

Theoretische Informatik 3

2. Übung

Abgabe der schriftlichen Lösungen bis zum 20. Mai 2008

Aufgabe 8 [6 Punkte]

Sei π die Präfixfunktion für ein beliebiges Muster $y = y_1 \cdots y_m \in \Sigma^*$ und sei δ die Überführungsfunktion von M_y . Betrachten Sie folgende auf der Menge $\{1, \dots, m\}$ definierte Funktion

$$\pi'(k) = \max \left\{ j \geq 0 \mid \begin{array}{l} y_1 \cdots y_j \text{ ist echtes Suffix von } y_1 \cdots y_k \\ \text{und im Fall } k < m \text{ ist } y_{j+1} \neq y_{k+1} \end{array} \right\}.$$

- Berechnen Sie π und π' für das Muster $y = (ab)^{10}$. (mündlich)
- Zeigen Sie, dass $\pi'(k) \leq \pi(k)$ für alle $k = 1, \dots, m$ gilt. (1 Punkt)
- Zeigen Sie, dass der KMP-Algorithmus bei einem Mismatch im Zustand k mindestens bis zum Zustand $\pi'(k)$ zurückspringt, bevor er das nächste Zeichen liest. (1 Punkt)
- Zeigen Sie, dass der KMP-Algorithmus auch bei Verwendung von π' anstelle von π korrekt arbeitet. (1 Punkt)
- Zeigen Sie, dass π' (wie π) in Zeit $O(m)$ berechenbar ist. (1 Punkt)
- Zeigen Sie, dass δ bei Kenntnis von π' in Zeit $O(\|\Sigma\|m)$ berechenbar ist. (2 Punkte)

Aufgabe 9 [mündlich]

Bubble-Sort sortiert eine Zahlenfolge a_1, \dots, a_n durch wiederholtes Vertauschen von benachbarten Folgengliedern:

```
1 for i := n - 1 downto 1 do
2   for j := 1 to i do
3     if  $a_j > a_{j+1}$  then vertausche  $(a_j, a_{j+1})$ 
```

- Wie verarbeitet Bubble-Sort die Eingabefolge 3, 6, 1, 7, 9, 2, 4, 8?
- Finden und beweisen Sie eine geeignete Invariante für die innere **for**-Schleife.

- Benutzen Sie die Schleifeninvariante aus b) für den Nachweis einer geeigneten Invariante für die äußere **for**-Schleife.
- Beweisen Sie die Korrektheit von Bubble-Sort mithilfe der Invariante aus c).
- Bestimmen Sie die Anzahl an Vergleichen, die Bubble-Sort im besten und im schlechtesten Fall benötigt.
- Verbessern Sie die bestcase-Komplexität von Bubble-Sort, indem Sie die äußere **for**-Schleife durch eine **repeat**-Schleife ersetzen, die für möglichst wenige Werte von i durchlaufen wird und insbesondere abbricht, sobald die Folge sortiert ist.

Aufgabe 10 [mündlich]

Konstruieren Sie einen vergleichsbasierten Algorithmus, der im schlechtesten Fall eine möglichst geringe Anzahl $V(n)$ an Vergleichen benötigt, um

- das Maximum und das Minimum,
- das größte und zweitgrößte Element

einer Folge von n Zahlen zu finden.

Hinweis: Für a) sind $\lceil 3n/2 \rceil - 2$ und für b) sind $n + \lceil \log_2 n \rceil - 2$ Vergleiche optimal.

Aufgabe 11 [mündlich]

Die Höhe $h(v)$ eines Knotens in einem Baum B sei die maximale Länge eines Pfades von v zu einem Blatt. Die Summe $\sum_v h(v)$ der Höhen aller Knoten in B bezeichnen wir mit $S(B)$. Zeigen Sie:

- In einem Heap H mit $n = 2^k$ Knoten ($k \geq 0$) ist $S(H) = n - 1$.
- In einem Heap H mit $n = 2^k - 1$ Knoten ($k \geq 1$) ist $S(H) = n - \log_2(n + 1)$.
- In einem Heap H mit n Knoten gilt $S(H) = \Theta(n)$.

Aufgabe 12 [4 Punkte]

Bestimmen Sie die minimale Anzahl an Vergleichen, die ein vergleichsbasierter Algorithmus im besten Fall benötigt, um

- eine Folge von n Zahlen zu sortieren, (mündlich)
- den Median einer Folge von n Zahlen zu finden, (mündlich)
- das Maximum einer Folge von n Zahlen zu finden, (2 Punkte)
- das Maximum und das Minimum einer Folge von n Zahlen zu finden. (2 Punkte)