CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL
Institut für Informatik, Arbeitsgruppe Algorithmen und Komplexität
Prof. Dr. K. Jansen, K.-M. Klein, F. Land M. Rau

04. Mai 2016

Hausaufgaben zur Vorlesung »Algorithmen und Datenstrukturen«

Blatt 4

Hausaufgabe 4.1 (Rekurrenz (3 Punkte))
Gegeben seien die folgenden Rekurrenzgleichungen für $a, b \in \mathbb{N}$:

1. $T(0) = a$, $T(n) = T(n-1) + b$ für alle $n \in \mathbb{N} \backslash \{0\}$
2. $T(1) = 1$,
   $T(n) = T(\frac{n}{2}) + n$ für alle $n \in \mathbb{N}$ derart, dass es ein $k \in \mathbb{N} \backslash \{0\}$ gibt mit $n = 2^k$,
   $T(n) = T(2^k)$ mit $k = \lfloor \log_2(n) \rfloor$ in den restichen Fällen.

Zeigen Sie, dass in beiden Fällen $T(n) \in \mathcal{O}(n)$ gilt.

Hausaufgabe 4.2 (Laufzeit (4 Punkte))
Die binäre Suche teilt einen Suchbereich sukzessive in zwei gleichgroße Teile auf. Dieses Verfahren funktioniert auch, wenn der Suchbereich sukzessive in drei gleichgroße Teile aufgeteilt wird. Dieses Verfahren heißt ternäre Suche. Ermitteln Sie die Anzahl der Rechenoperationen im folgenden Algorithmus. Gehen Sie davon aus, dass pro Zeile eine Rechenoperation benötigt wird. Geben Sie anschließend die Gesamtlaufzeit des Algorithmus gemäß der $\mathcal{O}$-Notation an.

Algorithmus SEARCH (A, x)

```
int   a = 0, b=n;
int   n = A.length;
int   k1,k2;
while a<b do
  k1 = (b+2*a)/3;
  k2 = (2*b+a)/3;
   if  (x = A[k1])  then
    return  k1;
  elseif  (x = A[k2]) then
    return  k2;
  elseif  (x < A[k1]) then
    b=k1;
  elseif  (x < A[k2]) then
    a=k+1;
    b=k2;
  else
    a = k2+1;
  fi
od
return -1
```

Hinweis: Stellen Sie eine Rekurrenzgleichung auf und verwenden Sie ohne Beweis, dass für $a, b \in \mathbb{N}_{>0}$ und $T : \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ mit

$$T(n) \leq \begin{cases} a & \text{falls } n = 1 \\ b + T(\lfloor n/k \rfloor) & \text{sonst} \end{cases}$$

gilt, dass

$$T(n) \leq a + b\lfloor \log_k(n) \rfloor$$

für alle $n \in \mathbb{N}_{>0}$ ist (vgl. Hausaufgabe 3.4).

Hausaufgabe 4.3 (Hanoi (3 Punkte))
Zeigen Sie, dass der in der Vorlesung präsentierte rekursive Algorithmus für die Türme von Hanoi die minimale Anzahl an Zügen berechnet, um einen Turm der Höhe n von Position 1 auf Position 2 über 3 zu verschieben.

Hinweis: Sie können diese Aussage mittels Induktion zeigen.

Programmieraufgabe 4.4 (Rucksack Problem)
Implementieren Sie das in der Vorlesung vorgestellte dynamische Programm, um das Rucksack-Problem zu lösen. Erweitern Sie das dynamische Programm so, dass eine entsprechende Teilmenge von Items bestimmt und zurückgegeben wird.

Abgabe der theoretischen und praktischen Aufgaben Donnerstag, den 12. Mai, bis spätestens 14 Uhr. Die Abgabe der theoretischen Aufgaben erfolgt im Schrein. Die Abgabe der praktischen Aufgaben erfolgt im iLearn.