

Übung 4

zu Theoretische Informatik III (für Softwaretechnik)

Aufgabe 1: Sortieren mit binären Bäumen

(9 Punkte)

Gegeben sind n Zahlen z_1, \dots, z_n (unsortiert). Diese sollen der Reihe nach eingegeben und hierbei in einen binären Suchbaum eingetragen werden. Anschließend wird der Baum inorder ausgegeben.

- Formulieren Sie diesen Algorithmus als Programm in Ada.
- Führen Sie eine Abschätzung der uniformen Zeitkomplexität im worst case, average case und best case durch. (Vgl. hierzu auch Einführung in die Informatik II, Analyse der Sortierverfahren.)

Aufgabe 2: Längster doppelpunktfreier Weg

(5 Punkte)

Gegeben sei ein gerichteter Graph $G = (V, E)$ und ein Knoten u . Man finde den längsten doppelpunktfreien Weg, der von u ausgeht, sowie dessen Länge (= Anzahl der Kanten im Weg).

Schreiben Sie einen Algorithmus hierfür und bestimmen Sie dessen unformale worstcase Komplexität in Abhängigkeit von der Anzahl der Knoten.

Aufgabe 3: (Backtracking)

(8+2 Punkte)

Ein Petrinetz ist ein 4-Tupel (S, T, F, M_0) , S = Menge der Stellen, T = Menge der Transitionen, $F \subseteq S \times T \cup T \times S$ = Flussrelation und $M_0 : S \rightarrow \mathbf{N}_0$ = Anfangsmarkierung. Sei $|S| = n$, $|T| = m$. Wir betrachten hier nur 1 - beschränkte Petrinetze (B/E - Netze), d.h. \forall Markierungen gilt: $M : S \rightarrow \{0, 1\}$.

- Schreiben Sie einen Algorithmus, der ein B/E - Netz einliest $(n, m, F_{i,j}, F_{j,i})$ mit $1 \leq i \leq n, 1 \leq j \leq m$. Die $F_{i,j}$ sind die in Transitionen eingehenden Kanten, die $F_{j,i}$ sind die ausgehenden Kanten. Eingelesen werden ferner M_0 (n stelliger Vektor) und anschließend eine Markierung M (als 0,1-Vektor). Der Algorithmus soll entscheiden, ob die Markierung M von M_0 aus erreichbar ist.
- Übertragen Sie den Algorithmus in ein Ada Programm und erstellen Sie einige Beispiele.