Lehrstuhl für Informatik I
Algorithmen und Komplexität
Universität Würzburg

Würzburg, den 03. November 2022

Prof. Dr. Alexander Wolff
Felix Klesen, M. Sc.

3. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2022/23)

Aufgabe 1 – Rekursionsgleichung aufstellen

SomeAlgo(int[ ] A, int $\ell = 1$, int $r = A.length$)
if $\ell == r$ **then return** $A[\ell]$
$g \leftarrow r - \ell + 1$
let $B[1..3]$ be new array of int
$B[1] \leftarrow$ SomeAlgo($A,\ \ell,\ \ell + \lceil g/2 \rceil - 1$)
$B[2] \leftarrow$ SomeAlgo($A,\ \ell + \lfloor g/4 \rfloor,\ \ell + \lfloor g/4 \rfloor + \lceil g/2 \rceil - 1$)
$B[3] \leftarrow$ SomeAlgo($A,\ \ell + \lfloor g/2 \rfloor,\ r$)
InsertionSort(B)
return $B[1]$

a) Was gibt SomeAlgo zurück? **1 Punkt**

b) Stellen Sie eine Rekursionsgleichung für die asymptotische Laufzeit $T(n)$ von SomeAlgo(A) auf (mit $n := A.length$). Denken Sie an den Basisfall. **2 Punkte**

Aufgabe 2 – Rekursionsgleichungen lösen

Sei $T\colon \mathbb{N} \to \mathbb{R}$ eine Funktion, so dass $T(n)$ für $n = 1$ einen konstanten Wert annimmt und für alle anderen $n \in \mathbb{N}$ durch eine der folgenden Rekursionsgleichungen definiert ist. Geben Sie für jedes der folgenden T eine Funktion g an, so dass $T \in \Theta(g)$. Eine Begründung Ihrer Lösung ist zwingend erforderlich.

Drei Teilaufgaben sind mit der Meistermethode lösbar (geben Sie den jeweiligen Fall an), eine jedoch nicht (diese wird mit 2 Punkten bewertet). Nutzen Sie für diese die Rekursionsbaummethode. Nehmen Sie an, dass $n = 2^m$, für ein $m \in \mathbb{N}$. **5 Punkte**

a) $T(n) = 3T(\lfloor n/9 \rfloor) + \sqrt{n}$
b) $T(n) = 4T(\lfloor n/2 \rfloor) + n^3$
c) $T(n) = 4T(\lfloor n/2 \rfloor) + n^2 \log_2 n$
d) $T(n) = 5T(\lfloor n/2 \rfloor) + n^2$

Aufgabe 3 – Algorithmen erkennen (2)

Geben Sie für jeden der gegebenen Algorithmen an, was er bewirkt und bestimmen sie möglichst genau seine Laufzeit (Groß-Oh-Notation).

a) Algorithmus1(int[ ] A, int $\ell = 1$, int $r = A.length$) **2 Punkte**

```
if 1 ≤ ℓ and ℓ < r and r ≤ A.length then
    key = A[ℓ]
    A[ℓ] = A[r]
    A[r] = key
    Algorithmus1(A, ℓ + 1, r − 1)
```

b) Algorithmus2(int[ ] A, int k) **2 Punkte**

```
n = A.length
c = 0
for i = 1 to n do
    for j = i + 1 to n do
        if A[i] + A[j] == k then
            c = c + 1
return c
```

c) Algorithmus3(int[ ] A) **2 Punkte**

```
n = A.length
for i = 1 to n do
    if A[i] ≠ 0 then
        a = A[i]
        A[i] = 0
        b = Algorithmus3(A)
        if a > b then
            return b;
        return a;
return 0
```

Aufgabe 4 – FunSort

Gegeben sei folgender Sortieralgorithmus in Pseudocode, wobei die Methode Merge aus MergeSort übernommen wurde.

```
FunSort(array of int A)
i = 1
while i < A.length do
    Merge(A, 1, i, i + 1)
    i = i + 1
```

a) Welchem Algorithmus, den Sie aus der Vorlesung kennen, ähnelt FunSort? Begründen Sie Ihre Antwort. **1 Punkt**

b) Sei $T_{max}(n)$ die maximale Laufzeit von FunSort über alle Eingaben der Größe n. Geben Sie eine Funktion f an, so dass $T_{max} \in \Theta(f)$.

Begründen Sie Ihr Ergebnis. **1 Punkt**

c) Sei $T_{min}(n)$ die minimale Laufzeit von FunSort über alle Eingaben der Größe n. Geben Sie eine Funktion g an, so dass $T_{min} \in \Theta(g)$.

Begründen Sie Ihr Ergebnis. **1 Punkt**

d) Beweisen Sie die Korrektheit des Algorithmus mit folgender Schleifeninvarianten:

Bei der i-ten Ausführung des while-Schleifenkopfes gilt, dass

(i) $A[1..i]$ *dieselben Elemente wie zu Beginn der Ausführung des Algorithmus enthält – jedoch sortiert.*

(ii) $A[i+1..A.length]$ *sich seit der Ausführung des Algorithmus nicht verändert hat.*

3 Punkte

Bitte geben Sie Ihre Lösungen bis **Donnerstag, 10. November 2022, 14:00 Uhr** einmal pro Gruppe über Wuecampus als pdf-Datei ab. Vermerken Sie dabei stets die Namen und Übungsgruppen aller BearbeiterInnen auf der Abgabe.

Grundsätzlich sind stets alle Ihrer Aussagen zu begründen und Ihr Pseudocode ist stets zu kommentieren.

Die Lösungen zu den mit PABS gekennzeichneten Aufgaben, geben Sie bitte nur über das PABS-System ab. Vermerken Sie auf Ihrem Übungsblatt, in welchem Repository (*sXXXXXX*-Nummer) die Abgabe zu finden ist. Geben Sie Ihre Namen hier als Kommentare in den Quelltextdateien an.