suchstrings s der Länge m in einem Text t der Länge n korrekt erkennt.

Nehmen Sie an, daß das das Pre-Processing bereits erledigt ist, daß man also für jedes $j = 1, \dots, m$ bereits den maximalen nichttrivialen Rand der Länge r_j kennt. Der Algorithmus legt zunächst den Suchstring linksbündig unter den Text, und vergleicht die Buchstaben von links her. Sobald er einen Mismatch an der Stelle $j + 1 < m$ erkennt, schneidet er von links her $j - r_j$ Buchstaben ab, und das Spiel beginnt von vorne. Eine äquivalente Sichtweise wäre, daß der das „Suchfenster" um den entsprechenden Betrag nach rechts verschiebt. Dies wird solange wiederholt, bis er entweder einen Match findet, oder aber zum Ende angelangt ist. Offensichtlich terminiert der Algorithmus, die Frage ist nur, ob er nicht eventuell einen Match „übersieht", wenn er das „Suchfenster" um mehr als einen Buchstaben nach rechts schiebt.

Beweisen Sie deshalb (durch Widerspruch):

Es gelte $s(i) = t(i)$ für $i = 1, \dots, j$ und $s(j+1) = x \neq y = t(j+1)$ für $j + 1$. Dann gibt es keinen Match links von der Position $j - r_j + 1$, d.h. der an die Stelle $k < j - r_j + 1$ verschobene Suchstring s kann nicht völlig mit dem darüberliegenden Ausschnitt des Textes t übereinstimmen (vgl. Skizze).



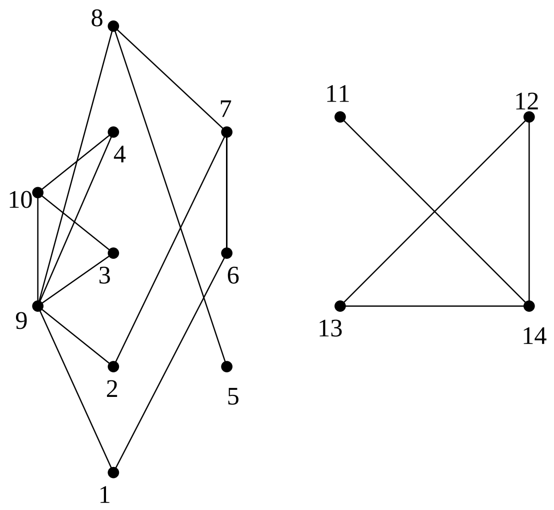
Aufgabe 21

a) Modifizieren Sie den Algorithmus **DFS** so, daß er, gestartet an der Wurzel eines binären Baumes, folgende Numerierungsstrategie verwendet:
Zunächst wird rekursiv der linke Unterbaum numeriert, dann bekommt die Wurzel die nächste verfügbare Nummer und schließlich wird der rechte Unterbaum rekursiv numeriert. Diese Numerierung wird als *Inorder*-Numerierung bezeichnet.

b) Zeigen Sie, daß die Knoten eines Suchbaumes aufsteigend sortiert sind, wenn sie in entsprechend ihrer Inorder-Nummern ausgegeben werden. Abstrakter formuliert, daß sich für jede Zuordnung $v \mapsto k(v)$ von Knoten zu Schlüsseln, welche der Suchbaumeigenschaft genügt, die Inorder–Nummern $inorder(v)$ und die Schlüsselwerte $k(v)$ nur durch eine feste monotone Bijektion unterscheiden.
Die Inorder–**DFS** durchläuft also die Knoten eines Suchbaumes genau in der durch die Schlüssel vorgegebenen Reihenfolge.

Aufgabe 22

Sei G der folgende Graph:



Wir nehmen an, daß die Prozedur **DFS/BFS** mit G aufgerufen wird und daß der Algorithmus immer zuerst die Knoten mit dem kleinsten Label betrachtet. Geben Sie die Präorder-, Postorder- und BFS-Numerierung von G an, die **DFS/BFS** dann erzeugen.