Algorithmen und Datenstrukturen

Informatik II

Probeklausur Sommersemester 2008

Aufgabe 1 (3 Punkte)

Beweisen Sie:
$lim_{n\to\infty} f(n)/g(n) = c,\ mit\ 0 < c < \infty \Longrightarrow f(n) \in \Theta(g(n))$.

Aufgabe 2 (10 Punkte)

Doppelt verkettete Listen (alle Teilaufgaben in Pseudocode):

1. Definieren Sie eine Klasse für eine dynamische doppelt verkettete Liste und die Klasse eines Listenelements zum Speichern von Integers. Geben Sie den Code für die Listenfunktionen $pushfront$ (Einfügen eines gegebenen Integers am Anfang) und $popfront$ (Erstes Element Löschen) an. (3 Punkte)
2. Entwickeln Sie die Funktionen $size$, die Elemente einer Liste zählt, und $splithalf$, die eine Eingabeliste in der Mitte teilt und zwei Listen zurückgibt. (3 Punkte)
3. Entwickeln Sie die Funktion $merge$, die zwei sortierte Listen in linearer Zeit zu einer sortierten Liste vereint. (3 Punkte)
4. Entwickeln Sie die Funktion $mergesort$, die eine Liste mit Hilfe der zuvor definierten Funktionen sortiert. (1 Punkte)

Benutzen Sie für Aufgabenteile 2,3 und 4 keine *push* und *pop* Operationen.

Aufgabe 3 (4 Punkte)

Illustrieren Sie die Arbeitsweise von QUICKSORT an der Eingabefolge

5 3 17 10 84 19 6 22 9

Wählen Sie als Pivotelement immer das erste Element des zu sortierenden Intervalls. Vergessen Sie nicht das Pivotelement nach erfolgter Partitionierung in die Mitte der beiden Teilintervalle zu tauschen.

Aufgabe 4 (6 Punkte) Heapsort

1. Geben Sie den Code für die *sink* Funktion an. (4 Punkte)
2. Bauen Sie mit Hilfe der *sink* Funktionen einen Heap auf. (2 Punkte)

Aufgabe 5 (6 Punkte)

Gegeben sei ein Feld $A[1..n]$ von n Zahlen, die aufsteigend sortiert sind. Schreiben Sie Algorithmen für folgende Probleme:

1. Aufbau eines binären Suchbaumes T für A. (3 Punkte)
2. Suche nach einer gegebenen Zahl x im Baum T (Lookup). (3 Punkte)

Geben Sie die jeweiligen Laufzeiten an !

Aufgabe 6 (5 Punkte)

Sei $G = (V, E)$ ein gerichteter Graph und $s \in V$ ein Knoten. Verwenden Sie BFS (Breitensuche), um für jeden Knoten v die minimale Länge (Anzahl der Kanten) eines Pfades von s nach v zu berechnen. Geben Sie einen entsprechenden Algorithmus im Pseudo-Code an.

Aufgabe 7 (6 Punkte)

Sei $G = (V, E)$ der gerichtete Graph definiert durch folgende Adjazenzlisten:

Knoten	**Adjazenzliste des Knoten**
a	b
b	c f e
c	d g
d	c h
e	f a
f	g
g	f h
h	

Wenden Sie den DFS-Algorithmus auf G an.

1. In welcher Reihenfolge werden die Knoten von G von der Tiefensuche besucht, sprich: Wie lauten die `dfsnum`'s der Knoten von G? (2 Punkte)
2. In welcher Reihenfolge werden die DFS-Aufrufe beendet, d.h., wie lauten die `compnum`'s der Knoten von G? (2 Punkte)
3. Geben Sie für jede Kante an, ob es sich um eine Baum-, eine Vorwärts-, eine Rückwärts- oder eine Querkante handelt. (2 Punkte)