TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Fachbereich Informatik
AG Algorithmen und Komplexität

Ausgabedatum: 20.11.2014
Version: 2014-11-19 16:04

4. Übungsblatt für alle Tracks zur Vorlesung Entwurf und Analyse von Algorithmen, WS 14/15

Abgabe: *Bis Freitag, 28.11.2014, 12:00 Uhr, Kasten im Treppenhaus 48-6.*

Basisaufgaben

B4.1: Speicherabbildungsfunktionen — 3 Punkte

Finden Sie eine sequentielle Speicherabbildungsfunktion, welche zu einer beliebigen $n \times n$ Matrix eine Erweiterung zu einer $(n+k) \times (n+k)$-Matrix, $k \geq 1$, ohne Umstellung der vorhandenen Elemente zulässt, und in Zeit $\mathcal{O}(1)$ (genauer: mit $\mathcal{O}(1)$ Elementaroperationen) berechnet werden kann. Begründen Sie die Korrektheit Ihrer Lösung!

B4.2: Potenzen der Adjazenzmatrix — 3 Punkte

a) Sei A die Adjazenzmatrix eines Digraphen G. Beweisen Sie, dass es in G für $k \in \mathbb{N}_0$ genau dann einen Weg von v_i nach v_j der Länge k gibt, wenn $(A^k)_{i,j} \neq 0$, d. h. wenn der Eintrag an Position i, j in der k-ten Potenz der Adjazenzmatrix ungleich 0 ist.

b) Verwenden Sie die Erkenntnis aus a), um die Anzahl der Wege der Länge n von einem Knoten v zu einem Knoten v' im vollständigen Digraphen $G = (V, V \times V)$ mit $|V| = n$ zu berechnen.

   Geben Sie eine geschlossene Form an; sie sollte mit höchstens n Elementaroperationen zu berechnen sein.

Hinweis: Können Sie eine Eigenschaft von A^k zeigen, die sowohl für a) als auch für b) hilfreich ist?

B4.3: Definitionen

3 Punkte

In dieser Aufgabe betrachten wir ein paar Definitionen nochmal genauer, die leicht für Verwirrung sorgen können.

a) Wir betrachten die folgenden Klassen von Bäumen:

   (i) ungeordnete erweiterte Binärbäume,

   (ii) geordnete erweiterte Binärbäume,

   (iii) ungeordnete 2-näre Bäume (= ungeordnete Binärbäume),

   (iv) geordnete 2-näre Bäume (= geordnete Binärbäume), und

   (v) binäre Bäume.

   Wir versuchen, diese Klassen etwas näher „kennenzulernen“.

   0) Erstellen Sie eine Liste mit (Verweisen im Buch zu) allen Definitionen.

   1) Geben Sie für jede der Klassen ein Beispiel mit mindestens fünf Knoten an.

   2) In welcher Beziehung stehen die Klassen zueinander?
   (Beispiel: A und B sind unvergleichbar oder $A \subsetneq B$)
   Geben Sie die jeweils stärksten Beziehungen an.

   3) Welche der Klassen sind Teilmengen von (Di-)Graphen?

b) Betrachten Sie folgende (nicht äquivalente) Definitionen für azyklische Graphen, wobei ein Weg $(v_1, \ldots, v_n)$ für $n \in \mathbb{N}$ *Kanten-einfach* heißt, wenn alle verwendeten Kanten paarweise verschieden sind.

   (Zur Abgrenzung nennen wir sind unsere alten „einfachen“ Wege auch *Knoten-einfach*.)

   Der gerichtete oder ungerichtete Graph $G = (V, E)$ heißt azyklisch, wenn

   (i) es keinen geschlossenen einfachen Weg gibt (Buch original).

   (ii) es keinen geschlossenen einfachen Weg der Länge ≥ 1 gibt.

   (iii) es keinen geschlossenen einfachen Weg der Länge ≥ 2 gibt.

   (iv) es keinen geschlossenen einfachen Weg der Länge ≥ 3 gibt (Errata zum Buch).

   (v) es keinen geschlossenen Kanten-einfachen Weg der Länge ≥ 1 gibt.

   (vi) es keinen geschlossenen Kanten-einfachen Weg der Länge ≥ 2 gibt.

   Der Einfachheit halber schreiben wir „G ist (iii)-azyklisch.“ statt „G ist azyklisch nach Definition (iii).“

   1) Welche der Definitionen sind „unnütz“ in dem Sinne, dass entweder nur endlich viele oder alle bis auf endlich viele (Di)Graphen azyklisch sind?

   2) Geben Sie Beispiele an, die zeigen, dass alle Definitionen unterschiedlich sind.

   3) Welche der Definitionen spiegelt Ihre Intuition zu dem Begriff „kreisfrei“ wider?

B4.4: Kombination von Baumtraversierungen 3 Punkte

Im Zuge der Diskussion des Traversierens von Bäumen

a) hatten wir festgestellt, dass weder In-, Pre- noch Postorder einen Baum eindeutig festlegt. Finden Sie für jeden der drei Fälle jeweils ein Beispiel, das diese Aussage belegt.

b) hatten wir bemerkt, dass die Pre- und die Postorder gemeinsam verwendet werden können, um den zugehörigen Baum eindeutig zu rekonstruieren. Entwerfen Sie einen Algorithmus, der genau dies leistet. Begründen Sie, warum Ihr Algorithmus die gestellte Aufgabe auch tatsächlich löst.

   Wenden Sie Ihr Verfahren zur Verdeutlichung beispielhaft auf die Pre- und Postorder eines beliebigen Baums mit

   - mindestens neun Knoten,
   - einer Höhe von mindestens vier und
   - mindestens einem Knoten, der mindestens drei innere Knoten als Kinder hat,

   an. Illustrieren Sie den Ablauf Ihres Algorithmus.

   Beachten Sie, dass nach Definition die Knotenlabel paarweise verschieden sind.

   Hinweis: Um einen Baum eindeutig aus seiner Pre- und Postorder zu rekonstruieren, muss man schrittweise die Wurzeln seiner Teilbäume identifizieren. Untersuchen Sie also, wie Sie anhand der Pre- und Postorder die Wurzel des Baumes bestimmen können, wie Sie anschließend die Wurzeln aller Teilbäume der Wurzel ablesen können usw. Aus den entsprechenden Beobachtungen ist es dann einfach möglich, ein passendes Verfahren abzuleiten.

Aufbauaufgaben

A4.1: Asymptotik der Fakultätsfunktion 3 Punkte

Beweisen Sie:
Für alle $c \in \mathbb{R}^+$ gilt

$$\omega\left(c^n\right) \ni n! \in o\left(n^n\right) \ . \qquad \text{(Lemma A: Fakultät)}$$

A4.2: Randfälle des Mastertheorems

3 + 2 Punkte

Betrachten Sie die folgende (schematische) Rekursionsgleichung:

$$T(n) = 2T\left(\frac{n}{2}\right) + f(n) \ ,$$

wobei beliebige Anker mit positiven Konstanten vorliegen können. Beweisen oder widerlegen Sie:

a) Wenn $f \in \Theta(n^2)$, dann gilt $T \in \Theta(f)$.

b) Wenn $f \in \Omega(n^2)$, dann gilt $T \in \mathcal{O}(f)$.

Tipp: Können Sie das Mastertheorem verwenden, auch wenn – wie hier – f nicht vollständig gegeben ist?

A4.3: Sätze über Breitensuche

3 + 3 Punkte

a) Beweisen Sie Satz 2.8:

> Sei $G = (V, E)$ ein gerichteter oder ungerichteter Graph, der als Adjazenzliste repräsentiert ist. Sei $w \in V$.
>
> i) `Breitensuche(w)` besucht jeden Knoten in V_w genau einmal und sonst keinen anderen Knoten.
>
> ii) `Breitensuche(w)` läuft in Zeit höchstens $\mathcal{O}(|V| + |E_w|)$.

b) Beweisen Sie Satz 2.9:

> Der von `Breitensuche(w)` erzeugte Aufrufbaum T_w ist ein Baum von kürzesten Wegen für G.