

Übungen zur Informatik III im WS 02/03

Blatt 5

Abgabe: 19.11.2002

Aufgabe 1. (3 Punkte)

Sei $(\mathcal{A}_k)_{k \in K}$ eine Folge von Algorithmen, die jeweils das UNION-FIND-Problem für eine Folge von $(n-1)$ UNION- und $m < n$ FIND-Operationen in Zeit $f(\mathcal{A}_k) := 3km + kn^2$ durchführen. Sei \mathcal{B} ein Algorithmus, der das UNION-FIND-Problem für die gleiche Folge in Zeit $\mathcal{O}(m + n \log n)$ durchführt. Wir setzen $K := \{1, 2, \dots, 5\}$.

- a) Geben Sie kleinste obere Schranken in \mathcal{O} -Notation bzgl. der Laufzeit für jedes Element aus $(\mathcal{A}_k)_{k \in K}$ an. Die von Ihnen angegebenen Laufzeitschranken seien im folgenden mit $g(\mathcal{A}_k)$ referenziert.
- b) Geben Sie für jedes Element aus $(\mathcal{A}_k)_{k \in K}$ an, wieviel Zeit jeweils für eine Folge von

- 1 UNION-, 1 FIND-Operation
- 9 UNION-, 9 FIND-Operationen
- 99 UNION-, 42 FIND-Operationen
- 1000 UNION-, 2 FIND-Operationen

benötigt wird. Vergleichen Sie diese Zeiten mit den entsprechenden Werten für $g(\mathcal{A}_k)$.

- c) Vergleichen Sie die in b) ermittelten Werte für $g(\mathcal{A}_k)$ mit den entsprechenden Werten für die Laufzeit von \mathcal{B} und von einem Algorithmus \mathcal{C} , dessen Laufzeit $5m + \frac{n^4}{2}$ beträgt, für die in Teil b) angegebenen Folgen von Operationen. Ist es bei der Abgrenzung von $(\mathcal{A}_k)_{k \in K}$ zu \mathcal{B} bzw. \mathcal{C} Ihrer Meinung nach sinnvoll, die exakten Laufzeiten zu vergleichen?

Aufgabe 2. (6 Punkte)

- a) Implementieren Sie eine Datenstruktur für das UNION-FIND Problem derart, dass bei UNION-Operationen nur ein Zeiger von einer der beiden Mengen auf die andere angelegt wird, eine UNION-Operation also in konstanter Zeit möglich ist (Sie dürfen annehmen, dass arithmetische Operationen nur eine Zeiteinheit in Anspruch nehmen). Die FIND-Operationen dürfen die Datenstruktur (hier) nicht verändern.

Wie lange dauert eine FIND-Operation? Können Sie dafür sorgen, dass für die maximale Dauer einer FIND-Operation eine obere Schranke angegeben werden kann?

- b) Nehmen Sie nun an, dass die FIND-Operationen ebenfalls die Datenstruktur verändern dürfen. Geben Sie eine Implementierung an, bei der k UNION-Operationen, gefolgt von $m - k$ FIND-Operationen, nur $\mathcal{O}(m)$ Zeit kosten.

Aufgabe 3. (6 Punkte)

Sei σ eine Folge von Instruktionen der Form INSERT(i) bzw. DELMIN, die auf einer anfangs leeren Menge S ausgeführt werden soll. Die Bedeutung der Instruktionen sei wie folgt vorgegeben:

$$\begin{aligned} \text{INSERT}(i) &: S := S \cup \{i\} \quad (i \in \{1, 2, \dots, n\}) \\ \text{DELMIN} &: S := S \setminus \min(S) \end{aligned}$$

Für $\sigma = \sigma_1\sigma_2 \dots \sigma_m$ soll gelten: $(\sigma_i = \text{INSERT}(k), \sigma_j = \text{INSERT}(l), i \neq j) \Rightarrow k \neq l$, d.h. es wird nicht zweimal das gleiche Element in S eingefügt. Sei $\mathcal{D}(\sigma)$ die Anzahl der DELMIN-Instruktionen in σ .

- a) Schreiben Sie einen Algorithmus, der σ als Eingabe erhält und für jede DELMIN-Instruktion σ_{d_i} ($1 \leq i \leq \mathcal{D}(\sigma)$) das Paar (i, m_i) ausgibt, wobei m_i das von σ_{d_i} aus S entfernte Element ist¹. Benutzen Sie zur Repräsentation Ihrer Daten nur Variablen und Felder. Geben Sie eine obere Schranke für die Laufzeit des Algorithmus an.
- b) Schreiben Sie einen Algorithmus, der das offline MIN-Problem durch Anwendung von UNION- und FIND-Operationen löst. Geben Sie auch hier eine obere Schranke für die Laufzeit des Algorithmus an und vergleichen Sie sie mit derjenigen aus Teil a).

Aufgabe 4. (2 Punkte)

Nehmen Sie an, dass bei einer Prioritätswarteschlange nur die Prioritäten $\{1, 2, \dots, c\}$, c konstant, auftreten. Geben Sie eine Implementierung dieses Schlangentyps mit den Operationen max (liefert ein Element maximaler Priorität und entfernt dies aus der Schlange) und insert(x, p) (x Objekt, p Priorität) an, wobei jede Operation nur konstante Zeit in Anspruch nehmen darf.

□

¹Ihr Algorithmus stellt eine Lösung für das sog. *offline MIN*-Problem dar.