Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 1

Vorlesung Algorithmentechnik im WS 08/09

Ausgabe 21. Oktober 2008
Abgabe 4. November, 15:30 Uhr (im Kasten vor Zimmer 319, Informatik-Hauptgebäude, 3. OG)

Bitte schreiben Sie nur Ihren Namen und keine Matrikelnummer auf Ihr Übungsblatt. Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Problem 1: Amortisierte Analyse

Modellieren Sie eine Queue mit zwei Stacks so, dass die amortisierten Kosten von Dequeue und Enqueue jeweils in $\mathcal{O}(1)$ sind. Geben Sie die Operationen Dequeue und Enqueue in Pseudocode an und begründen Sie die amortisierten Kosten.

Problem 2: Untere Schranken

Gegeben sei eine Folge $(a_i)_{i=1}^n$ von n Elementen. Zeigen Sie, dass man mindestens $\Omega(\log n)$ Vergleiche benötigt, um den Index $i_{\max}$ des maximalen Elementes $a_{i_{\max}}$ zu bestimmen.

Problem 3: Laufzeit rekursiver Funktionen

Bestimmen Sie den größten ganzzahligen Wert a mit der Eigenschaft, dass ein Algorithmus mit der durch

$$T_B(n) = a \cdot T_B(n/4) + n^2$$

bestimmten Laufzeit im Θ-Kalkül asymptotisch schneller ist, als ein Algorithmus mit der durch

$$T_A(n) = 7 \cdot T_A(n/2) + n^2$$

bestimmten Laufzeit.

Problem 4: Das Tiefen-Bestimmungs-Problem

Beim *Tiefen-Bestimmungs-Problem* soll eine Menge $\mathcal{F}$ von *disjunkten* Bäumen verwaltet werden, auf denen die folgenden drei Operationen definiert sind:

Make-Tree(v) erzeugt einen neuen Baum, dessen einziger Knoten v ist.

Find-Depth(v) bestimmt die Tiefe des Knotens v innerhalb des Baumes T, zu dem v gehört. Die Tiefe von v ist die Anzahl der Kanten auf dem Weg von v zur Wurzel des Baumes T.

ATTACH(r, v) macht den Knoten v zum Vorgänger des Knotens r. Dabei wird vorausgesetzt, dass r die Wurzel eines Baumes ist und dass v zu einem anderen Baum gehört als r (v ist nicht unbedingt die Wurzel eines Baumes).

Modellieren Sie die Prozeduren MAKE-TREE(v), FIND-DEPTH(v) und ATTACH(r, v) in Pseudocode so, dass die Laufzeit von m Operationen MAKE-TREE, FIND-DEPTH und ATTACH in $\mathcal{O}(m \cdot G(m))$ ist. Die Funktion $G(m)$ sei wie im Skript definiert. Modifizieren Sie dazu die UNION-FIND-Datenstruktur aus der Vorlesung, um die Tiefe der Bäume dynamisch zu verwalten.

Hinweis: Benutzen Sie die UNION-FIND-Datenstruktur, um die Knotenmengen der Bäume zu verwalten. Jeder Knoten v eines Baumes $T \in \mathcal{F}$ sei dabei mit einem Wert $d(v)$ annotiert, so dass folgende Eigenschaft erfüllt ist: Sei $v_0, v_1, v_2, \ldots, v_k$ die Folge von Knoten, die FIND(v_0) in der UNION-FIND Datenstruktur von v_0 zur Wurzel v_k durchläuft, d.h. FIND(v_0) liefert als Ergebnis v_k. Dann ist die Tiefe von v_0 im Baum T der Wert der Summe $\sum_{i=0}^{k} d(v_i)$. Der Wert $d(v_i)$ entspreche dabei der Differenz der Tiefe (in T) von v_i und seinem direkten Vorgänger v_{i+1} in der UNION-FIND Datenstruktur, d.h. $d(v_i) :=$ FIND-DEPTH(v_i) − FIND-DEPTH(v_{i+1}) $(i < k)$, $d(v_k) :=$ FIND-DEPTH(v_k). Beachten Sie, dass $d(v)$ negativ sein kann.

Problem 5: Berechnung starker Zusammenhangskomponenten

Gegeben sei ein gerichteter Graph $D = (V, E)$. Für $u, v \in V$ schreiben wir $u \rightleftharpoons v$, wenn es einen gerichteten Weg von u nach v und einen gerichteten Weg von v nach u gibt.

(a) Zeigen Sie, dass $\rightleftharpoons$ eine Äquivalenzrelation ist.

(b) Die Äquivalenzklassen zu $\rightleftharpoons$ heißen *starke Zusammenhangskomponenten*. Schreiben Sie einen Algorithmus in Pseudocode, der die starken Zusammenhangskomponenten findet.

Benutzen Sie dazu eine modifizierte Tiefensuche. Speichern Sie besuchte Knoten in einer Union-Find-Struktur, in der Knotenmengen vereinigt werden, die zu einer starken Zusammenhangskomponente gehören.