

# Übungsblatt 8

Abgabe bis 21.06.2018  
Besprechung: 25.06.2018 – 28.06.2018

## Aufgabe 1: Skiplists (5 Punkte)

Fügen Sie folgenden Elemente in eine Skiplist ein:

17, 3, 7, 11, 13, 19, 2, 23, 27.

Es stehen ihnen dazu folgende Zufallsbits zur Verfügung (von links nach rechts verbrauchen):

1101100101000110001.

Erhöhen Sie  $L_i$  so lange bis das Zufallsbit 1 ist. Geben Sie Ihre Datenstruktur nach jeder Einfügeoperation an.

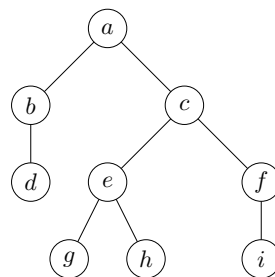
## Aufgabe 2: Skiplists und Binäre Suchbäume (5 Punkte)

Sei  $T$  ein binärer Suchbaum der Größe  $n$ , der nicht notwendigerweise balanciert ist. Beschreiben Sie einen Algorithmus in Pseudocode, der in Zeit  $\mathcal{O}(n)$  aus  $T$  eine Skipliste  $S$  bildet. Dabei soll  $S$  so aufgebaut sein, dass die Suche in  $S$  eine deterministische worst-case Laufzeit von  $\mathcal{O}(\log n)$  hat. Begründen Sie die Korrektheit Ihres Algorithmus.

## Aufgabe 3: Dynamisches Programmieren (1 + 4 + 4 + 1 Punkte)

Gegeben ein Baum  $T = (V, E)$ , wobei  $V$  die Menge der Knoten und  $E$  die Menge der Kanten zwischen diesen Knoten sind. Eine *unabhängige Menge* ist eine Menge  $M \subseteq V$ , sodass für alle  $v \in M$  gilt, dass  $w \notin M$ , falls es eine Kante  $\{v, w\} \in E$  gibt. Die unabhängige Menge  $M$  ist maximal, falls es keine andere unabhängige Menge gibt, die mehr Knoten als  $M$  enthält.

(a) Geben Sie eine maximale unabhängige Menge für das folgende Beispiel an:



- (b) Beschreiben Sie einen Algorithmus mit Dynamischem Programmieren, der eine maximale unabhängige Menge von  $T$  berechnet.
- (c) Begründen Sie die Korrektheit Ihres Algorithmus.
- (d) Wie ist die Laufzeit Ihres Verfahrens?

*Hinweis:* Lösungen, die kein Dynamisches Programmieren verwenden, werden mit 0 Punkten bewertet.

**Aufgabe 4:    Dynamisches Prog.: Palindrom**    (6 + 4 + (opt. 3 + 2) Punkte)

Ein *Palindrom* ist eine Zeichenfolge, die vorwärts und rückwärts gelesen gleich ist. Beispiele für Palindrome sind alle Zeichenfolgen der Länge 1, KAJAK, REITTIER, SEI FIES und DREH MAL AM HERD.

Gegeben sei eine Zeichenfolge  $S[1..n]$ . Eine *Teilfolge*  $S'$  von  $S$  besteht aus Buchstaben, die in  $S'$  in der selben Reihenfolge auftreten wie in  $S$ . Zum Beispiel ist ATMEN eine Teilfolge von ALGORITHMEN. Das *längste Palindrom* in  $S$  ist die längste Teilfolge von  $S$ , die ein Palindrom darstellt.

- (a) Geben Sie einen Algorithmus in Pseudocode an, der die Länge des längste Palindrom in  $S$  berechnet. Verwenden Sie dabei Dynamische Programmierung.
- (b) Begründen Sie die Korrektheit und die Laufzeit Ihres Algorithmus aus (a).
- (c) Geben Sie einen Algorithmus in Pseudocode an, der mit Hilfe von (a) das längste Palindrom in  $S$  berechnet und begründen Sie die Korrektheit und die Laufzeit Ihres Algorithmus.
- (d) Demonstrieren Sie die Funktionsweise Ihrer Algorithmen an der Zeichenfolge DADBBCAA.