Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Angelika Steger

WS 1996/7
Semestralklausur
27. Februar 1997

Effiziente Algorithmen und Datenstrukturen

Bearbeitungszeit für die Klausur ist 120 Minuten. Als Hilfsmittel ist nur ein beliebig beschriebenes DIN A4-Blatt erlaubt. Zum Bestehen müssen von den 40 möglichen Punkten mehr als 16 Punkte erreicht werden. Schreiben Sie bitte nicht in den Farben rot oder grün.

Aufgabe 1 (4 Punkte)

Beschreiben Sie eine Datenstruktur $\mathcal{S}$, die die folgenden Operationen zur Verfügung stellt:

INITIALIZE: Die Datenstruktur $\mathcal{S}$ wird initialisiert (und ist danach leer).

INSERT(x): Der Datensatz x wird in $\mathcal{S}$ eingefügt, wobei wir annehmen, daß jedem Datensatz ein (nicht unbedingt eindeutiger) Schlüssel `key`(x) zugeordnet ist.

EXTRACTALL: Für jedes in $\mathcal{S}$ enthaltene Element wird der Befehl „`print key`(x)" ausgeführt (in beliebiger Reihenfolge). Anschließend werden alle Datensätze aus $\mathcal{S}$ entfernt.

Die Datenstruktur soll so beschaffen sein, daß jede beliebige Folge von k Operationen, die mit INITIALIZE beginnt, in Zeit $\mathcal{O}(k)$ abläuft. Erklären Sie detailiert, warum Ihre Datenstruktur die verlangte Eigenschaft hat.

Aufgabe 2 (1+1+1+3 Punkte)

a) In einen zunächst leeren (2,3)-Baum werden nacheinander die Zahlen 1,2,3,4,8,7,6,5 (in dieser Reihenfolge) eingefügt. Beschreiben Sie, wie sich der (2,3)-Baum entwickelt. (Zeichnungen genügen!)

b) In einen zunächst leeren Splay-Tree werden nacheinander die Zahlen 1,2,3,4,8,7,6,5 (in dieser Reihenfolge) eingefügt. Beschreiben Sie, wie sich der Splay-Tree entwickelt. (Zeichnungen genügen!)

c) In einen zunächst leeren Fibonacci-Heap werden nacheinander die Zahlen 1,2,3,4,8,7,6,5 (in dieser Reihenfolge) eingefügt. Beschreiben Sie, wie sich der Fibonacci-Heap entwickelt. (Zeichnungen genügen!)

d) In einem nächsten Schritt werde nun jeweils das minimale Element entfernt. Wie sehen die drei Datenstrukturen danach jeweils aus?

Bitte wenden!

Aufgabe 3 (3+5 Punkte)

a) Gegeben seien drei sortierte Mengen mit jeweils m Elementen. Zeigen Sie, daß diese drei Mengen mit Hilfe von $5m-2$ Vergleichen gemischt, d.h. zu einer einzigen sortierten Menge vereinigt werden können.

b) 3-MergeSort sei eine Variante von MergeSort, bei der die zu sortierende Menge in 3 statt 2 gleich große Mengen aufgeteilt wird, diese dann jeweils rekursiv sortiert und schließlich gemischt werden. Zeigen Sie, daß 3-MergeSort zum Sortieren einer n-elementigen Menge, wobei $n = 3^k$ für ein $k \in \mathbb{N}$, höchstens $\frac{5}{3} n \log_3 n$ Vergleiche benötigt.

Aufgabe 4 (2+6 Punkte)

Eine Geldbetrag $n \in \mathbb{N}$ soll in Münzen mit natürlichzahligen Werten $1 = w_1 \leq \ldots \leq w_k$ herausgegeben werden, wobei die *Anzahl* der Münzen so klein wie möglich sein soll.

Ein Greedy-Verfahren geht wie folgt vor: wähle die (betragsmäßig) größte Münze, die kleiner gleich n ist, dann die größte, die kleiner gleich dem Restbetrag ist, u.s.w..

a) Geben Sie ein Beispiel (d.h. ein n und ein Münzsystem $w_1 \leq \ldots \leq w_k$) an, für das das Greedy-Verfahren *nicht* die optimale Lösung liefert.

b) Sei $w_i = c^{i-1}$ für alle $1 \leq i \leq k$ und ein $c \in \mathbb{N}$. Zeigen Sie, daß das Greedy-Verfahren für dieses Münzsystem für jede natürliche Zahl n eine optimale Lösung liefert.

Aufgabe 5 (6 Punkte)

Sei $G = (V, E)$ ein Graph auf n Knoten. Ein *Hamilton'scher Pfad* in G ist ein Pfad der Länge n (mit beliebigen aber verschiedenen Anfangs- und Endpunkten).

Überlegen Sie sich einen Algorithmus, der in ungerichteten Graphen $G = (V, E)$, die die Bedingung

$$\deg(u) + \deg(v) \geq n - 1 \qquad \text{für alle } u \neq v \in V \text{ mit } \{u, v\} \notin E$$

erfüllen, einen Hamilton'schen Pfad bestimmt. Die Laufzeit Ihres Algorithmus sollte nicht $\mathcal{O}(|V| \cdot |E|)$ überschreiten.

Bemerkung: Sie dürfen Algorithmen der Vorlesung verwenden!

Aufgabe 6 (8 Punkte)

Es sei $G = (V, E)$ ein zusammenhängender Graph und $\ell : E \to \mathbb{R}$ eine Längenfunktion, so daß $\ell(e_1) \neq \ell(e_2)$ für alle $e_1, e_2 \in E$ mit $e_1 \neq e_2$. Zeigen Sie: In G gibt es genau einen spannenden Baum minimaler Länge.

Viel Erfolg!