



Hausaufgaben zur Vorlesung »Algorithmen und Datenstrukturen«

Blatt 4

Hausaufgabe 4.1 (Rekurrenzgleichung (2 Punkte))

Lösen Sie die durch

$$T(0) = 0, \quad T(n) = 3n(n-1) + T(n-1) + 1 \text{ für alle } n \in \mathbb{N} \setminus \{0\}$$

definierte Rekurrenzgleichung und beweisen Sie Ihr Ergebnis, beispielsweise durch vollständige Induktion.

Hausaufgabe 4.2 (Ternäre Suche (3 Punkte))

Die binäre Suche teilt einen Suchbereich sukzessive in zwei gleichgroße Teile auf. Diese Verfahren funktioniert auch, wenn der Suchbereich sukzessive in drei etwa 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)

```
1  int n = A.length;
2  int a = 0, b = n; int k1, k2;
3  while a < b do
4      k1 = (b + 2 * a) / 3;
5      k2 = (2 * b + a) / 3;
6      if (x = A[k1]) then
7          return k1;
8      elseif (x = A[k2]) then
9          return k2;
10     elseif (x < A[k1]) then
11         b = k1;
12     elseif (x < A[k2]) then
13         a = k1 + 1;
14         b = k2;
15     else
16         a = k2 + 1;
17     fi
18 od
19 return -1
```

Hinweis: Zeigen Sie, dass sich der Suchbereich nach jeder Iteration drittelt. Stellen Sie dann eine Rekurrenzgleichung auf und verwenden Sie Hausaufgabe 3.2 um eine geschlossene Form dafür zu bekommen.

Hausaufgabe 4.3 (Rekursion (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 3 zu verschieben.

Hausaufgabe 4.4 (Programmierung (2 Punkte))

Gegeben ist ein Array T der Länge n , welches n ganze Zahlen des Typs `int` enthält. Implementieren Sie das in der Vorlesung vorgestellte Verfahren Bubblesort, um das Array T zu sortieren. Modifizieren Sie das Verfahren so, dass der Algorithmus vorzeitig abbricht, wenn das Array bereits in einem Zwischenschritt sortiert ist.

Verwenden Sie die von uns bereitgestellten Dateien und testen Sie Ihr Programm mit dem bereitgestellten Testprogramm. Verwenden Sie zur Ausführung des Testprogramms den Befehl `"python test.py Sort <Pfad>"`. Abgaben, die sich nicht mit dem Testprogramm testen lassen, werden mit 0 Punkten bewertet.

Abgabe der theoretischen Aufgaben Freitag, den 15. Mai, bis spätestens 8 Uhr im Schrein. Die Abgabe der Programmieraufgaben erfolgt per EMail an Ihren Übungsleiter.