

Übungen zur Vorlesung
Datenstrukturen, Algorithmen und Programmierung 2 (DAP2)
Sommersemester 2007

Blatt 3

Aufgabe 3.1 (5 Punkte)

Implementiere einen Algorithmus (C, C++ oder Java) auf Basis der Tiefensuche, der einen gerichteten Graphen in T-, F-, C- und B-Kanten einteilt und überlege, wie man mit Hilfe dieser Einteilung entscheiden kann, ob der Graph kreisfrei ist.

Aufgabe 3.2 (5 Punkte)

Sei $G = (V, E)$ ein gerichteter Graph, der durch seine Adjazenzmatrix gegeben ist. Ein Knoten $v \in V$ heißt *schwarzes Loch*, falls v keine ausgehenden Kanten hat und es für alle $w \in V \setminus \{v\}$ eine Kante von w nach v gibt. Beschreibe einen Algorithmus (C, C++, Java oder Pseudocode) mit begründeter Laufzeit $O(|V|)$, der testet, ob G ein schwarzes Loch enthält.

Aufgabe 3.3 (5 Punkte)

In einer Stadt soll das Straßennetz optimiert werden, welches als ein gerichteter Graph $G = (V, E)$ mit $|V| = n$ in Form einer Adjazenzmatrix und einer Adjazenliste vorliegt. Die nummerierten Knoten stellen Vermessungs- und zugleich Straßenabschnittsendpunkte dar. Die gerichteten Kanten geben an, in welche Fahrtrichtung ein Straßenabschnitt befahrbar ist. Zwecks Optimierung des Straßennetzes ist es erst einmal notwendig, alle Sackgassen ausfindig zu machen (wir setzen voraus, dass es keine Einbahnstraßensackgassen gibt). Wähle die zur Lösung dieses Problems effizientere Datenstruktur und begründe Deine Entscheidung anhand einer Laufzeitanalyse. Beschreibe einen Algorithmus (C, C++, Java oder Pseudocode) unter Benutzung der gewählten Datenstruktur, der alle Sackgassenendpunkte (Sackgassenendpunkt ist ein Vermessungspunkt, an dem eine Sackgasse endet) ausgibt.

Aufgabe 3.4 (5 Punkte)

Welches Intervall $[i, j]$ ist für die Berechnung von $w(i, j)$ auf einem Segmentbaum mit $n = 2^k$ Blättern eine *worst-case*-Eingabe? Begründe deine Antwort.